



ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Grado en Ingeniería Electrónica Industrial y Automática

TRABAJO FIN DE GRADO

Título:

**ESTACIÓN METEOROLÓGICA DE BAJO
COSTE PARA EL ENTORNO AGRÍCOLA**

Autores:

Hugo Hilario Reyes
Luis Vallejo Segovia

Tutor: Santiago Torres Álvarez

Julio 2022

Agradecimientos:

A nuestro tutor Don Santiago Torres Álvarez, profesor contratado doctor, el cual nos ha ayudado a desarrollar nuestro prototipo, resolviendo las posibles dudas y los diferentes problemas.

A Doña Candelaria Fernández Mascareño, responsable de la Biblioteca y la Hemeroteca de la Escuela Superior de Ingeniería y Tecnología, por su simpatía y su buen hacer, mostrando disposición para ayudarnos.

También queremos agradecer a Manuel Fernández Vera, PAS del departamento de Ingeniería Informática y de Sistemas, José Francisco Gómez González, profesor titular del departamento de Ingeniería Industrial y José Fernando Pereza Hernández, los cuales nos han ayudado en la medida de lo posible.

Y finalmente, a nuestras familias y amigos que nos han mostrado su apoyo incondicional desde el inicio del grado hasta la ejecución final del TFG.

Índice General

I.	Resumen.....	5
II.	Abstract.....	6
III.	Memoria.....	7
	1. Objetivos	
	2. Alcance	
	3. Antecedentes	
	4. Magnitudes a medir	
	5. Características de los elementos de la estación	
	6. Diseño de la estación meteorológica	
	7. Resultados	
	8. Futuras mejoras	
	9. Bibliografía	
IV.	Presupuesto y mediciones.....	94
V.	Conclusiones.....	100
VI.	Conclusions.....	102
VII.	Anexos.....	104
	Anexo I: Archivos de los diseños 3D.....	105
	Anexo II: Programación.....	106
	Anexo III: Fichas técnicas de los componentes.....	116

I. Resumen

El objetivo principal de este trabajo es llevar a cabo el prototipo funcional de una estación meteorológica de bajo coste de utilización en el entorno agrícola.

En las Islas Canarias existe una amplia red de estaciones meteorológicas repartidas por el territorio insular. Sin embargo, se tiene la necesidad de incorporar estaciones meteorológicas en las explotaciones agrarias, permitiendo la modernización y mejora en su gestión a los usuarios.

Para determinar los componentes necesarios en el prototipo, se ha realizado un estudio previo de las magnitudes a medir. Se han contrastado distintas fuentes que nos han permitido conocer las variables meteorológicas de interés en el entorno agrícola. Tras fijar los parámetros que se van a medir, se han aportado las soluciones técnicas que permitirán realizar el registro de dichos parámetros. De esta manera, se han empleado una serie de elementos hardware para componer el prototipo de la estación meteorológica, siendo necesario un soporte digital e interfaces de comunicación para los mismos.

Una vez seleccionados los componentes, se ha procedido a efectuar el diseño de la estación meteorológica. En primer lugar, se ha realizado la estructura general del código para un funcionamiento concreto de la estación. A continuación, se han programado los elementos de la estación para obtener los parámetros necesarios, realizando también las conexiones eléctricas de los mismos. Después, han sido diseñadas las carcasas de cada componente, confiriéndoles cierto grado de protección. Cabe destacar que se ha requerido de la elaboración de una estructura específica para algunos de los componentes. Finalmente, se ha ensamblado el prototipo de la estación meteorológica.

Por último, se realizó la puesta en marcha de la estación, realizando una serie de mediciones y el análisis de los resultados.

Palabras Clave: Estación meteorológica, bajo coste, sensores, programación, conexión, variables meteorológicas, componentes, carcasas y Arduino.

II. Abstract

The main objective of this final bachelor work is to carry out the functional prototype of a low-cost meteorological station to be used in the agricultural environment.

In the Canary Islands there is an extensive network of meteorological stations spread throughout the island's territory. However, incorporating meteorological stations in farms has become necessary, allowing the modernisation and improvement in their management for users.

In order to determine the needed components for the prototype, a preliminary study of the magnitudes to be measured was carried out. Different sensors have been compared to allow the knowledge of the meteorological variables of interest in the agricultural environment. After establishing the parameters to be measured, technical solutions were provided to enable the recording of these parameters. In this way, a series of hardware elements have been used to make up the meteorological station prototype, requiring a digital support and communication interfaces for them.

Once the components had been selected, the meteorological station was designed. Firstly, the general structure of the code was created for the specific operation of the station. Next, the elements of the station were programmed to obtain the necessary parameters, also making the electrical connections to them. The shields of each component were then designed, giving them a certain degree of protection. We should bring up that a specific structure was required for some of the components. Finally, the meteorological station prototype was assembled.

Finally, the set up of the station was carried out, obtaining a series of measurements and analysing the results.

Key Words: meteorological station, low cost, sensors, programming, connection, meteorological variables, elements, shields y Arduino.



ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Grado en Ingeniería Electrónica Industrial y Automática

III. MEMORIA

TRABAJO FIN DE GRADO

Título:

**ESTACIÓN METEOROLÓGICA DE BAJO
COSTE PARA EL ENTORNO AGRÍCOLA**

Índice de la memoria

1. OBJETIVOS	14
1.1. OBJETIVOS TÉCNICOS	15
1.2. OBJETIVOS ACADÉMICOS	15
2. ALCANCE	16
3. ANTECEDENTES	16
4. MAGNITUDES A MEDIR	17
4.1. TEMPERATURA DEL AIRE	17
4.2. PRESIÓN ATMOSFÉRICA	18
4.3. HUMEDAD	19
4.3.1. Humedad del Aire	19
4.3.2. Humedad del Suelo	20
4.4. VIENTO	20
4.5. CALIDAD DEL AIRE	21
4.6. RADIACIÓN SOLAR	22
4.7. VOLUMEN DE PRECIPITACIONES	23
4.8. SUMARIO DE MAGNITUDES	24
5. CARACTERÍSTICAS DE LOS ELEMENTOS DE LA ESTACIÓN	26
5.1. SOPORTE DIGITAL Y COMUNICACIONES EN LA ESTACIÓN	26
5.1.1. Arduino	26
5.1.2. Protocolos de comunicación	27
5.1.3. SketchUp	29
5.1.4. Ultimaker Cura	30
5.1.5. Fritzing	30
5.2. ELEMENTOS HARDWARE QUE COMPONEN LA ESTACIÓN	31
5.2.1. Placa de desarrollo Arduino Mega 2560	31
5.2.2. Sensor DHT22	33
5.2.3. Sensor MQ-135	34
5.2.4. Sensor HC-SR04	36
5.2.5. Sensor ML8511	37
5.2.6. Sensor BMP280	38
5.2.7. Sensor HD-38	40
5.2.8. Sensor FC-03	41
5.2.9. Pantalla LCD	42
5.2.10. Módulo adaptador para tarjeta micro SD	44
5.2.11. Real Time Clock	45

6. DISEÑO DE LA ESTACIÓN METEOROLÓGICA	46
6.1. ESTRUCTURA GENERAL DEL CÓDIGO	46
6.2. CONEXIÓN Y PROGRAMACIÓN DE LOS COMPONENTES	53
6.2.1. Temperatura y humedad	53
6.2.2. Calidad del aire	55
6.2.3. Pluviómetro	56
6.2.4. Radiación UV	58
6.2.5. Presión y altitud	60
6.2.6. Humedad del suelo	61
6.2.7. Anemómetro	62
6.2.8. Pantalla LCD	64
6.2.9. Módulo para tarjeta micro SD	66
6.2.10. RTC	68
6.3. MONTAJE DEL PROTOTIPO	69
6.3.1. Protección de los componentes	69
6.3.1.1. Arduino Mega 2560	69
6.3.1.2. MQ135	70
6.3.1.3. HC-SR04	71
6.3.1.4. ML8511	72
6.3.1.5. BMP280	72
6.3.1.6. HD-38	73
6.3.1.7. FC-03	73
6.3.1.8. Pantalla LCD	74
6.3.1.9. Módulo para tarjeta micro SD	75
6.3.1.10. RTC	75
6.3.2. Pluviómetro y anemómetro	76
6.3.3. Estructura del prototipo	78
6.4. MANUAL DE USUARIO	80
7. RESULTADOS	81
7.1. CONDICIONES DE LA MEDICIÓN	81
7.2. INCIDENTES EN LA MEDICIÓN	82
7.3. MANIPULACIÓN DE LOS DATOS	83
8. FUTURAS MEJORAS	88
8.1. SOLUCIÓN A LOS FALLOS OBTENIDOS	88
8.2. NUEVAS MAGNITUDES	88
8.2.1. PH del agua	89
8.2.2. Humedad de las Hojas	89
8.3. COMUNICACIÓN CON EL SERVIDOR Y ENTRE LOS DISPOSITIVOS	90
8.3.1. Módulo wifi ESP8266 12F y ThingSpeak	90

8.3.2. ZigBee	91
9. BIBLIOGRAFÍA	91

Índice de figuras

Figura 1: Arduino Mega 2560 [29].	31
Figura 2: Esquema patillaje (<i>pinout</i>) Arduino Mega 2560 [31].	32
Figura 3: Sensor DHT22 [33].	33
Figura 4: Sensor MQ-135.	35
Figura 5: Curva característica de la sensibilidad del MQ-135 [36].	36
Figura 6: Sensor HC-SR04.	36
Figura 7: Sensor ML8511 [39].	37
Figura 8: Tensión de salida frente a la intensidad UV [40].	38
Figura 9: Sensor BMP280 [41].	39
Figura 10: Sensor HD-38 [35].	40
Figura 11: Sensor FC-03.	41
Figura 12: Pantalla LCD.	42
Figura 13: Parte trasera pantalla LCD más el módulo I2C.	43
Figura 14: Módulo micro SD [51].	44
Figura 15: RTC DS1307 [53].	45
Figura 16: Diagrama de flujo del funcionamiento de la estación.	47
Figura 17: Extracto del código. Librerías incluidas.	48
Figura 18: Extracto del código. Variables destacadas.	49
Figura 19: Extracto del código. Comprobación de la conexión con tarjeta micro SD.	49
Figura 20: Extracto de código. Ejemplo llamado de las funciones.	50
Figura 21: Extracto del código. Modo <i>Sleep</i> .	51
Figura 22: Extracto del código. Función del pulsador de parada.	52
Figura 23: Circuito de conexión del sensor DHT22.	54
Figura 24: Extracto de código. Función para obtener la temperatura del aire.	55
Figura 25: Circuito de conexión del sensor MQ-135.	55
Figura 26: Circuito de conexión del sensor HC-SR04.	57
Figura 27: Extracto de código. Medición de la distancia con sensor de ultrasonidos.	57
Figura 28: Circuito de conexión del sensor ML8511.	59

Figura 29: Extracto de código. Obtención de la radiación UV.	59
Figura 30: Extracto de código. Interpolación a partir de la curva del sensor ML8511.....	59
Figura 31: Circuito de conexión del sensor BMP280.	60
Figura 32: Modificación en la librería “Adafruit_BMP280.h”.....	61
Figura 33: Circuito de conexión del sensor HD-38.....	61
Figura 34: Circuito de conexión del sensor FC-03.	63
Figura 35: Extracto de código. Habilitación de interrupciones del anemómetro.	63
Figura 36: Extracto de código. Función para contar interrupciones del anemómetro.	64
Figura 37: Circuito de conexión de la pantalla LCD con módulo I2C.....	65
Figura 38: Extracto de código. Comandos de inicio LCD.	65
Figura 39: Circuito de conexión del módulo adaptador tarjeta micro SD.	67
Figura 40: Extracto de código. Comprobación de conexión con la tarjeta micro SD.....	67
Figura 41: Circuito de conexión del RTC.	68
Figura 42: Extracto de código. Configuración del RTC.	69
Figura 43: Parte superior de la carcasa de la Arduino Mega.	70
Figura 44: Parte inferior de la carcasa de la Arduino Mega.....	70
Figura 45: Diseño 3D de la carcasa del sensor MQ-135.....	71
Figura 46: Diseño 3D de la carcasa del sensor HC-SR07.....	71
Figura 47: Diseño 3D de la carcasa del sensor ML8511.....	72
Figura 48: Diseño 3D de la carcasa del sensor BMP280.	72
Figura 49: Diseño 3D de la carcasa del sensor HD-38.	73
Figura 50: Diseño 3D de la parte inferior del FC-03.	73
Figura 51: Diseño 3D de la parte superior del FC-03.	74
Figura 52: Diseño 3D de la carcasa de la LCD.	74
Figura 53: Diseño 3D de la carcasa del módulo para tarjeta micro SD.	75
Figura 54: Diseño 3D de la carcasa del RTC.	75
Figura 55: Estructura del pluviómetro.	76
Figura 56: Eje unido al tubo mediante los rodamientos.....	77
Figura 57: Estructura del anemómetro.	78
Figura 58: Caja estanca con la plancha de metacrilato.	79
Figura 59: Interior de la caja estanca con los componentes.	80

Figura 60: Pantalla LCD y botones de inicio y parada.	81
Figura 61: Extracto del archivo generado en formato <i>json</i>	83
Figura 62: Gráfica comparativa de la temperatura de los sensores BMP280 y DHT22.	84
Figura 63: Gráfica del registro humedad relativa del aire.	85
Figura 64: Velocidad del viento el día de la medición [57].	86
Figura 65: Gráfica del registro de la velocidad del viento.	86
Figura 66: Registro radiación UV.	87

Índice de tablas

Tabla 1: Sumario de las magnitudes a medir.	25
Tabla 2: Características principales de Arduino Mega 2560 [30].	32
Tabla 3: Características relevantes del sensor DHT22 [33].	34
Tabla 4: Características relevantes del sensor MQ-135 [36].	35
Tabla 5: Características relevantes del sensor HC-SR04 [38].	37
Tabla 6: Características relevantes del sensor ML8511 [40].	38
Tabla 7: Características relevantes del sensor BMP280 [43].	39
Tabla 8: Características relevantes del sensor HD-38 [45].	41
Tabla 9: Características relevantes del sensor FC-03 [48].	42
Tabla 10: Características relevantes LCD con módulo I2C [50].	43
Tabla 11: Características relevantes módulo micro SD [52].	44
Tabla 12: Características relevantes del RTC DS1307 [55].	45
Tabla 13: Códigos de error.	50
Tabla 14: Código de colores de los cables utilizados.	53

1. OBJETIVOS

1.1. OBJETIVOS TÉCNICOS

El objetivo técnico del TFG es llevar a cabo el prototipo de una estación meteorológica de bajo coste para el entorno agrícola en las Islas Canarias.

Se trata de diseñar y elaborar un prototipo funcional que permita la medición de distintos parámetros de interés mediante una serie de sensores, la recolección de los valores medidos y su posterior tratamiento para mostrar, de forma útil e intuitiva, la evolución de las condiciones agroclimáticas de una explotación agraria concreta. De este modo, se pretende contribuir en el desarrollo de la agricultura 4.0, en la que, mediante el uso de la tecnología, es posible incrementar la productividad y disminuir las repercusiones medioambientales que conlleva la agricultura.

Otro de los objetivos es dar cobertura al TFG titulado “Aplicación fullstack Mi Finquita”, del compañero Javier Martín, estudiante del Grado en Ingeniería Informática. Una vez recopilados los datos mediante la estación meteorológica, serán introducidos en dicha aplicación, que permitirá la visualización de los valores recogidos facilitando la gestión de los cultivos.

1.2. OBJETIVOS ACADÉMICOS

El objetivo académico del TFG es desarrollar de forma integrada las distintas competencias de la titulación: específicas, generales, transversales y básicas.

Se pretende llevar a cabo un proyecto en el ámbito de las tecnologías de la Ingeniería Industrial, en el que se sinteticen e integren las competencias adquiridas. Durante el desarrollo del proyecto se pondrán en práctica los conocimientos relativos a la realización de mediciones, cálculos y valoraciones, resolviendo problemas con iniciativa y creatividad, aplicando el razonamiento crítico y las habilidades desarrolladas en el campo de la Ingeniería Electrónica Industrial.

Además, se ejercitará la capacidad de organización y planificación, potenciando las habilidades de trabajo en equipo.

2. ALCANCE

Actualmente, está surgiendo la necesidad de poseer un mayor control sobre los aspectos que afectan a la producción de las explotaciones agrícolas y la calidad de los cultivos. Para ello, es necesario aportar soluciones técnicas que permitan a los agricultores obtener los valores de las magnitudes que influyen en el desarrollo de sus cultivos, conociendo mejor el estado de los mismos y facilitando la toma de decisiones en su gestión. Con ello, se conseguirá optimizar el consumo de recursos tales como el agua de riego o fertilizantes, impulsando una agricultura más sostenible y modernizada.

3. ANTECEDENTES

En las Islas Canarias existen varias plataformas y servicios que permiten visualizar los datos recogidos por distintas estaciones meteorológicas que se encuentran repartidas en el territorio insular. Ejemplos de ello en la isla de Tenerife son Apalmet o AgroCabildo.

En el caso de AgroCabildo, se trata de un servicio cuyo objetivo es formar e informar a las personas del medio rural. A través de sus estaciones meteorológicas se miden distintos parámetros dependiendo del tipo de estación, obteniendo medidas diarias y medias mensuales de los distintos parámetros. Los datos recogidos y la localización exacta de las estaciones son de uso público.

También cabe destacar la existencia de GRAFCAN, que es una empresa pública de la Comunidad Autónoma de Canarias, que gestiona el conocido como Sistema de Observación Meteorológica de Canarias, con una amplia red de estaciones equipadas con numerosos sensores, permitiendo la visualización y el acceso a los datos recogidos en las islas.

Los factores que distinguen la estación que se pretende llevar a cabo de las ya existentes en las Islas Canarias son, en primer lugar, el bajo coste de las mismas. Los componentes de la estación posibilitan la obtención de medidas fiables mediante sensores de bajo coste. En segundo lugar, no sería necesario depender de estaciones que se encuentren a mayor o menor distancia de la explotación agrícola de interés, sino que surge la posibilidad de tener una estación propia que permita obtener todas las variables necesarias para gestionar adecuadamente los cultivos.

Para poder desarrollar el prototipo de la estación meteorológica se ha realizado un estudio previo de las magnitudes a medir, buscando las soluciones técnicas para cuantificar dichas magnitudes. En el proceso de decisión de los parámetros que medirá el prototipo es de vital importancia conocer las necesidades del usuario. Por ello, se ha acudido al Servicio Técnico de Agricultura y Desarrollo Rural del Cabildo de Tenerife, poniéndonos en contacto

con Clemente Méndez Hernández, Agente Especialista en Cultivos Subtropicales. Se ha consultado qué magnitudes se necesita que mida una estación meteorológica enfocada hacia el entorno agrícola y, en base a eso, se ha decidido que el prototipo mida una serie de magnitudes concretadas a lo largo del proyecto. Algunas magnitudes de utilidad que no se han incluido entre las funcionalidades de la estación, se han desarrollado como futuras mejoras de la misma.

4. MAGNITUDES A MEDIR

4.1. TEMPERATURA DEL AIRE

La temperatura atmosférica es uno de los elementos más importantes del clima y se trata de una magnitud que refleja el nivel de calor del aire en un lugar y un momento concretos, resultado del intercambio de calor entre la atmósfera y la tierra [1]. Su unidad de medida en el Sistema Internacional es el Kelvin (K).

La capa de la atmósfera que está en contacto con la superficie terrestre es la troposfera. Dicha capa posee 10 km de altura, a lo largo de los cuales, la temperatura desciende aproximadamente 6 °C cada kilómetro. Además, en la troposfera tiene lugar la formación de nubes y la precipitación [2].

Utilidad de medir la temperatura del aire

Medir y conocer la temperatura del aire de una zona o una localización concreta es vital para el desarrollo de los cultivos. A través de la medición de la temperatura se puede determinar la cantidad de energía que necesitan los cultivos para poder germinar y desarrollarse. Dicha cantidad de energía se conoce como grados-día y con ella se puede determinar la influencia que ejerce la temperatura en el crecimiento de los cultivos y otros seres vivos, como los insectos. De esta forma se podrán predecir las etapas fenológicas de los cultivos, ya que la cantidad de energía que requieren para desarrollarse depende de su estado fenológico.

Se debe tener en cuenta que el nivel de actividad de las plantas depende de la temperatura a la que se encuentran, según cada especie de planta, teniendo su mayor índice de actividad a un nivel de energía llamada temperatura óptima. El metabolismo de las plantas funciona cuando la temperatura a la que se encuentran está situada entre su nivel mínimo o temperatura base y su temperatura máxima [1][3].

Gracias a la medición de la temperatura podremos saber cuándo y dónde tienen lugar sucesos de temperaturas extremas, de forma que podemos determinar los niveles de riesgo de heladas o agobio calórico de las plantas.

Además, la temperatura, junto con otros parámetros, se utiliza en diversos métodos de cálculo de la evapotranspiración de referencia (ET_o), permitiendo determinar las necesidades de riego de los cultivos. Algunos de los métodos más utilizados son el método de Hargreaves, el método de Blaney-Criddle o el método de Penman-Monteith [4][5].

4.2. PRESIÓN ATMOSFÉRICA

La presión atmosférica es la fuerza ejercida por el aire de la atmósfera en la Tierra. Concretamente, se trata de la presión que se ejerce en un punto por la columna de aire desde dicho punto hasta la parte superior de la atmósfera. Su unidad de medida en el Sistema Internacional es el Pascal (N/m²).

De esta forma, a mayor altura, menor presión, ya que el volumen de aire que ejerce la presión disminuye. A nivel del mar la presión atmosférica media es de 101325 Pa y a medida que aumenta la altitud, esta presión disminuye, por lo que es muy relevante conocer la altura a la que se mide la presión atmosférica [2][6]. Aun así, la presión atmosférica no es igual para puntos de igual altitud ya que, debido a los distintos procesos atmosféricos, la concentración de las moléculas que conforman el aire de la atmósfera varía de unas zonas a otras. Cuando se tiene una mayor concentración de moléculas de aire, se tienen altas presiones, que es lo que se denomina Anticiclón. Con una concentración menor, se tienen bajas presiones, que en meteorología se define como Borrasca [7].

Otro factor que afecta a la presión atmosférica es la temperatura. Con una mayor temperatura se tendrá menor presión y viceversa, a menor temperatura, mayor será la presión atmosférica [8].

Utilidad de medir la presión atmosférica

La presión es fundamental para favorecer el crecimiento y el desarrollo de un cultivo. Las zonas que poseen una mayor presión atmosférica suelen ser aquellas en las que se producen menos precipitaciones. Por otra parte, en las zonas de menor presión el intercambio de gases del cultivo con la atmósfera también es menor y, por tanto, las plantas disponen de menor cantidad de dióxido de carbono, necesario en el proceso de fotosíntesis [8].

La presión atmosférica condiciona la previsión meteorológica, de tal forma que se puede prever la ocurrencia de sucesos meteorológicos como tormentas y precipitaciones, siendo de gran utilidad para el cuidado de los cultivos [6].

4.3. HUMEDAD

4.3.1. Humedad del Aire

La humedad es la cantidad de vapor de agua que se encuentra en un volumen de aire determinado. La humedad del aire se puede expresar en términos de humedad absoluta, humedad específica y humedad relativa, siendo esta última la más utilizada para expresar la humedad [2].

La humedad absoluta es la masa de vapor de agua que contiene una unidad de volumen de aire, expresándose en g/m^3 . La humedad específica es la masa de vapor de agua por unidad de masa de aire, expresada en g/kg . La humedad relativa es la relación entre la presión parcial del vapor del agua y la presión de vapor saturante para la misma temperatura. La presión parcial del vapor de agua corresponde con la humedad absoluta del aire a cierta temperatura y la presión de vapor saturante es la máxima cantidad de vapor de agua contenida en cierto volumen de aire para la misma temperatura. La humedad relativa se expresa en % [2][9].

Para medir la humedad relativa se utilizan mayormente los higrómetros de cabello, haciendo uso de pelos de personas o de crin de caballo debido a su sensibilidad ante cambios en la humedad. También se mide mediante otros higrómetros, como el de cloruro de litio, o mediante los psicrómetros [9][10].

Utilidad de medir la humedad relativa

Es importante conocer la humedad de la atmósfera ya que resulta determinante para la producción de las plantas, debiendo tener valores más o menos altos dependiendo de la especie.

Las mediciones de la humedad relativa, junto con la temperatura, sirven para prever la ocurrencia de heladas [3]. Además, combinada con distintas variables, la humedad relativa permite estimar la evapotranspiración de referencia, con lo que se puede estimar las necesidades de riego del cultivo.

Tener un control de la humedad relativa también es útil ya que la humedad atmosférica puede favorecer la aparición de enfermedades causadas por hongos o bacterias en las plantas [10].

4.3.2. Humedad del Suelo

La humedad del suelo es la cantidad de agua que se encuentra en un volumen determinado de tierra [11].

Dependiendo del tipo de suelo, el agua de la lluvia se drenará a capas más profundas a mayor o menor velocidad. De esta forma, cada tipo de suelo y sus diferentes capas tendrán una capacidad diferente para la retención de agua.

El agua se puede drenar rápidamente hacia capas inferiores si los espacios entre las partículas del suelo son muy grandes y, por tanto, esta agua no estará disponible para las plantas. Si los espacios entre las partículas son más pequeños, debido a la tensión superficial del agua, ésta se ve retenida y queda a disposición de las plantas. El agua de los distintos compuestos orgánicos que forman el suelo y el agua adherida a las partículas del suelo no pueden ser absorbidas por las plantas. Es por ello por lo que la mayor parte del agua que pueden absorber las plantas es aquella que queda retenida por la tensión superficial [12].

La humedad adecuada del suelo dependerá sobre todo de la especie de la planta, ya que cada especie requiere una cantidad de agua diferente. Además, se pueden definir una serie de regímenes de humedad del suelo en función del nivel alcanzado del manto freático y de la existencia de agua retenida a una presión determinada [11].

Utilidad de medir la humedad del suelo

Conocer las condiciones de humedad en el suelo es de gran importancia ya que el recurso hídrico en el suelo es determinante para el correcto desarrollo de los cultivos, siendo el agua el elemento que condiciona las características de productividad y fertilidad del suelo [13].

El grado de humedad del suelo es un factor que ayuda a saber cuándo es el mejor momento para sembrar y para regar, gestionando mejor los recursos y propiciando unas mejores condiciones para el crecimiento de las plantas [11].

4.4. VIENTO

El viento es el movimiento de masas de aire debido a las corrientes convectivas que tienen lugar en la atmósfera terrestre, concretamente en la troposfera [14]. Debido a la energía térmica que se recibe del sol, que produce diferencias térmicas, y la reducción de la temperatura a mayores altitudes, se produce una transformación de energía térmica a energía cinética. Dicha

transformación se traduce en el desplazamiento del aire de forma vertical y horizontal. Por lo general, son de mayor magnitud los movimientos horizontales [2].

Además, las diferencias de presión en la atmósfera producen movimientos de aire desde las zonas de alta presión hacia las zonas de bajas presiones. También es un factor relevante la actuación de las fuerzas producidas por la rotación de la tierra, que influyen considerablemente en la dirección del viento [14].

El viento se caracteriza principalmente por su velocidad y su dirección. La velocidad del viento se mide en m/s o en km/h, y para ello se utiliza un anemómetro. La dirección del viento es el punto en el horizonte desde donde viene, indicándose mediante los puntos cardinales (N, S, E, W, NE, NW, SE, SW). La dirección del viento se mide utilizando una veleta [3].

Utilidad de medir la dirección y velocidad del viento

Tanto la dirección como la velocidad del viento son parámetros imprescindibles para el cálculo de la evapotranspiración de los cultivos.

Medir los parámetros relativos al viento es importante debido a la erosión que éste produce en los suelos, así como a los daños que puede causar a los cultivos según las velocidades que alcance (daños en las flores o en los frutos, entre otros) [3].

Controlar la velocidad y dirección del viento permitirá planificar de forma óptima la aplicación de tratamientos fitosanitarios, además de conocer el recorrido de posibles elementos beneficiosos o dañinos para las plantas y el transporte del polen [3][15].

Por último, puede utilizarse esta información para diseñar y llevar a cabo la instalación de cortinas o elementos cortavientos que protejan a los cultivos de posibles daños [3].

4.5. CALIDAD DEL AIRE

La contaminación atmosférica o también llamada contaminación del aire es la presencia y acumulación de sustancias y gases en aire, las cuales pueden producir un riesgo o daño para seres vivos. Así pues, la calidad del aire se puede representar con un índice mediante el cual se reflejan las cantidades de contaminantes presentes en el aire.

Utilidad de medir la calidad del aire

Las plantas tienen una gran sensibilidad a la contaminación, sufriendo enfermedades y daños ante la presencia de contaminantes en el aire. Los efectos se pueden manifestar en los tejidos de la planta o en sus funciones metabólicas, pudiendo tener consecuencias fatales para ellas.

El principal gas que puede producir enfermedades es el dióxido de azufre (SO₂), tanto por una concentración mayor como por una de menor nivel. Una exposición menor durante tiempo largos puede producir un amarillamiento de la hoja. En cambio, si se produce una concentración mayor de los niveles de SO₂, se puede llegar a generar una necrosis de color rojo o anaranjado.

Otros compuestos bastante dañinos para las plantas son el monóxido de carbono (CO) y los óxidos de nitrógeno (NO_x), en especial el dióxido de nitrógeno (NO₂), que puede producir necrosis y clorosis de color negro o marrón en las plantas [16].

4.6. RADIACIÓN SOLAR

La radiación solar es la energía del sol, realizada por medio de ondas electromagnéticas, la cual, se propaga en todas direcciones. Es una emisión de energía en ondas cortas, sufriendo un debilitamiento en la atmósfera. La superficie terrestre devuelve la radiación en forma de onda larga. La radiación solar se mide en W/m² y para ello se utiliza un piranómetro [17].

Hay diferentes tipos de radiación según su incidencia en la superficie terrestre. Pueden ser: solar directa, que es la radiación que no sufre difusión; solar difusa, que es la radiación que experimenta desviaciones en su trayectoria; y solar reflejada, que es la radiación en la que se produce el efecto albedo (se refleja con la superficie terrestre).

Sin embargo, también se puede clasificar según los tipos de rayos que contiene la radiación: infrarrojos, visibles, y ultravioletas, estos últimos los más dañinos para los cultivos.

Utilidad de medir la radiación solar

La radiación es una de las magnitudes más importantes para cualquier cultivo, por lo tanto, se tiene que realizar un seguimiento exhaustivo. Influye en procesos como la fotosíntesis, los balances de agua y energía, y el crecimiento y desarrollo del cultivo, procesos determinantes para el crecimiento y rendimiento potencial de las plantas.

Además, una radiación inapropiada para la plantación puede causar diferentes enfermedades, grietas en frutos y hoja enrollada, causados por una alta radiación, y maduración manchada o frutos huecos, producidos por una radiación inferior [18][19].

4.7. VOLUMEN DE PRECIPITACIONES

La precipitación es la caída desde la atmósfera hasta la superficie terrestre de gotas de agua o cristales de hielo. De esta manera, se consideran formas de precipitación la lluvia, la nieve o el granizo, y variantes como el aguanieve o la llovizna [3][20].

Las precipitaciones se originan en las nubes, cuando las gotas que contienen aumentan su tamaño y peso y acaban cayendo debido a la gravedad [20].

La cantidad de precipitación se mide en mm, que se corresponde con la altura a la que llega la precipitación sobre una superficie plana de paredes verticales. Si las precipitaciones son en forma de lluvia se suelen expresar en l/m^2 , es decir, los litros de agua contenidos en una superficie de un m^2 . Ambas unidades de medida son equivalentes, puesto que 1 mm es lo mismo que 1 l/m^2 . El instrumento de medida utilizado es el pluviómetro [21].

Utilidad de medir el volumen de precipitaciones

La medición del volumen de las precipitaciones en una zona determinada permite conocer el comportamiento de dichas precipitaciones, mediante la aplicación de procedimientos estadísticos con los distintos datos registrados. Esto se debe a que, a partir de los registros de varios parámetros relativos a las precipitaciones de una región, se pueden obtener modelos estadísticos de predicción relativos a la frecuencia y a la magnitud de las precipitaciones [20].

De esta forma, permitirá planificar el riego y el drenaje de los cultivos, economizando así el uso del agua, mejorando la producción y prediciendo daños físicos en los cultivos. También permite la planificación de las cosechas y las siembras, así como la utilización de fertilizantes [3].

4.8. SUMARIO DE MAGNITUDES

Tras analizar cada magnitud cuya medición resulta de interés para el proyecto, se ha llevado a cabo una recopilación de todas ellas, así como los instrumentos de medida que se emplean para cuantificar dichas magnitudes, su unidad de medida y una breve descripción de sus utilidades en el entorno agrícola. Esta recopilación se muestra en la Tabla 1:

VARIABLES A MEDIR	SENSOR DE LA ESTACIÓN	UNIDAD DE MEDIDA	UTILIDAD DE LA MEDIDA EN LOS CULTIVOS
Temperatura	DHT22 y BMP280	Grados (°C)	<ul style="list-style-type: none"> - Determinar la cantidad de energía que necesitan los cultivos. - Predecir etapas fenológicas. - Saber los sucesos de temperatura extrema. - Para el cálculo del método de la evapotranspiración de referencia (ET_o).
Humedad relativa	DHT22	Porcentaje (%)	<ul style="list-style-type: none"> - Prever la ocurrencia de heladas. - Estimar la evapotranspiración de referencia. - Evitar la aparición de enfermedades.
Presión atmosférica	BMP280	Milibar (mbar)	<ul style="list-style-type: none"> - Prever la ocurrencia de sucesos meteorológicos como tormentas y precipitaciones.

Velocidad del viento	Anemómetro con el sensor FC-03	Kilómetro por hora (km/h)	<ul style="list-style-type: none"> - Cálculo de la evapotranspiración de los cultivos. - Posibles daños en la vegetación y en el suelo. - Planificar la aplicación de tratamientos fitosanitarios. - Análisis de si es necesario una instalación de cortinas o elementos cortavientos.
Calidad del aire	MQ135	Partes por millón (ppm)	<ul style="list-style-type: none"> - Evadir la producción necrosis y clorosis de color negro o marrón en las plantas
Radiación solar	ML8511	Milivatios por centímetro cuadrado (mW/cm ²)	<ul style="list-style-type: none"> - Para evitar la aparición de enfermedades: grietas en los frutos, hoja enrollada, maduración manchada o frutos huecos. - Aumentar el rendimiento.
Volumen precipitaciones	Pluviómetro con el sensor HC-SR04	Litros por metro cuadrado (L/m ²)	<ul style="list-style-type: none"> - Obtener modelos estadísticos de predicción relativos a la frecuencia y a la magnitud de las precipitaciones.

Tabla 1: Sumario de las magnitudes a medir.

5. CARACTERÍSTICAS DE LOS ELEMENTOS DE LA ESTACIÓN

En este apartado, por una parte, se explicarán los dispositivos digitales, los entornos de programación y las comunicaciones necesarias para el correcto funcionamiento de los elementos de la estación. Por otra parte, se describirán las características de los sensores y el resto de los componentes hardware implementados en el diseño final, cuyas fichas técnicas se encuentran en el Anexo III.

5.1. SOPORTE DIGITAL Y COMUNICACIONES EN LA ESTACIÓN

5.1.1. Arduino

Arduino es una plataforma de hardware libre, donde una placa con un microcontrolador, a través de un circuito integrado, permite ejecutar órdenes grabadas en la memoria.

Para poder desarrollar correctamente el proyecto, el software utilizado es el facilitado por Arduino, basado en el entorno de desarrollo Arduino IDE, que nos permite desarrollar e interactuar con dispositivos tanto por hardware como por software.

Dicho entorno de programación consiste en: un editor de texto, un compilador, un depurador y un constructor de interfaz gráfica o GUI. Este entorno también nos permite agregar y actualizar librerías de forma rápida y sencilla, avisando de nuevas actualizaciones de librerías ya instaladas. Además, permite gestionar las tarjetas y placas oficiales de Arduino, pero además también da soporte a tarjetas no oficiales [22].

Las principales características que se han tenido en cuenta para la elección de Arduino para poder llevar a cabo nuestra estación han sido las siguientes:

1. Sus componentes tienen un bajo coste, siendo este factor muy importante debido a que el proyecto tiene como principal objetivo que las estaciones no requieran una gran inversión económica.
2. Permite la utilización de gran cantidad de módulos, sensores y actuadores, pudiendo realizar todo tipo de proyectos con numerosas aplicaciones.

3. Se programa con un lenguaje de programación basado en C++, que es un lenguaje de programación con el que se ha trabajado a lo largo del desarrollo del grado, por lo que estamos familiarizados con él.
4. Las placas de desarrollo basadas en Arduino están pensadas para funcionar con baterías. Aunque actualmente, el prototipo inicial no consta de dichas baterías, en un futuro se trataría de implementarlas para que además de que el dispositivo sea de bajo coste, tenga el menor impacto medioambiental, contribuyendo con la lucha contra el cambio climático.
5. Es de código abierto, es decir, de libre acceso. Los usuarios pueden utilizar los códigos y adaptarlos a los diferentes contextos de manera gratuita. Además, se promueve el intercambio de ideas y la colaboración para mejorar tus programas gracias a la amplia comunidad existente de Arduino.

5.1.2. Protocolos de comunicación

Para la conexión entre los distintos sensores y elementos utilizados y la placa de desarrollo Arduino Mega 2560, se han empleado distintos protocolos de comunicación que permiten transmitir los datos de los sensores a la placa de desarrollo. Estos protocolos de comunicación son: UART, I2C y SPI. A continuación, se concretan las características principales de cada protocolo:

Protocolo UART

El protocolo UART (recepción y transmisión asíncronas universales) es un protocolo de comunicación que posibilita la comunicación serie con los distintos dispositivos.

Este sistema de comunicación es asíncrono, como su propio nombre indica, por lo que no se necesita una señal de reloj para sincronizar la comunicación, sino que es la línea de datos la que tiene las señales de inicio y parada. Además, como la comunicación que se produce es comunicación serie, los datos se transmiten bit a bit.

El sistema UART permite la comunicación únicamente entre un maestro y un esclavo, no puede darse la comunicación multimaestro y multiesclavo y se realiza mediante dos líneas, RX y TX, siendo RX la línea para la recepción y la línea TX para la transmisión de datos. Habitualmente las líneas están cruzadas entre los dos dispositivos que van a comunicarse, estando la línea TX del dispositivo que transmite la información conectada a la línea RX del dispositivo receptor, y viceversa.

Antes de que el dispositivo transmisor envíe datos a través del bus de datos, el dispositivo UART añade un bit de inicio, un bit de paridad y un bit de parada. Una vez transmitido el dato, el dispositivo receptor sustrae los bits de datos de los bits de inicio, paridad

y de parada. Mientras no se producen comunicaciones, el estado de las líneas se mantiene a 1, por lo que, al iniciar la comunicación, el bit de inicio es 0. Para indicar el fin de las comunicaciones, el bit de parada está a alta. El bit de paridad sirve para detectar errores en la comunicación.

En nuestra placa de desarrollo de Arduino, la comunicación UART tiene lugar conectando la placa con el otro dispositivo mediante los pines TX y RX. Si conectamos nuestra placa de desarrollo de Arduino con otro PC a través de su puerto USB, la comunicación también tiene lugar mediante el sistema UART [23].

Protocolo I2C

La comunicación I2C (*Inter-Integrated Circuit*) es otro tipo de comunicación serie en la que la transferencia de datos se realiza mediante un único cable. Es de tipo síncrono, por lo que requiere de una señal de reloj, y puede tener varios maestros y/o esclavos.

La placa de desarrollo Arduino necesita solo dos pines para establecer la comunicación mediante I2C y únicamente uno de ellos para transmitir datos. Estos pines son el SDA (*Serial Data*) y el SCL (*Serial Clock*). El pin SDA es el que se encarga de la transmisión de datos entre el maestro y el esclavo, mientras que el pin SCL comparte la señal de reloj, que controla el maestro, entre éste y el esclavo.

Para iniciar la comunicación, la línea SDA pasa de alta a baja cuando la línea de reloj SCL está a alta, mientras que, para finalizar la comunicación, la línea SDA pasa de baja a alta con la línea SCL a alta. De esta forma, siempre que se va a iniciar o a detener la comunicación, la línea SDA cambia de estado con SCL a alta. Durante el resto de la comunicación, los cambios de estado de SDA tienen lugar con la línea de reloj a baja.

Después de la condición de inicio de la comunicación, se envía una serie de bits que sirve como identificador de los esclavos, de manera que el maestro solo se comunica con el esclavo cuya identificación coincide con la que se ha enviado. Tras el identificador, el maestro envía un bit que indica que va a enviar datos al esclavo o si va a solicitarlos. A continuación, el dispositivo receptor envía un bit a 0 para indicar que los datos se han recibido con éxito. Después de comprobar que se está recibiendo con éxito, se envían los bits de datos, cuya longitud es siempre de 8 bits. Al finalizar el envío de bits de datos, se vuelve a enviar el bit de comprobación que indica si éstos se recibieron adecuadamente.

Debido a los bits de comprobación, este tipo de comunicación tiene una sobrecarga de datos, por lo que la velocidad a la que se transfieren los datos es más lenta en comparación con el protocolo SPI [24].

Protocolo SPI

El protocolo de comunicación SPI (*Serial Peripheral Interface*) es otro de los tipos de comunicación serie síncrono, por lo que también es necesario una línea de reloj para realizar la

transferencia de datos. Este tipo de comunicación es más rápido en comparación con los anteriormente descritos, I2C y UART.

El protocolo SPI no permite la existencia de varios maestros, es un único maestro el que controla la comunicación con varios esclavos. Para establecer la comunicación se necesitan 4 líneas. Las líneas MISO (*Master In Slave Out*), MOSI (*Master Out Slave In*) y SCK o SCLK conectan al dispositivo maestro con los dispositivos esclavos, mientras que la línea SS o CS es una línea específica que conecta cada esclavo con el maestro sin necesidad de un identificador. La línea MISO permite enviar datos del esclavo al maestro, la línea MOSI permite enviar datos del maestro al esclavo, la línea SCK es la línea de reloj controlada por el maestro y, por último, la línea SS permite que el maestro active o desactive las comunicaciones con cada esclavo.

Para que se establezca la comunicación SPI, se suceden una serie de pasos. En primer lugar, el maestro realiza la configuración de la transmisión de datos, la frecuencia del reloj y el modo de transmisión. Mediante la línea SS el maestro indica con qué esclavo va a comunicarse. Después el maestro y el esclavo utilizan las líneas MISO y MOSI para enviarse entre ellos los bits de datos. Al finalizar la transmisión de todos los bits de datos, se realiza una copia de los datos en el buffer de recepción. La comunicación se detiene cuando el maestro lo indica mediante la línea SS e interrumpe la señal de reloj.

A diferencia del protocolo I2C, en las comunicaciones SPI no se requiere de un identificador único para los esclavos y tampoco se tiene ningún protocolo para realizar comprobación de errores en la comunicación [25].

5.1.3. SketchUp

La estación meteorológica contendrá distintos componentes que servirán para medir las variables de interés, guardar los datos medidos y mostrar los resultados al usuario. Con el objetivo de proporcionar cierto grado de protección a dichos elementos, se han diseñado carcasas a medida para cada uno de ellos. Estas carcasas, obtenidas mediante impresión 3D con filamento de PLA, se han diseñado mediante el software *SketchUp Make 2017*. También se ha utilizado dicho software para diseñar y elaborar el anemómetro de la estación.

La razón de la utilización de este programa es que se trata de un software que permite realizar diseños 3D de forma sencilla, es fácil de aprender a utilizar desde cero, permite importar y exportar archivos en numerosos formatos, tanto modelos en 2D como modelos en 3D, y, además, su versión PRO tiene un periodo de prueba gratuita de 30 días. Dicha versión es la que nos ha permitido diseñar los distintos elementos en 3D, exportando los modelos en formato “.obj”, pudiendo realizar más adelante una conversión al formato de impresión con buena calidad en los resultados [26].

5.1.4. Ultimaker Cura

Después realizar los diseños a medida, tanto para las carcasas de los sensores de la estación meteorológica como para el propio anemómetro, para convertir el formato “.obj” que se ha obtenido de *SketchUp Make* al formato de impresión “.gcode” (que es con el que trabajan las impresoras 3D) se ha hecho uso del software *Ultimaker Cura*, especializado en impresión 3D.

Dicho software posee múltiples funcionalidades que nos permitirán ajustar los distintos parámetros de impresión. En primer lugar, escogeremos la impresora de la cual se va a hacer uso para imprimir. Nos permite escoger el modelo exacto, ya que, de esta forma, el programa establecerá ciertas características ya definidas, como son las dimensiones de la propia impresora. En nuestro caso, las impresoras que se han utilizado son la BQ Hephestos 2 y la Creality CR-10S, disponibles para su uso en la BiblioLab de la Escuela Superior de Ingeniería y Tecnología de la ULL.

También podremos escoger el tipo de material con el que se realizará la impresión, que en nuestro caso es PLA. Además, permite efectuar cambios en los ajustes de impresión de forma sencilla, como por ejemplo la altura de las capas, el grosor de las paredes, el porcentaje y tipo de relleno, la temperatura de calentamiento del material, la velocidad de la impresión, las características de los soportes generados (utilizados para sostener y poder imprimir las partes sin apoyos), entre otros.

Una vez realizadas todas las configuraciones se procesa el diseño y el software nos indica el peso aproximado del objeto que se ha impreso, así como la duración de la impresión. Antes de guardar en el formato adecuado, podemos acudir a la vista previa para ver cómo se va a imprimir nuestro diseño. Esto es bastante útil para comprobar que la impresora realizará correctamente la impresión. Una última opción que permite llevar a cabo este software es monitorizar la impresión conectando la impresora a nuestro PC [27].

5.1.5. Fritzing

Son numerosos los elementos que se han utilizado para poder llevar a cabo la estación meteorológica, por lo que se han realizado diversas conexiones entre los sensores y módulos con la placa de desarrollo Arduino Mega 2560. Para representar las conexiones del circuito de la estación se ha utilizado el software gratuito *Fritzing* [28].

Mediante dicho software, se ha representado la conexión de cada sensor y elemento con la protoboard y la placa de desarrollo. El objetivo es facilitar el montaje de la estación meteorológica y que ésta se pueda replicar de forma más sencilla en el futuro.

5.2. ELEMENTOS HARDWARE QUE COMPONEN LA ESTACIÓN

5.2.1. Placa de desarrollo Arduino Mega 2560

La placa de desarrollo a utilizar es la Arduino Mega 2560 (Fig. 1). La principal característica que se tuvo en cuenta para la elección de la misma, frente a la Arduino UNO, es el mayor número de entradas y salidas disponibles, ya que tiene 54 pines digitales y 16 entradas analógicas.

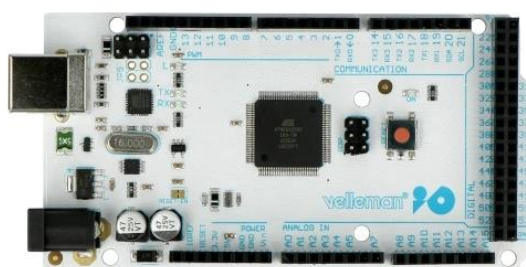


Figura 1: Arduino Mega 2560 [29].

Además de tener un mayor de entradas y salidas, la Arduino Mega 2560 cuenta con un tamaño mayor de las diferentes memorias: FLASH 256 KB, en la que se almacena el código, SRAM 8 KB, que almacena las variables, equivalente a las RAM de los ordenadores y EEPROM 4 KB, en la que se almacena la información que debe mantenerse, aunque la placa no esté alimentada.

A continuación, se muestra la Tabla 2, realizada a modo de resumen de otras características relevantes de la placa de desarrollo Arduino Mega 2560, pudiendo encontrar más información en la ficha técnica de la placa de desarrollo, contenida en el documento Anexos.

Microcontrolador	ATmega2560
Tensión de entrada recomendada	7 - 12 VDC
Nivel lógico	5 V
Resolución	0 a 1023 (10 bits)
Puertos Seriales	4
Corriente del pin de 3,3 V	50 mA
Interrupciones	5

Velocidad del reloj	16 MHz
---------------------	--------

Tabla 2: Características principales de Arduino Mega 2560 [30].

Como ya se ha mencionado, la placa de desarrollo posee numerosos pines digitales, analógicos, pines destinados a las comunicaciones serie mediante UART, I2C y SPI, etc. Es esencial conocer la función que tiene cada uno de estos pines, y para ello se ha hecho uso del esquema del patillaje que se muestra a continuación en la Figura 2:

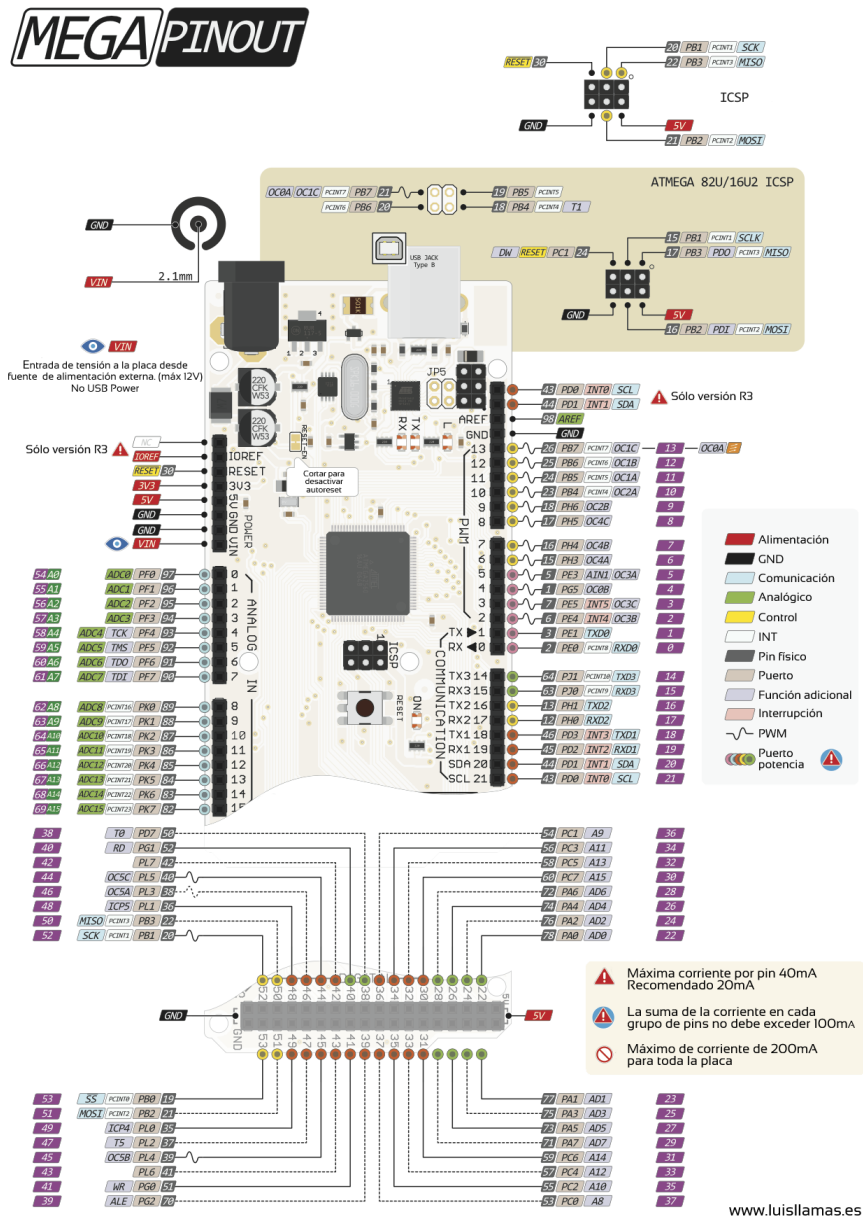


Figura 2: Esquema patillaje (*pinout*) Arduino Mega 2560 [31].

Finalmente, se mencionan las dos formas que se han utilizado para alimentar la placa de desarrollo Arduino Mega 2560:

1. Mediante USB: Es la forma de alimentación más empleada en el proyecto, dado que es la que nos permite conectar la Arduino Mega con el ordenador. Únicamente es capaz de proporcionar 5 V y contiene un fusible PPTC que limita la corriente que pueden consumir la placa y los componentes.
2. Mediante jack: la alimentación mediante *jack* tiene lugar por medio de un adaptador de corriente AC/DC. En esta entrada, la Arduino dispone de un regulador cuya función es asegurarse de proporcionar un voltaje adecuado a la placa. Para el correcto funcionamiento, el voltaje de entrada tiene que encontrarse entre 7 V y 12 V. Para el proyecto se ha adquirido una alimentación de 9 V para asegurar que no se sobrecalienta el regulador [32].

5.2.2. Sensor DHT22

El sensor DHT22 es un sensor digital de temperatura y humedad relativa del aire (Fig. 3). Se trata de un sensor de fácil utilización, con una buena fiabilidad en las medidas que recoge y de bajo coste.

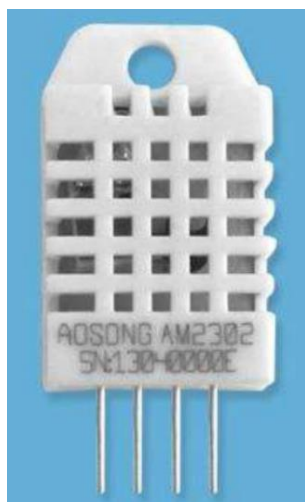


Figura 3: Sensor DHT22 [33].

La principal característica por la cual dicho sensor es idóneo es que no necesita calibración, debido a que se han guardado de fábrica unos coeficientes de calibración en la memoria OTP (*One-Time-Programmable*) del sensor, permitiendo obtener datos de gran

fiabilidad. Asimismo, otra cualidad relevante es que posee un mayor rango de medición de la temperatura en comparación de su homólogo inferior (DHT11), pudiendo medir temperaturas desde $-40\text{ }^{\circ}\text{C}$ hasta $80\text{ }^{\circ}\text{C}$. Además, es más preciso y permite una mayor resolución. Para el correcto funcionamiento del sensor, se recomienda la utilización de una resistencia de $10\text{ k}\Omega$ para filtrar el posible ruido externo [34].

A continuación, se muestra la Tabla 3, donde se exponen las características relevantes del sensor DHT22.

Voltaje de operación	3 V a 6 V
Resolución Temperatura	0,1 $^{\circ}\text{C}$
Resolución Humedad	0,1 % RH
Rango de medición de humedad	0 a 100 % RH
Tiempo de sensado	2 s

Tabla 3: Características relevantes del sensor DHT22 [33].

5.2.3. Sensor MQ-135

El sensor MQ-135 es un sensor de gas cuya utilidad es la detección de la presencia de gases en aire, así como la medición de la concentración de los mismos. El sensor que se ha adquirido se encuentra dispuesto como un módulo precableado, tal y como se muestra en la Figura 4. Se trata de un sensor electroquímico, por lo que su resistencia cambia ante la presencia de ciertos gases en el ambiente.



Figura 4: Sensor MQ-135.

Es posible utilizar el sensor como digital o analógico. La conexión digital permitirá obtener una señal cuando la lectura del sensor alcance cierto valor. El valor de las lecturas a partir del cual el sensor activa la línea digital se ajusta con el potenciómetro contenido en el módulo del sensor. La conexión analógica permitirá obtener la lectura del sensor de la concentración del conjunto de gases detectables por el mismo en el aire, que se expresa en ppm (partes por millón).

Este sensor es sensible a la presencia de numerosos gases, entre los que se encuentran: Alcohol, Benceno, Tolueno, Acetona, Humo, Dióxido de carbono (CO₂), Monóxido de carbono (CO), Óxidos de nitrógeno (NO_x), Amonio (NH₄), Amoníaco (NH₃). Varios de ellos son nocivos para el desarrollo de las plantas [35].

En la Tabla 4 se exponen algunas de las principales características del sensor MQ-135, recogidas en la ficha técnica del mismo:

Voltaje de operación	5 V
Temperatura de uso	-10 a 45 °C
Humedad relativa del ambiente	< 95 %RH
Resistencia de detección (R _s)	30 kΩ - 200 kΩ (100 ppm NH ₃)
Rango de detección	10 - 1000 ppm
Tiempo de calentamiento	±24 h

Tabla 4: Características relevantes del sensor MQ-135 [36].

A continuación, se muestra la curva característica de la sensibilidad del sensor MQ-135 ante la presencia de varios gases (Fig. 5). Esta gráfica representa la variación de la relación

R_s/R_o frente a las partes por millón que detecta el sensor, siendo R_o la resistencia del sensor ante 100 ppm de NH_3 en el aire limpio. De esta forma, en el aire limpio R_o siempre tendrá el mismo valor que R_s . Esto queda reflejado en la gráfica por la línea azul de referencia.

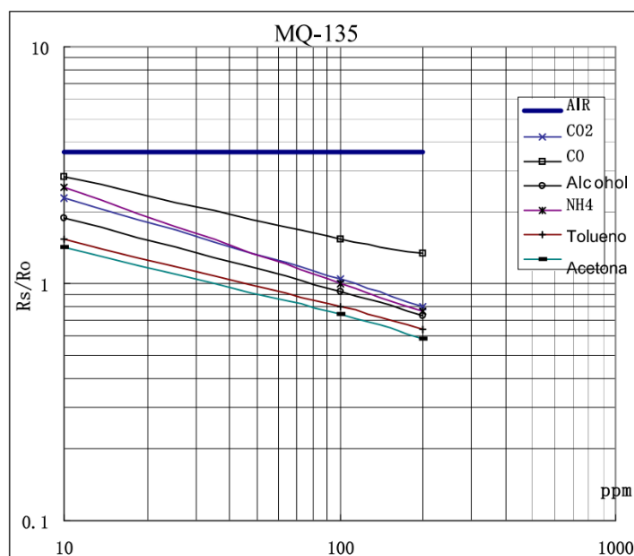


Figura 5: Curva característica de la sensibilidad del MQ-135 [36].

5.2.4. Sensor HC-SR04

El sensor HC-SR04 (Fig. 6) es el sensor de ultrasonidos seleccionado para el proyecto con el fin de determinar el volumen de precipitaciones. Dicho sensor está conformado por dos transductores: un emisor y un receptor. El sensor emite un sonido ultrasónico a través del emisor, y espera hasta que el sonido rebote en un objeto para ser recogido por el otro transductor, el receptor. A través de la medición del periodo de tiempo transcurrido desde que el sonido fue enviado hasta que fue recibido, se puede obtener la distancia entre el sensor y el objeto de manera sencilla.



Figura 6: Sensor HC-SR04.

Para la elaboración de la estación meteorológica, se ha tenido en cuenta la alta presión que posee el sensor, de 3 mm. Esto es de mucha utilidad, debido a que el sensor se va a utilizar para la medición de la altura de agua en un recipiente, lo que nos permitirá calcular las precipitaciones acumuladas. Además, es un dispositivo muy barato en comparación con otros sensores capaces de medir la distancia [37].

Voltaje de Operación	5 V
Rango de medición	2 a 450 cm
Duración mínima del pulso de disparo	10 μ s
Tiempo de sensado	50 ms
Frecuencia de ultrasonido	40 kHz

Tabla 5: Características relevantes del sensor HC-SR04 [38].

5.2.5. Sensor ML8511

Una de las variables que va a registrar nuestra estación meteorológica es la radiación solar. Para ello, se ha decidido utilizar el sensor ML8511 (Fig. 7), capaz de detectar la radiación ultravioleta (UV).

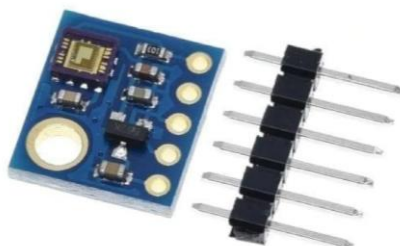


Figura 7: Sensor ML8511 [39].

En la siguiente tabla se han recogido algunos de los parámetros principales que caracterizan al sensor ML8511, siendo muy importante conectar el sensor a la tensión de operación indicada, sin sobrepasar los límites de 2,7 V como valor mínimo y 3,6 V como valor máximo.

Voltaje de operación	3,3 V
Salida de voltaje analógica lineal	1,0 V a 3,0 V
Longitud de onda detectada	280 a 390 nm
Temperatura de operación	-20 °C a 70 °C

Tabla 6: Características relevantes del sensor ML8511 [40].

El sensor convierte la corriente fotoeléctrica en tensión, entregando una señal analógica de voltaje que es directamente proporcional a la cantidad de luz UV detectada [39].

A continuación, en la Figura 8, se muestra la gráfica del sensor que representa el voltaje de salida frente a la intensidad UV, a distintas temperaturas, para una longitud de onda de 365 nm. Esta característica es de mucha utilidad para la calibración del sensor y de uso necesario para la programación del mismo. Con ella podemos conocer la tensión de salida que proporcionará el sensor para un nivel determinado de radiación ultravioleta en mW/cm^2 .

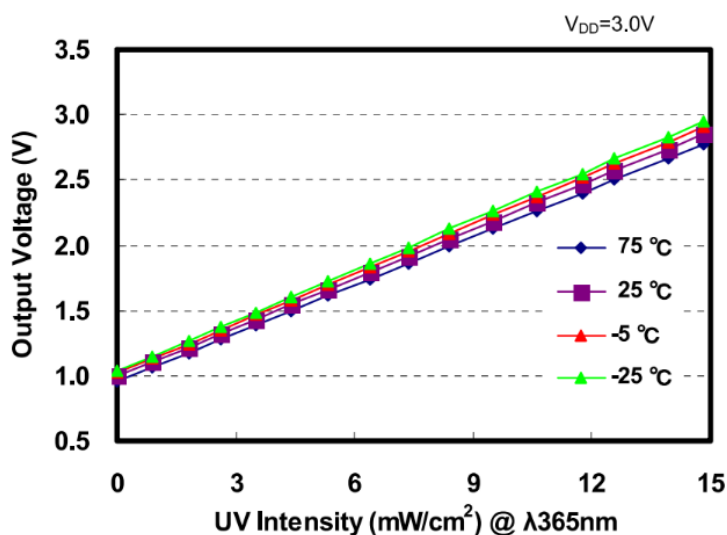


Figura 8: Tensión de salida frente a la intensidad UV [40].

5.2.6. Sensor BMP280

El sensor BMP280 (Fig. 9) mide la presión atmosférica y la temperatura, permitiendo obtener el valor de altitud sobre el nivel del mar a partir de la presión atmosférica registrada. Se basa en la tecnología *Bosch Sensortec*, lo que permite obtener medidas con gran precisión y

estabilidad. Además, posee una buena compatibilidad electromagnética, lo que protege al sensor de la influencia de campos electromagnéticos.

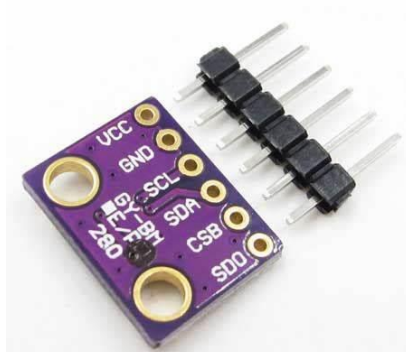


Figura 9: Sensor BMP280 [41].

Se ha escogido dicho sensor sobre su homólogo inferior, el BMP180, por su mejora en la precisión tanto para la temperatura como para la presión, por la disminución en el tamaño y por la reducción de 12 μA a 2,7 μA en el consumo. Otra ventaja es que, además de incorporar la comunicación por I2C, esta nueva versión implementa la SPI, siendo idónea para futuras mejoras de la estación, permitiendo la conexión de varios sensores iguales [42].

Voltaje de operación	1,8 V a 3,3 V
Rango presión	300 a 1100 hPa
Resolución presión	0,12 hPa
Precisión presión	1 hPa
Rango de temperatura	-40 °C a 85 °C
Resolución temperatura	0,01 °C
Precisión temperatura	1 °C
Tasa de medida	Hasta 157 Hz

Tabla 7: Características relevantes del sensor BMP280 [43].

5.2.7. Sensor HD-38

El sensor HD-38 (Fig. 10) es un sensor capaz de medir la humedad del suelo, lo que permitirá al usuario de la estación meteorológica controlar adecuadamente el sistema de riego de un cultivo en función de las condiciones de humedad que tenga el suelo. Además, las medidas obtenidas del mismo contribuirán a la generación de modelos de recurso hídrico, los cuales son de mucha utilidad, no solamente para el riego, sino para la aplicación de tratamientos fitosanitarios y otras aplicaciones.



Figura 10: Sensor HD-38 [35].

Para llevar a cabo su funcionamiento, el higrómetro utiliza dos sondas encargadas de hacer pasar corriente de una a otra a través del suelo. El sensor obtiene un valor de resistencia al paso de dicha corriente, consiguiendo una lectura analógica de entre 0 y 1023 bits (0 a 5 V).

A mayor cantidad de agua en el suelo se tiene una mejor conducción de la electricidad, por lo que el sensor obtiene una resistencia más baja. Si por el contrario el suelo posee una menor cantidad de agua, la resistencia medida será mayor. Los valores obtenidos de la medición analógica van desde 0, para un suelo muy húmedo, hasta 1023 en un suelo muy seco.

A continuación, se muestran en la Tabla 8 las principales características que determinan el funcionamiento de este sensor:

Voltaje de operación	3,3 V a 5,0 V
Corriente de funcionamiento	20 mA

Corriente de salida	50 mA
Pines de salida	Pin digital (D0) y pin analógico (A0)
Chip controlador	LM393
Tiempo de respuesta	1,3 μ s

Tabla 8: Características relevantes del sensor HD-38 [45].

La principal característica por la cual se ha seleccionado el sensor HD-38 frente al FC-28, que es otro sensor de humedad del suelo, es la incorporación de una sonda anticorrosiva, impidiendo la oxidación del material. Por consiguiente, se consigue una mayor robustez y vida útil del sensor [44] [46].

5.2.8. Sensor FC-03

El FC-03 (Fig. 11) es un módulo que utiliza el opto-interruptor infrarrojo MOCH22A para detectar interrupciones en el paso de la luz infrarroja. Para ello, este sensor posee un emisor y un receptor de infrarrojos, que se encuentran separados 5 mm. Este espacio permite que un objeto pueda interrumpir el haz de luz, generando los pulsos en la señal lógica, que son digitalizados mediante el comparador LM393. Para detectar dichos pulsos mediante la placa de desarrollo de Arduino, se hará uso de las interrupciones, que se pueden detectar por flancos de subida o de bajada en la señal.



Figura 11: Sensor FC-03.

En nuestro caso, mediante un encoder ranurado que se ha obtenido mediante impresión 3D, podemos convertir los pulsos eléctricos que mide el sensor al interrumpirse el haz de luz, a velocidad angular y a velocidad lineal, lo que nos permitirá llevar a cabo la medición de la velocidad lineal del viento.

Se recomienda alimentar el sensor con 3,3 V, y también se puede añadir un condensador de 100 nF, que servirá como filtro para evitar falsos disparos de la interrupción [37].

Algunas características relevantes del sensor son las que se muestran a continuación en la Tabla 9:

Voltaje de operación	3,3 V a 5,0 V
Comparador	LM393
Longitud de onda del emisor	950 nm
Tipo de emisor	Fotodiodo IR
Tipo de receptor	Fototransistor

Tabla 9: Características relevantes del sensor FC-03 [48].

5.2.9. Pantalla LCD

Para poder mostrar texto y visualizar los datos recogidos en la estación meteorológica, se incorpora una pantalla LCD (*Liquid Crystal Display*). Dicha pantalla (Fig. 12), tiene un tamaño de 16x2, es decir, consta de 2 filas y 16 columnas donde se mostrarán los diferentes caracteres. Su fondo es de color azul mientras que las letras se muestran en blanco.



Figura 12: Pantalla LCD.

El principal problema que presenta la pantalla es la cantidad de conexiones que necesita con la placa de desarrollo de Arduino para funcionar adecuadamente. Necesita la unión de 16 pines con la placa de desarrollo. Estos pines son:

- RS: para controlar en qué parte de la memoria se escriben los datos.
- RW: para seleccionar entre el modo de lectura o escritura.
- E: pin de enable.
- D0-D7: 8 pines donde se escriben los registros.
- V0: pin para controlar el contraste.
- A y K: pines para controlar la retroiluminación de la pantalla.
- VSS y VDD: pines de tierra y alimentación.

Para solucionar dicho problema se incorpora un módulo de adaptación a I2C (Fig. 13), puesto que permite controlar la pantalla con el Arduino con tan solo los pines SDA y SCL. Además, dicho módulo cuenta con un potenciómetro para aumentar y disminuir el brillo de la pantalla [49].



Figura 13: Parte trasera pantalla LCD más el módulo I2C.

En la Tabla 10 se pueden apreciar algunas características técnicas de la pantalla LCD que se ha empleado:

Tipo de pantalla	LCD monocromática
Voltaje de alimentación	5 V
Controlador	HD44780
Modo de operación	4 y 8 bits

Tabla 10: Características relevantes LCD con módulo I2C [50].

5.2.10. Módulo adaptador para tarjeta micro SD

Uno de los dispositivos de suma importancia para el funcionamiento de la estación meteorológica es el módulo micro SD (Fig. 14). Esto se debe a que se va a hacer una tarjeta micro SD para almacenar los archivos generados en formato “.json” y “.csv”, los cuales contendrán los registros de las medidas de la estación meteorológica. El módulo se comunicará mediante el protocolo SPI, ya que necesita una alta tasa de transferencia.

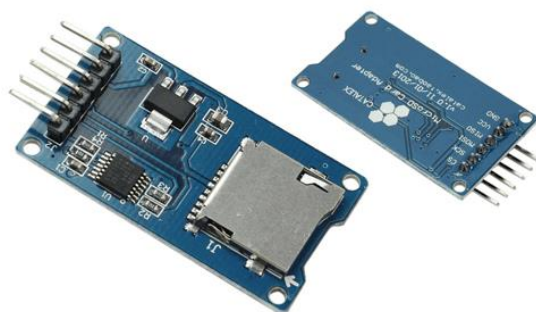


Figura 14: Módulo micro SD [51].

La ventaja que presenta este módulo es que ofrece una memoria casi ilimitada con relación al tamaño de los ficheros que nos permite guardar. Además, es no volátil, por lo que no se pierde la información cuando se desconecta la alimentación. Finalmente, puede ser extraída y conectada a un ordenador para el tratamiento de los datos que se hayan registrado [51].

A continuación, se han incluido en la Tabla 11 las características más importantes del módulo adaptador para tarjetas micro SD. Cabe destacar que el módulo trabaja con una tensión de 3,3 V, por lo que posee tanto un regulador de tensión como un circuito integrado para nivelar la señal de alimentación de 5 V a 3,3 V.

Tensión de alimentación	5 V
Tensión de funcionamiento	3,3 V
Interfaz de comunicación	SPI
Tarjetas aceptadas	SD, SDSC o SDHC

Tabla 11: Características relevantes módulo micro SD [52].

5.2.11. Real Time Clock

Tener un registro de la fecha y la hora es de gran utilidad para el proyecto debido a la necesidad de saber el momento en el que se han obtenido las mediciones de la estación meteorológica.

Hay dos formas de conseguir la fecha y la hora con Arduino: por medio del software a través de la librería “Time.h”, o por hardware con módulos RTC (*Real Time Clock*). La primera opción tiene el gran inconveniente de que no se puede almacenar la hora ni la fecha, es decir, si en algún momento se cierra el programa, el reloj comienza la cuenta desde cero. Para solucionar dicho problema, se ha utilizado un módulo RTC, que consiste en un dispositivo hardware con un circuito electrónico cuya función es mantener la hora y la fecha.

El RTC elegido es el modelo DS1307 (Fig. 15), que se caracteriza por tener un bajo consumo de energía y además por incluir una fuente propia, de la cual se puede alimentar si en algún momento se pierde la alimentación principal.

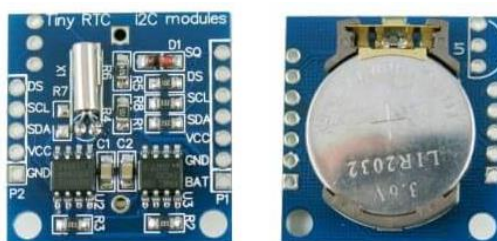


Figura 15: RTC DS1307 [53].

El dispositivo se comunica con la Arduino Mega 2560 mediante el bus I2C y almacena los registros en formato BCD (*Binary-Coded Decimal*). Este formato es un estándar de datos para representar números decimales en el sistema binario, donde dichos números decimales son codificados con una secuencia de 4 bits, ya que son los necesarios para representar los dígitos del 0 al 9 [53] [54].

En la siguiente tabla se reflejan los principales parámetros que caracterizan el RTC empleado en la estación meteorológica:

Voltaje de alimentación	5,0 V
Voltaje de la batería	3,0 V
Frecuencia nominal del cristal	32,768 kHz
Temperatura de operación	0 °C a 70 °C

Tabla 12: Características relevantes del RTC DS1307 [55].

6. DISEÑO DE LA ESTACIÓN METEOROLÓGICA

El diseño de la estación meteorológica implica llevar a cabo ciertos procesos que, tras completarlos, permitirán realizar las mediciones necesarias de las condiciones meteorológicas.

En primer lugar, se describe la estructura general que posee el código que permite tener a la estación un funcionamiento concreto, explicando la secuencia de acciones que se van a realizar y las distintas partes del código.

Tras haber expuesto el funcionamiento del código, se mostrarán las conexiones y la programación de todos los componentes. Se hace hincapié en las conexiones entre el elemento que se va a programar y la placa de desarrollo, profundizando en la programación que requiere cada sensor.

Una vez que se ha concretado el funcionamiento general del código, así como la programación y conexión de los componentes, se procede a la explicación del montaje del prototipo y su puesta en operatividad.

Finalmente, se incluye el manual de usuario, en el que se precisa cómo utilizar la estación meteorológica para que siga el funcionamiento especificado.

6.1. ESTRUCTURA GENERAL DEL CÓDIGO

El código que se ha realizado para que la estación meteorológica lleve a cabo su cometido, posee una estructura general específicamente diseñada para un funcionamiento concreto de la misma. El código completo se encuentra en el Anexo II.

De forma sintetizada, el funcionamiento de la estación es el que sigue. Tras cargar el programa en la placa de desarrollo y mantenerla alimentada mediante USB o a través del jack, se realizan una serie de configuraciones y comprobaciones iniciales. Entre ellas se encuentra la creación de los ficheros de la tarjeta micro SD en los que se van a recoger los datos medidos en los formatos “.csv” y “.json”. Una vez creados los archivos, un mensaje mostrado por la pantalla LCD nos indica que debemos pulsar el botón verde para el comienzo de las mediciones. Tras pulsar dicho botón, se comprueban las conexiones con los sensores, para registrar errores en la conexión en caso de producirse. A continuación, se producen las mediciones de cada parámetro de forma secuencial y después de medir todas las variables, se muestran sus valores por la pantalla LCD para que el usuario pueda ver las condiciones meteorológicas actuales. Seguidamente, se guardan los datos recogidos en la tarjeta micro SD para los dos formatos

anteriormente mencionados. Por último, la estación queda suspendida durante 10 minutos, en los cuales no se producen mediciones ni se muestra ningún dato por pantalla. En este periodo de tiempo será posible pulsar el botón de detención de la estación, evitando así que entre en un nuevo ciclo de medidas.

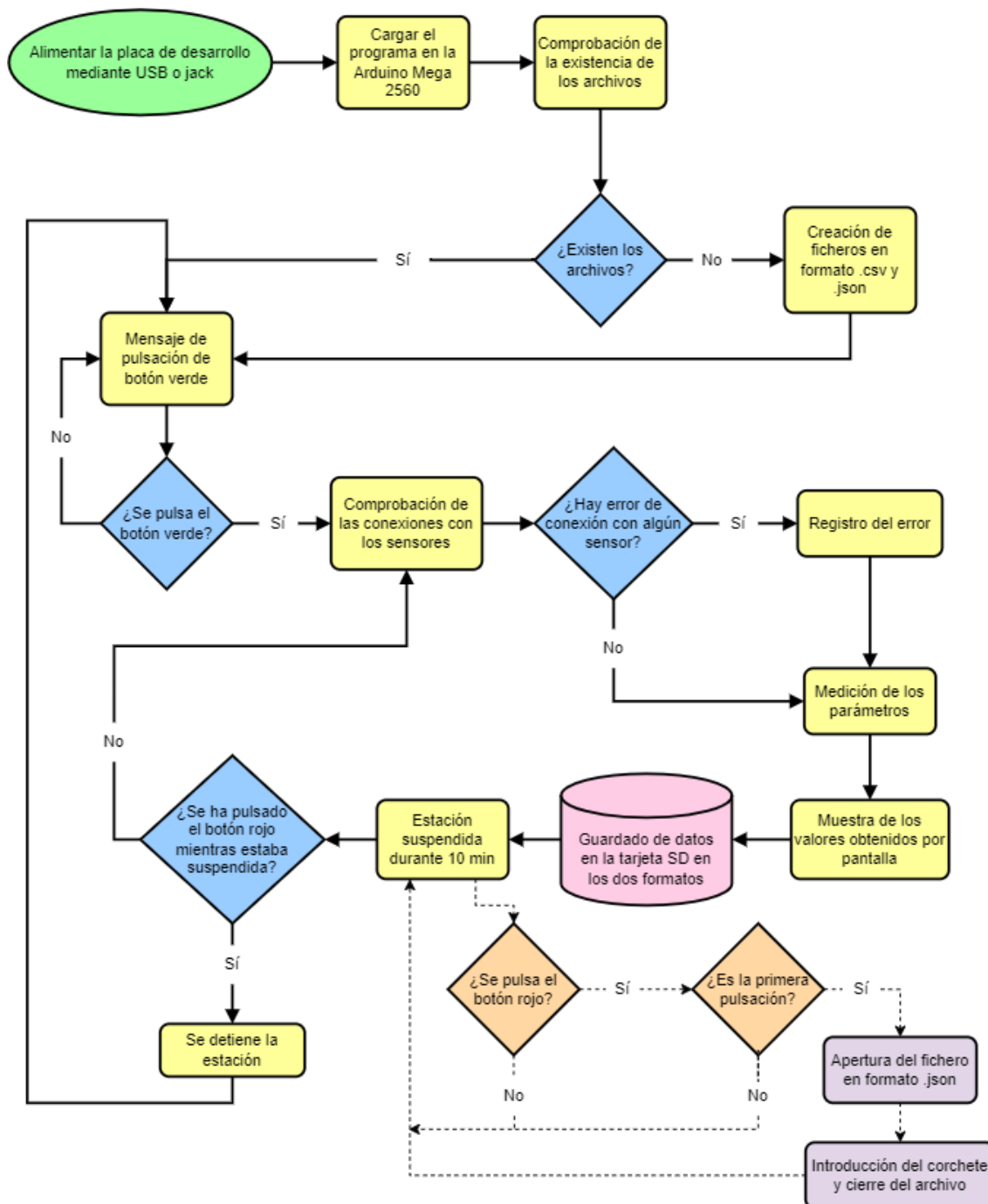


Figura 16: Diagrama de flujo del funcionamiento de la estación.

Realizando un análisis más exhaustivo de la estructura general del código que permite llevar a cabo el funcionamiento descrito, podemos encontrar 4 partes diferenciadas:

1. Bloque de declaraciones e inicializaciones.
2. Bloque de configuración *void setup ()*.
3. Bloque principal *void loop ()*.
4. Bloque de interrupciones e implementación de las funciones de los sensores.

Bloque de declaraciones e inicializaciones

En esta primera parte del código se pueden encontrar las librerías incluidas, las variables globales declaradas e inicializadas y la definición de los distintos pines utilizados y objetos.

En la Figura 17 se muestran todas las librerías necesarias para el funcionamiento y programación de los distintos elementos de la estación

```
// LIBRERÍAS:  
  
#include <RTClib.h>  
#include <Adafruit_BMP280.h>  
#include <SD.h>  
#include <SPI.h>  
#include <LowPower.h>  
#include <LiquidCrystal_I2C.h>  
#include <Wire.h>  
#include <DHT.h>  
#include <MQ135.h>
```

Figura 17: Extracto del código. Librerías incluidas.

En cuanto a la declaración de variables, cabe destacar la presencia de las variables “parada” y “memoria_corchete”, mostradas en la Figura 18. La variable “parada” se inicializa con el valor 2, pero a lo largo del código podrá tomar dos valores, 0 y 1. Cuando se pulsa el botón de inicio de la estación, se le asigna un 0 a la variable, mientras que, si se pulsa el botón de parada, recibe el valor de 1. Esto servirá para ejecutar unas zonas del código u otras, como se verá en el bloque principal *void loop ()*. Por otro lado, la variable “memoria_corchete” se inicializa a 0 y puede tomar dos valores, 0 y 1. Sirve para evitar que el efecto rebote del pulsador de parada genere más de 1 corchete de cierre del archivo *json*. Este funcionamiento se concreta con la explicación del bloque de las interrupciones.


```

// Variable para apagar y encender la estación.
int parada = 2;

// Variable para evitar repetición del corchete final en fichero JSON.
int memoria_corchete = 0;

```

Figura 18: Extracto del código. Variables destacadas.

Bloque de configuración void setup ()

A continuación, encontramos el bloque de configuración, en el que se van a ejecutar ciertas instrucciones una vez al cargar el programa.

En esta parte del código se configura la velocidad del puerto serial, se habilita la interrupción del pulsador que permitirá iniciar las mediciones, se inician la pantalla LCD y el RTC, ajustándose para mostrar la fecha y la hora actuales, se realiza la configuración de los pines digitales y se inicializa la tarjeta micro SD. En caso de no poder inicializar la tarjeta SD, la estación queda detenida ya que sin poder registrar los datos que miden los sensores, esta carece de funcionalidad. Esto último es lo que se muestra en la Figura 19:

```

// SD (Si no se puede conectar, se detiene la medición):
Serial.println("Iniciando tarjeta micro SD...");
if (!SD.begin(53)) {
    Serial.println("No se pudo iniciar la tarjeta");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("No se pudo");
    lcd.setCursor(0, 1);
    lcd.print("iniciar tarjeta.");
    lcd.clear();
    delay(2000);
    lcd.setCursor(0, 0);
    lcd.print("Estacion parada");
    while(1);
}
Serial.println("Inicio de SD exitoso.");

```

Figura 19: Extracto del código. Comprobación de la conexión con tarjeta micro SD.

Tras inicializar la tarjeta micro SD, se crean los ficheros “.csv” y “.json” y se escribe en ellos la cabecera del fichero, la cual solamente ha de insertarse al inicio de la ejecución de la captación de medidas. De esta forma, se generan las columnas para cada variable que va a registrar la estación en el fichero “.csv” y el corchete de inicio “[” necesario en el formato

json. En el caso del fichero “.csv”, es necesario separar con comas lo que queramos incluir en celdas distintas de la hoja de cálculo, que va a ser manipulada una vez recogidos los datos.

Bloque principal void loop ()

Tras el bloque de configuración encontramos la función principal del programa, el *void loop ()*. Esta parte del código es en la que se van a ejecutar los distintos comandos en bucle mientras la Arduino Mega 2560 esté funcionando.

En primer lugar, se habilita la interrupción del pulsador de inicio para permitir que, al pulsar el botón, la variable “parada” cambie su valor a 0. La variable toma dicho valor para poder entrar en el bucle *while* y comenzar así con las mediciones. Mientras no se pulse el botón de inicio, la variable seguirá teniendo un valor distinto de 0 y la estación se mantendrá a la espera sin realizar ninguna medición.

Cuando pulsemos el botón de inicio, permitiendo la entrada en el bucle *while*, la estación comenzará a actuar. En un primer momento, se comprueban las conexiones con varios de los sensores, enviando una señal de error si no se detectan. Cada error posee un código asociado que quedará registrado en el campo “observaciones” de los archivos generados. Estos códigos de error son los que se especifican a continuación en la Tabla 13:

Código de error	Descripción
EC.1	Error de conexión con el RTC.
EC.2	Error de conexión con el sensor DHT22.
EC.3	Error de conexión con el sensor BMP280.

Tabla 13: Códigos de error.

Tras estas comprobaciones, se produce la llamada a las funciones creadas para obtener las medidas de cada sensor. De esta manera, se registra el valor que devuelve cada función en una variable específica. Como ejemplo de ello, se muestra en la Figura 20 la llamada a las funciones del sensor DHT22, para obtener tanto la temperatura como la humedad relativa del aire.

```
float resultado_temperatura = funcion_DHT22_temperatura ();
float resultado_humedad = funcion_DHT22_humedad ();
```

Figura 20: Extracto de código. Ejemplo llamado de las funciones.

Después, se muestran los resultados por la pantalla LCD para que el usuario pueda ver las condiciones meteorológicas actuales, además de mostrarlos por el monitor serial en caso de que la Arduino Mega esté conectada al ordenador. Tras ello, se guardan los datos en los ficheros creados en el bloque de configuración.

Por último, se detendrá la estación durante 10 minutos, periodo de tiempo durante el cual la Arduino Mega 2560 se mantendrá en un estado de hibernación, el modo *Sleep*. Al entrar en este modo de funcionamiento, se detiene la ejecución del programa y se apagan ciertas partes físicas de la placa de desarrollo.

La razón por la que se ha incluido el modo *Sleep* en el funcionamiento de la estación es que no nos interesa obtener mediciones continuamente, es más útil que exista un periodo de tiempo entre las muestras. Además, mediante el uso de este modo, evitamos que durante el tiempo de espera exista un consumo innecesario de energía, lo que es un factor importante para una futura estación alimentada con baterías.

A continuación, se muestra la forma en la que se ha integrado el modo *Sleep* de Arduino en el programa (Fig. 21), para lo que ha sido necesario incluir la librería “LowPower.h”. Para activar el modo *Sleep*, se introduce la función “LowPower.powerDown”, que tiene 3 argumentos. El primero concreta que la hibernación durará 4 segundos, el segundo apaga el convertidor analógico-digital y el tercero mantiene el circuito BOD encendido. Este circuito debe mantenerse encendido ya que su utilidad es la detección de niveles bajos de tensión en la placa, manteniendo la integridad de la misma [56].

Para determinar el tiempo de hibernación en la función “LowPower.powerDown”, se ha escogido un tiempo de 4 segundos entre varias opciones permitidas (15 ms, 30 ms, 60 ms, 120 ms, 250 ms, 500 ms, 1 s, 2 s, 4 s, 8 s y *forever*), ya que este tiempo es múltiplo tanto de 1 minuto, para las pruebas en el laboratorio, como de 10 minutos, para la medición en condiciones reales.

De esta manera, para conseguir el tiempo de hibernación que nos interese, se incluye la función anterior en un bucle *for*. Para obtener un tiempo de hibernación de 10 minutos (600 segundos), se va a iterar 150 veces el modo *Sleep* de 4 segundos, tal y como se muestra en la Figura 21.

```
// ACTIVACIÓN MODO SLEEP (10 min):  
  
for (int i = 0; i < 150; i++){  
    LowPower.powerDown(SLEEP_4S, ADC_OFF, BOD_ON);  
    lcd.noDisplay();  
    lcd.noBacklight();  
}
```

Figura 21: Extracto del código. Modo *Sleep*.

Finalmente, tras salir del modo *Sleep*, se vuelve a empezar el ciclo de registro de datos desde el bucle *while*, que se ejecutará un número indefinido de veces hasta pulsar el botón de parada de la estación. Cabe destacar que únicamente se podrá detener la estación mientras esta se encuentre en modo *Sleep*.

Bloque de interrupciones e implementación de las funciones de los sensores

Este bloque es la última parte de la estructura del código. En esta parte se implementan todas las funciones necesarias para obtener las lecturas de los sensores, así como las interrupciones de los pulsadores de inicio y parada de la estación meteorológica.

Un aspecto destacable de las funciones de los sensores es que el valor que devuelven no es el resultado de una única medición, sino que se devuelve un valor promedio de 4 u 8 medidas consecutivas, dependiendo del sensor. Esto permite minimizar el error obtenido de la lectura de los sensores realizando una media de los valores registrados por los mismos.

En cuanto a las interrupciones de los pulsadores, al producirse una interrupción, se acude a la función que se haya asociado a dicha interrupción y se ejecuta su código. Tras ejecutarse la función asociada a la interrupción, el programa sigue ejecutándose desde el punto en el que se produjo la interrupción.

En el caso de la función del pulsador de inicio, se le asigna el valor 0 a la variable “parada”, lo que permite entrar en el bucle *while*, en el que se realizan las mediciones. En la función del pulsador de parada, se le asigna el valor 1 a la variable “parada”, lo que hace que se salga del bucle *while*, y además se introduce el corchete final en el archivo “.json” para respetar el formato. Para evitar que se puedan introducir varios corchetes por repetidas pulsaciones del botón de parada, se asigna el valor 1 a la variable “memoria_corchete”, ya que una condición para poder introducir el corchete es que dicha variable valga 0. Este funcionamiento se muestra a continuación en la Figura 22:

```
void PULSADOR_PARADA () {
    delay(80);
    parada = 1;
    archivoJSON = SD.open("ParteJ.txt", FILE_WRITE);

    if (archivoJSON && (memoria_corchete == 0)) {
        archivoJSON.println();
        archivoJSON.println("]");
        archivoJSON.close();
        memoria_corchete = 1;
        delay(2000);
    }
}
```

Figura 22: Extracto del código. Función del pulsador de parada.

6.2. CONEXIÓN Y PROGRAMACIÓN DE LOS COMPONENTES

El conexionado de los distintos componentes de la estación se ha realizado mediante cables para protoboard siguiendo el código de colores que se muestra a continuación en la Tabla 14:











Color del cable	Significado
	Rojo Alimentación a 5 V.
	Verde Alimentación a 3,3 V.
	Negro GND.
	Naranja Conexión con pines digitales y analógicos.
	Amarillo Conexión con pines digitales y analógicos.
	Morado Para interfaz SPI: MISO. Para interfaz I2C: SCL.
	Gris MOSI (Interfaz SPI).
	Marrón SCK (Interfaz SPI).
	Blanco CS (Interfaz SPI).
	Azul SDA (Interfaz I2C).

Tabla 14: Código de colores de los cables utilizados.

6.2.1. Temperatura y humedad

Para poder obtener las variables de temperatura y humedad relativa, se ha utilizado el sensor DHT22, cuyas características fueron explicadas en el apartado 5.2.2.

Para poder comunicarse con el microprocesador de la Arduino Mega, se han llevado a cabo las conexiones que se muestran en la Figura 23 según lo descrito en el *datasheet*. Este sensor dispone de 4 pines, que se enumeran a continuación:

- Vcc: alimentación a 5 V.
- Serial data: pin para leer y escribir los datos, conectado al pin digital 22 del Arduino Mega 2560.
- NC: pin vacío.
- GND: conectado a la tierra del Arduino.

Además, para el correcto funcionamiento del sensor, es necesario colocar una resistencia de 10 kΩ entre el pin serial data y el NC, ya que de esta forma se consigue eliminar el ruido de las medidas.

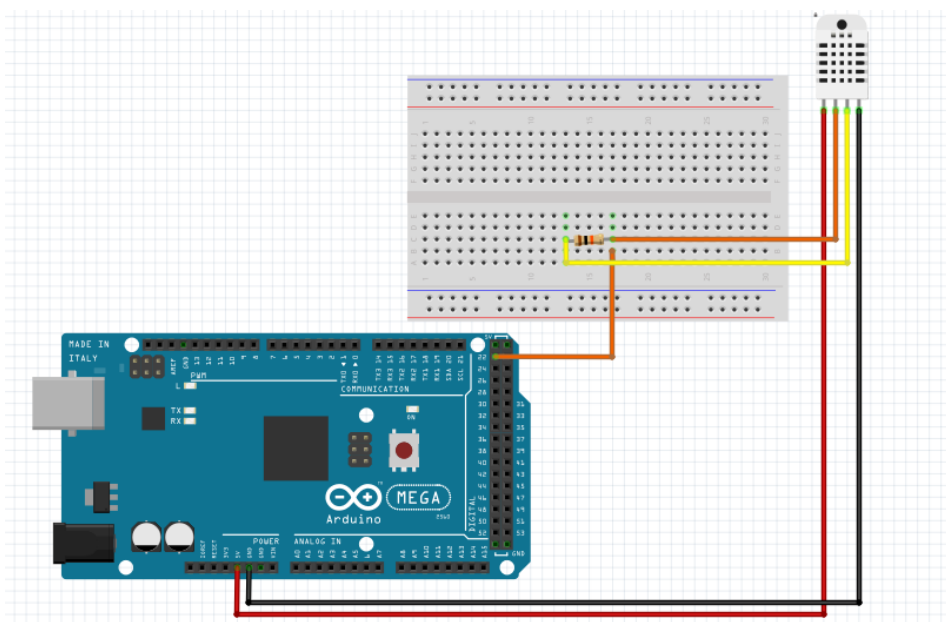


Figura 23: Circuito de conexión del sensor DHT22.

En cuanto a programación de dicho sensor, se debe instalar e incorporar al código la librería “DHT.h”, la encargada de comunicarse con el sensor.

Se han creado dos funciones: “funcion_DHT22_temperatura” encargada de recoger la medición de la temperatura y “funcion_DHT22_humedad” para obtener la humedad relativa del aire. La estructura de ambas es semejante, quedando resumida en tres partes (Fig. 24):

- Bucle *for* de 4 ciclos. Su función es recoger la medida del sensor 4 veces y así poder realizar una media, obteniendo un valor más preciso y salvando las variaciones entre una medida y otra.
- Comando “dht.readTemperature” o “dht.readHumidity”, donde se llama a la librería para adquirir los valores de humedad y temperatura.
- Un *delay* de 2,1 s, debido a que el tiempo de sensado del sensor es de 2 s.

```
float funcion_DHT22_temperatura (){

    float temperatura = 0;

    for(int i = 0; i < 4; i++){
        float temp = dht.readTemperature();
        temperatura = temperatura + temp;
        delay(2100);
    }
    temperatura = temperatura / 4;
    return (temperatura);
}
```

Figura 24: Extracto de código. Función para obtener la temperatura del aire.

6.2.2. Calidad del aire

Para la obtención de la calidad del aire se ha hecho uso del sensor MQ-135, cuyas conexiones con la placa de desarrollo Arduino Mega 2560 se muestran en la Figura 25 y son las siguientes:

- Vcc: alimentación a 5 V.
- GND: conectado a la tierra de la Arduino.
- A0: pin para leer y escribir los datos, conectado al pin analógico A0 del Arduino Mega 2560.
- D0: pin para conexión digital con los pines de la placa de desarrollo. No utilizado.

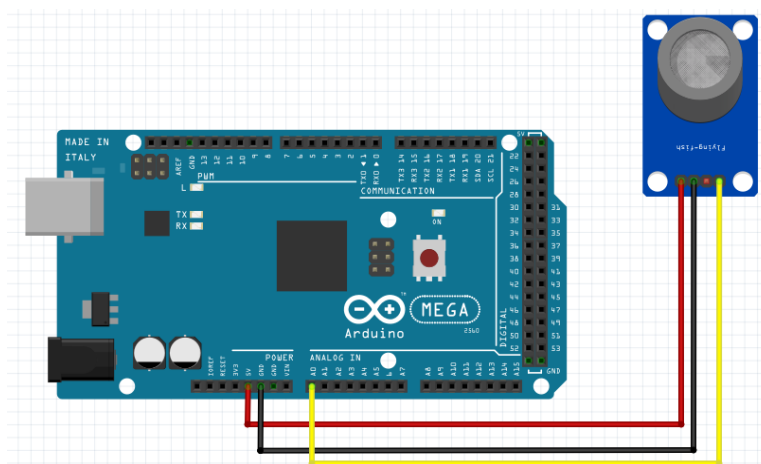


Figura 25: Circuito de conexión del sensor MQ-135.

En cuanto a la programación del sensor MQ-135, se ha empleado la librería “MQ135.h”, que nos permitirá obtener las lecturas deseadas del sensor. Para obtener el registro de calidad del aire se ha implementado la función “funcion_MQ135” que, a partir del valor en ppm de concentración de gases en el aire, devuelve un mensaje dependiendo del rango en el que se encuentre dicho valor.

En primer lugar, se obtiene el valor en ppm mediante el comando “sensor_MQ135.getPPM()”, con el que se conseguirá un valor medio empleando un bucle *for* de 4 ciclos. Una concentración normal de gases en el aire se sitúa entre 250 y 450 ppm aproximadamente. Por ello, valores por debajo de 250 ppm se pueden traducir como buena calidad del aire. Desde 450 hasta 800 ppm la calidad del aire desciende notoriamente, y es a partir de los 800 ppm donde las concentraciones pueden derivar en efectos negativos para la salud tanto de las personas como de las plantas.

Cabe destacar que este sensor lo que nos permite medir es la calidad del aire, lo que significa que se está midiendo la inmisión, no la emisión de gases contaminantes. Esto quiere decir que no se mide la cantidad de contaminantes que se evacúan a la atmósfera desde su foco emisor, sino que mide la concentración de contaminantes a nivel del suelo a cierta distancia del foco.

6.2.3. Pluviómetro

Debido a que en la actualidad no se dispone de un sensor capaz de medir directamente las precipitaciones acumuladas, en nuestro proyecto se ha diseñado la estructura necesaria en la que se almacenarán las precipitaciones y se medirá la distancia hasta la superficie del agua mediante un sensor de ultrasonidos.

El HC-SR04 se conecta con la placa de desarrollo Arduino Mega 2560 (Fig. 26) a través de los pines:

- Vcc: alimentación de 5 V.
- Trig: pin encargado de enviar el pulso, conectado al pin digital 47.
- Echo: pin encargado de recibir el pulso, conectado al pin digital 45.
- GND: conectado a la tierra de la Arduino.

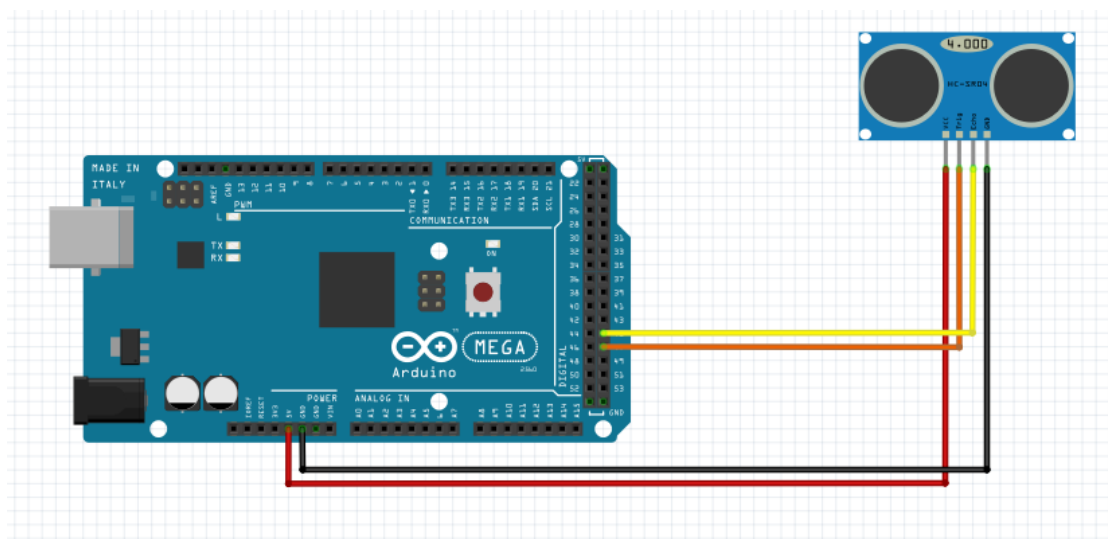


Figura 26: Circuito de conexión del sensor HC-SR04.

Una vez conectado el sensor con la placa, se ha creado la función “funcion_HCSR04”, en la que se interpretará la distancia del sensor, realizando los cálculos oportunos y obteniendo las precipitaciones acumuladas.

Este sensor no necesita ninguna librería, sino que únicamente se deben definir los pines digitales. En el bloque de implementación de las funciones, se añade la función correspondiente al HC-SR04, donde se emplea un bucle *for* de 8 ciclos para obtener el valor medio de la distancia (Fig. 27). En esta ocasión se realizan 8 ciclos en vez de 4 ya que las medidas que detecta presentan más variación que en otros casos.

```

for(int i = 0; i < 8; i++){
    digitalWrite(Trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trigger, LOW);

    tiempoHC = pulseIn(Echo, HIGH);
    distanciaHC = (tiempoHC * 0.0172);

    float distanciaHC_metros = (distanciaHC - 4.75)/100.0;
    distancia_media = distancia_media + distanciaHC_metros;
    delay(250);
}

```

Figura 27: Extracto de código. Medición de la distancia con sensor de ultrasonidos.

Cabe destacar que la cara del cubo en la que se encuentra el sensor de ultrasonidos es más alta que el resto. De esta forma, el borde del sensor está elevado 4,75 cm con respecto al borde superior del cubo, factor que se debe tener en cuenta para calcular la distancia con respecto al agua.

El sensor obtiene la distancia enviando un pulso durante 10 μs mediante la activación del *trig*. 10 μs es el tiempo mínimo del pulso de disparo, según el *datasheet*. Como se ha comentado anteriormente, el *echo* es el encargado de recopilar dicho pulso, y mediante el comando “pulseIn” se consigue tener el tiempo que transcurre desde que fue enviado hasta que fue recibido. Posteriormente, sabiendo que la velocidad del sonido (V_s) es de 344 m/s a 20 °C, y aplicando la fórmula (Ec. 1), se tiene la distancia en cm.

$$Distancia = \left(\frac{Vs}{1\text{ s}} \cdot \frac{1\text{ s}}{1000000\ \mu\text{s}} \cdot \frac{100\text{ cm}}{1\text{ m}} \right) / 2 \quad (1)$$

Finalmente, una vez conseguido el valor de distancia, se puede establecer que la altura del agua es igual a la altura del recipiente menos la distancia obtenida por el sensor. Conociendo la altura del agua y el volumen que ocupa en el recipiente, determinamos la cantidad de precipitaciones. Este cálculo se realizará cada cierto tiempo, por lo que se podrá ver la evolución de las precipitaciones acumuladas.

6.2.4. Radiación UV

Para la obtención de los niveles de radiación ultravioleta se ha utilizado el sensor ML8511, cuyas conexiones con la placa de desarrollo Arduino Mega 2560 se aprecian en la Figura 28 y son las siguientes:

- Vin: alimentación a 5 V. No utilizado.
- 3V3: alimentación de 3,3 V.
- OUT: salida analógica del sensor al pin A1 de la placa.
- EN (Enable): utilizado como referencia de tensión. Se conecta a 3V3 y al pin A2 de la placa.
- GND: conectada a la tierra de la Arduino.

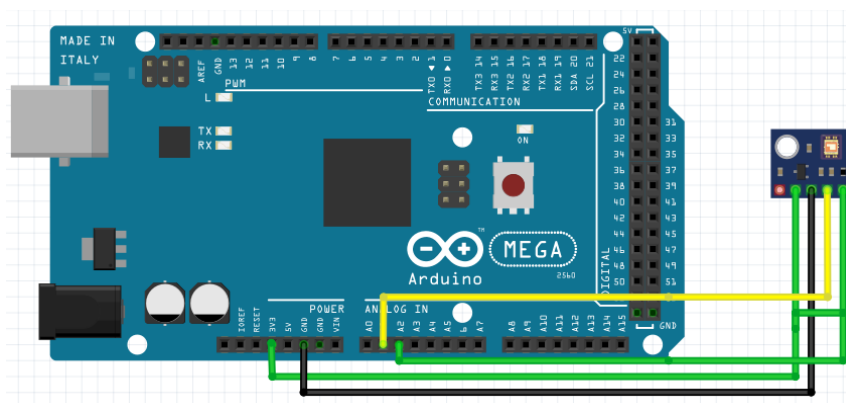


Figura 28: Circuito de conexión del sensor ML8511.

De tal forma que podamos conseguir lecturas de radiación ultravioleta, se han empleado las funciones: “funcion_ML8511”, “averageAnalogRead” y “mapfloat”. La primera es la función principal, en la que se realiza el bucle *for* característico para obtener el valor medio, en este caso de 4 ciclos (Fig. 29). Dentro de esta función, se llama a función “averageAnalogRead” para obtener un valor medio de la lectura analógica del sensor tanto para la salida analógica del sensor como para el nivel de referencia. La salida analógica del sensor nos da un valor en función de la radiación que detecta, que junto con el valor de referencia nos permitirá obtener la tensión de salida correspondiente.

```

for(int i = 0; i < 4; i++){
  nivel_uv = averageAnalogRead(UVOUT);
  nivel_ref = averageAnalogRead(REF_3V3);
  voltaje_salida = 3.3 / nivel_ref * nivel_uv;
  intensidad_uv = mapfloat(voltaje_salida, 1.0, 2.85, 0.0, 15.0);
  radiacion_uv = radiacion_uv + intensidad_uv;
  delay(100);
}

```

Figura 29: Extracto de código. Obtención de la radiación UV.

Para hallar la intensidad de la radiación ultravioleta se emplea la función “mapfloat” (Fig. 30), que permite realizar una interpolación a partir de la gráfica de la tensión de salida frente a la intensidad UV (Fig. 8). Para la interpolación se han tomado como referencia el valor mínimo y el valor máximo para una temperatura de 25 °C.

```

float mapfloat(float x, float in_min, float in_max, float out_min, float out_max) {
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

```

Figura 30: Extracto de código. Interpolación a partir de la curva del sensor ML8511.

6.2.5. Presión y altitud

Con el sensor BMP280 vamos a poder realizar las mediciones de la presión atmosférica, la altitud y la temperatura. Dicho valor de temperatura se comparará con la medición obtenida mediante el sensor DHT22, para conseguir un valor de temperatura más preciso.

El BMP280 se comunica a través de interfaz I2C con la placa de desarrollo Arduino Mega. Las conexiones con la placa (Fig. 31), siguiendo lo establecido en el datasheet, son las siguientes:

- Vcc: alimentación de 3,3 V.
- GND: conectado a la tierra de la Arduino.
- SCL: pin 21 correspondiente al SCL de la placa.
- SDA: pin 20 correspondiente al SDA de la placa.

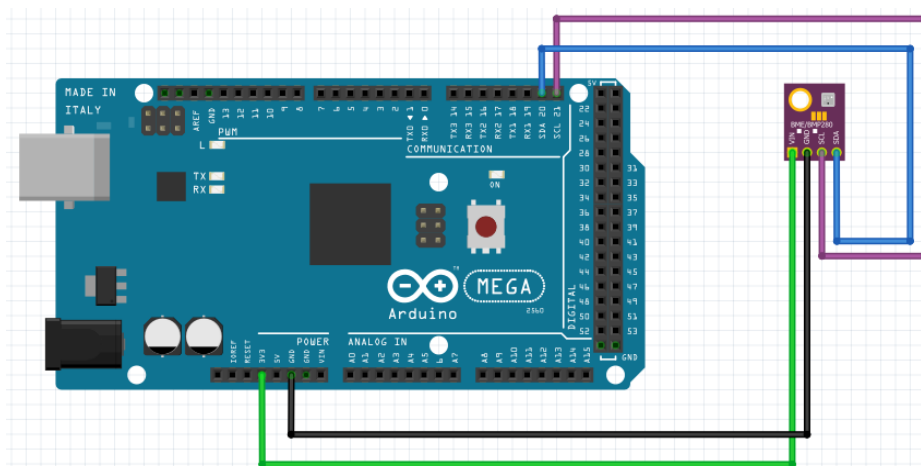


Figura 31: Circuito de conexión del sensor BMP280.

Durante la programación del sensor se tuvieron problemas para detectar el sensor, ya que la Arduino Mega 2560 no era capaz de establecer comunicación. Para solventar el error, se ha utilizado el ejemplo disponible en el entorno de Arduino “I2C_scanner”. Su utilidad es localizar la dirección de los dispositivos I2C y si se encuentran conectados. El programa establecía que el sensor se encontraba en la dirección 0x76, mientras que la dirección configurada en la librería que se ha utilizado “Adafruit_BMP280.h” era la 0x77. Tras corregir la dirección del dispositivo I2C, se detectó adecuadamente el sensor, obteniendo los valores requeridos.

La modificación que se realizó para detectar el sensor BMP280 se muestra en la Figura 32, estableciendo su dirección por defecto y su dirección alternativa como 0x76.

```

/#!/
 * I2C ADDRESS/BITS/SETTINGS
 */
#define BMP280_ADDRESS (0x76) /**< The default I2C address for the sensor. */
#define BMP280_ADDRESS_ALT
(0x76) /**< Alternative I2C address for the sensor. */

```

Figura 32: Modificación en la librería “Adafruit_BMP280.h”.

En cuanto a la programación del sensor, se realiza la misma estructura que los demás. Se implementa una función para obtener cada parámetro (presión, altitud, temperatura), en cuyo interior se tiene un bucle *for* para llevar a cabo la media entre 4 valores. Únicamente se utilizan los comandos “sensorBMP.readTemperature()”, “sensorBMP.readPressure()” y “sensorBMP.readAltitude()”, incluidos dentro de cada función para obtener la lectura del sensor.

6.2.6. Humedad del suelo

Como se ha mencionado en el apartado 5.2.7, el sensor seleccionado para poder medir la humedad del suelo es el HD-38. Junto con la sonda anticorrosiva, este sensor posee un módulo donde se encuentra el comparador LM393 y sus 4 pines de conexión.

- Vcc: alimentación de 5 V.
- GND: conectado a la tierra de la Arduino.
- D0: pin digital no utilizado.
- A0: pin analógico conectado al A3 de la Arduino.

Las conexiones del sensor HD-38 con la placa de desarrollo Arduino Mega 2560 se muestran en la Figura 33:

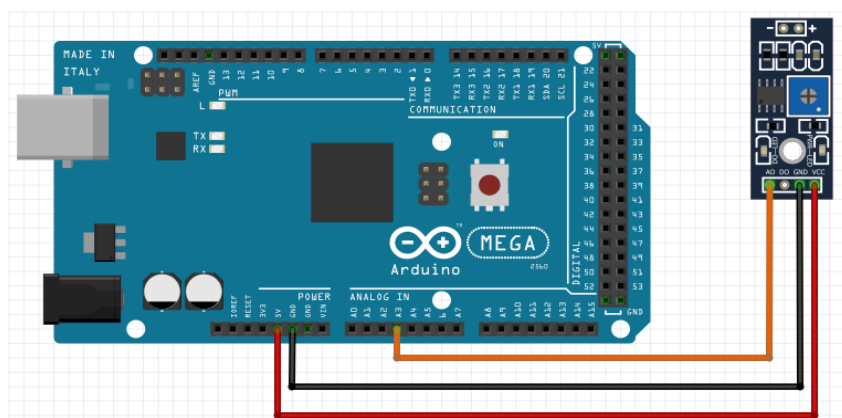


Figura 33: Circuito de conexión del sensor HD-38.

Como en el caso del sensor HC-SR04, tampoco necesita ninguna librería para su funcionamiento. Únicamente debemos definir, en el bloque de declaraciones e inicializaciones, el pin analógico A3 en el que se conecta el sensor. Para adquirir la lectura de humedad del suelo, se ha utilizado la función “funcion_HD38”. Dentro de la función, mediante el comando “analogRead(lectura_HD38)”, se ha obtenido la lectura del sensor. A continuación, se ha convertido el valor de dicha lectura a un valor de tensión, multiplicando por la tensión de alimentación (5 V) y dividiendo entre la resolución (1023). Por último, se ha procedido a mostrar el valor de tensión como un porcentaje que refleja el grado de humedad del suelo, donde 100% es húmedo y 0% es seco. El porcentaje se ha obtenido a través de la ecuación que encontramos a continuación (Ec.2):

$$\text{Porcentaje} = \left(1 - \frac{\text{voltaje leído}}{\text{voltaje máximo (5 V)}}\right) \cdot 100 \quad (2)$$

6.2.7. Anemómetro

Para llevar a cabo el anemómetro se ha diseñado una estructura que permite que el sensor FC-03 detecte las interrupciones de un disco ranurado conectado a un eje que gira a una velocidad angular proporcional a la velocidad lineal del viento. Las conexiones del módulo FC-03 con la placa de desarrollo se muestran en la Figura 34. Los pines del sensor son los siguientes:

- Vcc: alimentación a 3,3 V.
- GND: conectado a la tierra de la Arduino.
- D0: pin digital para comunicación con la placa de desarrollo.
- A0: pin analógico no utilizado.

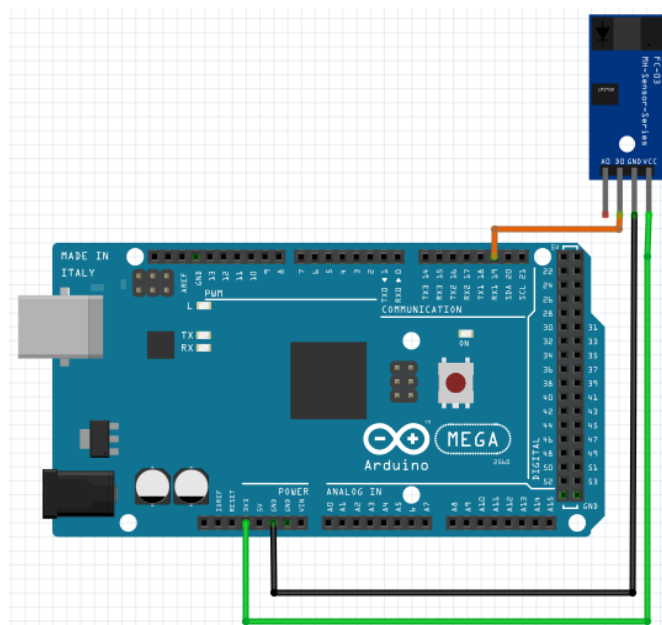


Figura 34: Circuito de conexión del sensor FC-03.

En lo referido a la programación del FC-03, no es necesario la utilización de ninguna librería. Para obtener el registro de la velocidad lineal del viento se ha implementado la función “funcion_FC03”, que a partir de las interrupciones detectadas por el sensor y realizando una serie de cálculos permite obtener dicho valor de velocidad.

En primer lugar, es necesario declarar e inicializar ciertas variables locales. A continuación, se realiza un bucle *for* de 30 iteraciones que permitirá obtener un valor medio. En esta ocasión se efectúan más iteraciones porque la variabilidad de las medidas es elevada y porque la medición de interrupciones se realiza cada segundo, dando lugar a 30 segundos de medición.

Dentro del bucle, habilitamos las interrupciones detectadas por el sensor FC-03 (Fig. 35), conectado al pin 19 de la placa de desarrollo, que permanecerán habilitadas durante 1 segundo (1000 ms).

```

attachInterrupt(digitalPinToInterrupt(19), ANEMOMETRO, FALLING);
delay(1000);
detachInterrupt(digitalPinToInterrupt(19));
  
```

Figura 35: Extracto de código. Habilitación de interrupciones del anemómetro.

Quando habilitamos las interrupciones, se asocia a dichas interrupciones la función “ANEMOMETRO”, que únicamente se encarga de contabilizar las interrupciones que se producen (Fig. 36).

```
void ANEMOMETRO() {  
    num_int++;  
}
```

Figura 36: Extracto de código. Función para contar interrupciones del anemómetro.

Tras realizar la cuenta de las interrupciones que se han producido en 1 s, se realizan una serie de operaciones con las que se obtendrá un valor de velocidad lineal del viento. En primer lugar, obtenemos el número de vueltas que dado el encoder. Es necesario saber que el encoder posee 12 ranuras, ya que este dato es esencial para obtener el número de vueltas que ha dado. El número de vueltas por minuto (rpm) se determina a partir del número de interrupciones que se han obtenido en un periodo de tiempo determinado aplicando los factores de conversión necesarios.

Una vez calculado el número de vueltas que se tendrían en un minuto, podemos determinar la velocidad angular del encoder. Esto se consigue al aplicar los factores de conversión para pasar de rpm a rad/s. Después, calculamos la velocidad del viento en m/s multiplicando la velocidad angular por el radio del encoder, que es de 0,015 cm. Para aportar el dato de velocidad en km/h se multiplica por 3,6.

Finalmente, tras obtener las 30 medidas de velocidad del viento, se realiza la media de dichas medidas y se devuelve ese valor.

6.2.8. Pantalla LCD

Con el fin de poder observar los datos directamente en la propia estación meteorológica se ha colocado la pantalla LCD de 16x2.

Las conexiones con la placa de desarrollo de Arduino se han simplificado en gran medida gracias a la incorporación del módulo I2C (Fig. 37), previamente explicado. Por tanto, únicamente se necesitan 4 pines:

- GND: conectado a la tierra de la Arduino.
- VCC: alimentación de 5 V.
- SDA: pin 20 correspondiente al SCA de la Arduino.
- SCL: pin 21 correspondiente al SCL de la Arduino.

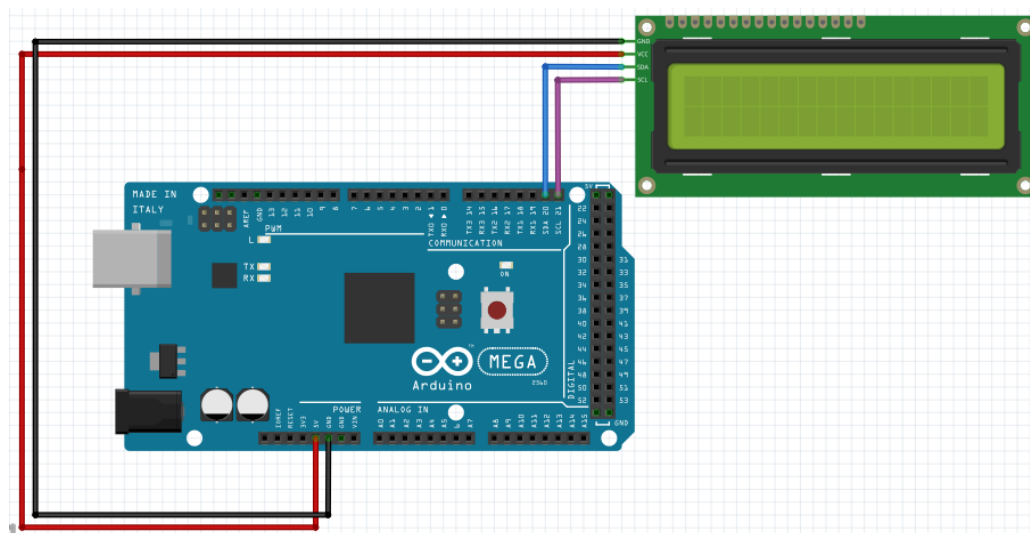


Figura 37: Circuito de conexión de la pantalla LCD con módulo I2C.

En lo referido a su programación, es necesario destacar una serie de cuestiones. En primer lugar, se debe instalar la librería “LiquidCrystal_I2C.h”, permitiéndonos realizar las funciones asociadas a la pantalla. A continuación, se indica la creación del objeto “lcd” en el bloque de declaraciones e inicializaciones, donde aparece la dirección del I2C y también el tamaño de la misma, siendo en este caso de 16x2. Posteriormente, se procede al inicio de la pantalla, con los comandos que aparecen en la Figura 38. Los comandos que se muestran tienen las siguientes utilidades respectivamente: inicio del módulo I2C, iluminación del fondo y limpiar la pantalla para poner el cursor en la parte superior izquierda de la misma.

```

// Inicio de pantalla LCD:
lcd.init();
lcd.backlight();
lcd.clear();

```

Figura 38: Extracto de código. Comandos de inicio LCD.

Finalmente, para mostrar el texto en un espacio concreto de la pantalla, se emplea el comando “setCursor()”. En el interior de los paréntesis se incluyen dos cifras: la primera cifra indica el número de la columna (0-15) y la segunda cifra indica la fila (0-1). Un ejemplo de ello es “setCursor(1,1)”, que permite mostrar el mensaje en la 2 columna y la 2 fila de la pantalla.

6.2.9. Módulo para tarjeta micro SD

Como se ha mencionado en el apartado 5.2.10, el módulo para tarjeta micro SD es de suma importancia, dado que se crearán un archivo “.csv” y otro “.json” en la tarjeta micro SD, en los que se recogerán todas las medidas realizadas durante el periodo de tiempo que se desee.

El archivo en formato “.csv” nos permite abrirlo utilizando *Excel*, siendo este el programa seleccionado para realizar las gráficas de los diferentes resultados. Por otro lado, el archivo “.json” permite el intercambio de datos con la página web resultado del TFG asociado a este proyecto de la Estación Meteorológica, realizado por un estudiante del Grado en Ingeniería Informática en el curso actual, y que tiene como título “Aplicación fullstack Mi Finquita”.

Las comunicaciones con la placa de desarrollo se realizan mediante interfaz SPI. En la Figura 39 se muestran las conexiones de un módulo para tarjeta SD, pero son las mismas conexiones que para el caso del adaptador micro SD. La única diferencia que se puede apreciar es la ranura del adaptador. En el módulo para tarjeta SD tenemos 8 pines, de los cuales 6 se utilizan para el adaptador de micro SD:

- GND: conectado a la tierra de la Arduino.
- VCC: pin que se utiliza para alimentarlo a 3,3 V. Pin no incluido en el adaptador de micro SD.
- VCC: alimentado con 5 V de la Arduino.
- CS: pin 53 de la Arduino.
- MOSI: pin 51 de la Arduino.
- SCK: pin 52 de la Arduino.
- MISO: pin 50 de la Arduino.
- GND: pin no utilizado.

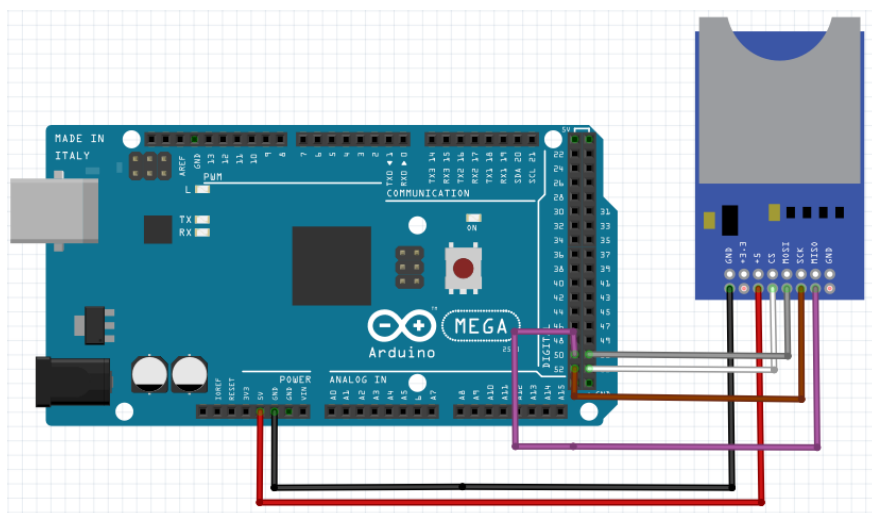


Figura 39: Circuito de conexión del módulo adaptador tarjeta micro SD.

Para programar el módulo para tarjeta micro SD, es necesario instalar la librería “SD.h”. Un apartado de suma importancia es la comprobación de la comunicación de la Arduino con el dispositivo. Si la comunicación no se efectúa correctamente, el programa se detendrá ya que, si no se guardan los datos de las mediciones, la estación no sería de utilidad. Para ello, se emplean las instrucciones que se muestran en la Figura 40, con las cuales comprobamos si se puede iniciar la tarjeta, impidiendo que el programa avance en caso de no detectar la SD.

```

// SD (Si no se puede conectar, se detiene la medición):
Serial.println("Iniciando tarjeta micro SD...");
if (!SD.begin(53)) {
    Serial.println("No se pudo iniciar la tarjeta");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("No se pudo");
    lcd.setCursor(0, 1);
    lcd.print("iniciar tarjeta.");
    lcd.clear();
    delay(2000);
    lcd.setCursor(0, 0);
    lcd.print("Estacion parada");
    while(1);
}
Serial.println("Inicio de SD exitoso.");

```

Figura 40: Extracto de código. Comprobación de conexión con la tarjeta micro SD.

Una vez que se ha comprobado la conexión, se procede a la creación de los archivos. En la función *void setup* se crearán los diferentes archivos. Por un lado, se crea el archivo “.csv” donde se insertarán los campos de la tabla, es decir el nombre de las variables medidas y sus

unidades. Por otro lado, el archivo “.json”, donde únicamente se generará el corchete de inicio “[”. Los comandos de creación de los archivos se encuentran dentro de una condición en la que, si los archivos ya están creados, no es necesario realizar el proceso.

Finalmente, cuando se ha terminado de realizar la recopilación de las mediciones de los sensores, se abrirán los archivos mediante la función “SD.open”, para registrar en ellos dichos valores.

6.2.10. RTC

El módulo RTC DS1307 dispone de 13 pines en total, pero debido a que la comunicación se realizará utilizando el protocolo I2C, la conexión de los pines (Fig. 41) se lleva a cabo de la siguiente forma:

- Vcc: alimentación de 5 V.
- GND: conectado a tierra de la Arduino.
- SDA: pin 20 correspondiente al SCL de la Arduino.
- SCL: pin 21 correspondiente al SCL de la Arduino.

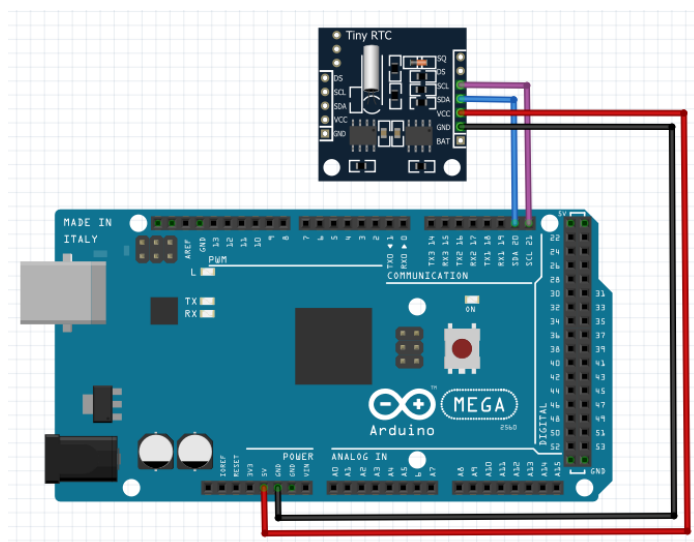


Figura 41: Circuito de conexión del RTC.

Como en la mayoría de los componentes, el RTC DS1307 requiere de la instalación de la librería “RTClib.h”. Seguidamente, se configuran la fecha y la hora, como se puede observar en la Figura 42.

```
// Inicio de RTC y configuración de fecha y hora:  
rtc.begin();  
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));  
Serial.print(__DATE__);  
Serial.print(" ");  
Serial.print(__TIME__);
```

Figura 42: Extracto de código. Configuración del RTC.

Para concluir, a través del comando “now”, se podrá obtener el valor actual del RTC de día, mes, año, hora, minuto y segundo. Como ejemplo, “now.year()” mostrará el valor del año.

6.3. MONTAJE DEL PROTOTIPO

6.3.1. Protección de los componentes

Para asegurar la protección de los diferentes componentes se han diseñado, con el programa *SketchUp*, una serie de envolturas, que posteriormente se han impreso en las diferentes impresoras 3D del BiblioLab.

Como características generales, todas las estructuras constan de dos partes, una parte superior y otra inferior, que serán unidas mediante tornillos y tuercas de 2,5 mm de diámetro. Además, en dichas partes figuran unos topes con el fin de asegurar que los elementos no se desplacen en el interior de sus carcasas y queden fijos. Por último, se ha dejado una ranura a la altura de los pines, para permitir la conexión de los cables.

Para acceder a los diseños en el formato de *SketchUp* “.skp” y en formato objeto “.obj”, acudir al Anexo I.

6.3.1.1. Arduino Mega 2560

La carcasa de la placa de desarrollo de Arduino Mega 2560 está compuesta por dos piezas, la parte superior (Fig. 43) y la parte inferior (Fig. 44). En cuanto a la tapa, se han diseñado 3 ranuras que permiten el acceso a todos los pines de la placa. Además, tiene una serie de topes que evitan los movimientos de la Arduino dentro de la carcasa.

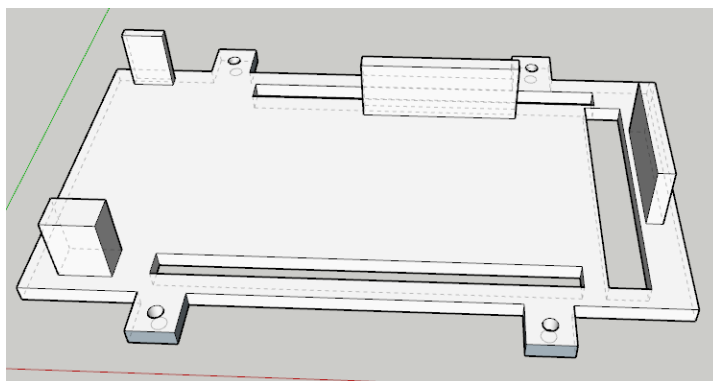


Figura 43: Parte superior de la carcasa de la Arduino Mega.

La parte inferior de la carcasa está provista de dos huecos rectangulares a uno de los lados, que permitirán la alimentación de la placa tanto por USB como mediante el *jack*. También posee una serie de topes en la base para sujetar la placa.

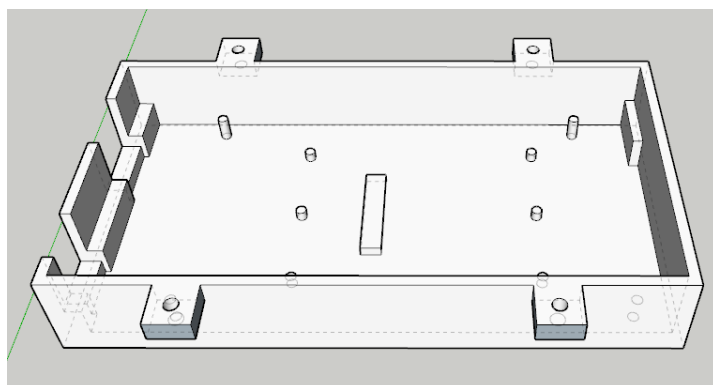


Figura 44: Parte inferior de la carcasa de la Arduino Mega.

6.3.1.2. MQ135

Para el diseño del sensor MQ-135, se ha creado un hueco circular en la tapa, que permitirá detectar eficazmente la cantidad de concentración de gases. Para evitar el movimiento del sensor en el interior de la carcasa, en las esquinas de la tapa se han colocado unos bordes al efecto (Fig. 45).

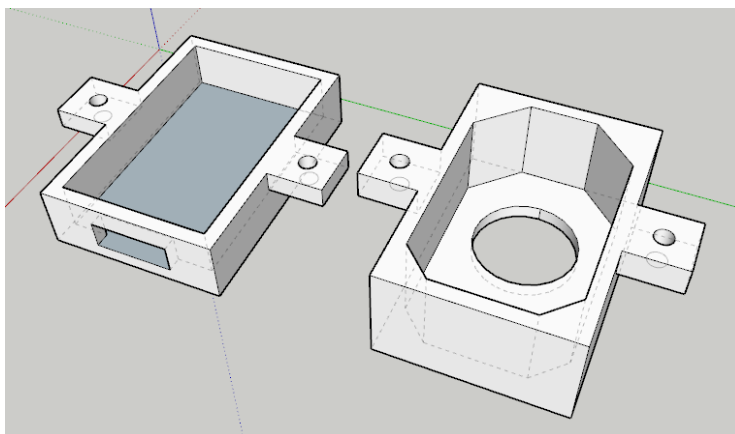


Figura 45: Diseño 3D de la carcasa del sensor MQ-135.

6.3.1.3. HC-SR04

En la Figura 46 se puede observar el diseño en 3D de la carcasa del sensor Hc-SR04, que se destaca por la introducción de dos orificios en la parte inferior de la tapa, mediante los cuales el componente podrá enviar y recibir el pulso. Además, se ha realizado una superficie rectangular con 4 agujeros de 3,5 mm de diámetro, que permitirá fijar la estructura a la pared del recipiente de las precipitaciones.

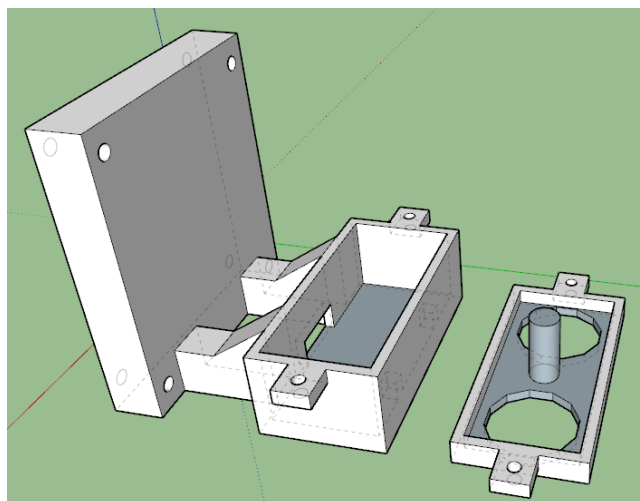


Figura 46: Diseño 3D de la carcasa del sensor HC-SR07.

6.3.1.4. ML8511

La carcasa del sensor ML8511 (Fig. 47), además de incluir las características generales de las demás estructuras, también incluye un orificio por el cual se detectan los rayos UV.

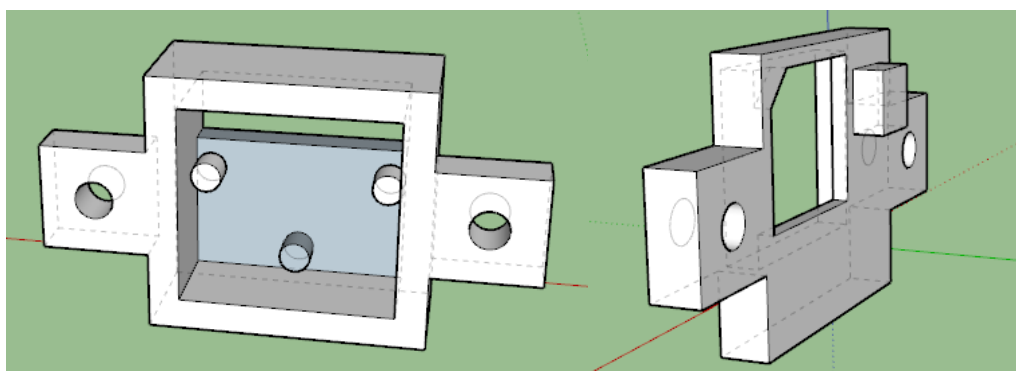


Figura 47: Diseño 3D de la carcasa del sensor ML8511.

6.3.1.5. BMP280

Para el diseño de la envolvente del sensor BMP280 (Fig. 48), se imitado a la estructura del sensor DHT22, es decir, en la parte de arriba de la carcasa, se ha incluido una serie de orificios formando una rejilla, con el objetivo de permitir el contacto del componente con el exterior, y así recopilar mejor las mediciones.

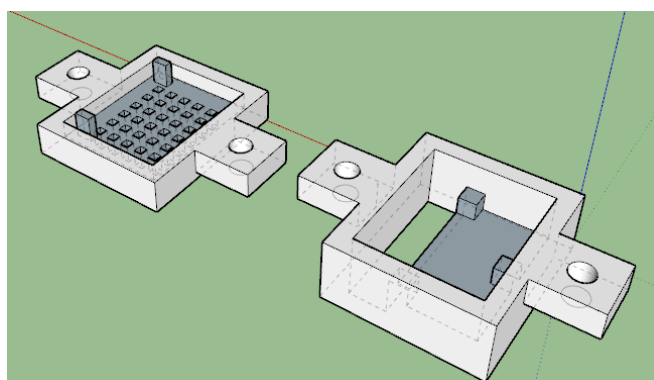


Figura 48: Diseño 3D de la carcasa del sensor BMP280.

6.3.1.6. HD-38

El diseño de la carcasa para el sensor HD-38 (Fig. 49) posee topes para evitar movimientos del sensor dentro de la carcasa y una ranura que permite la conexión de la sonda con el módulo.

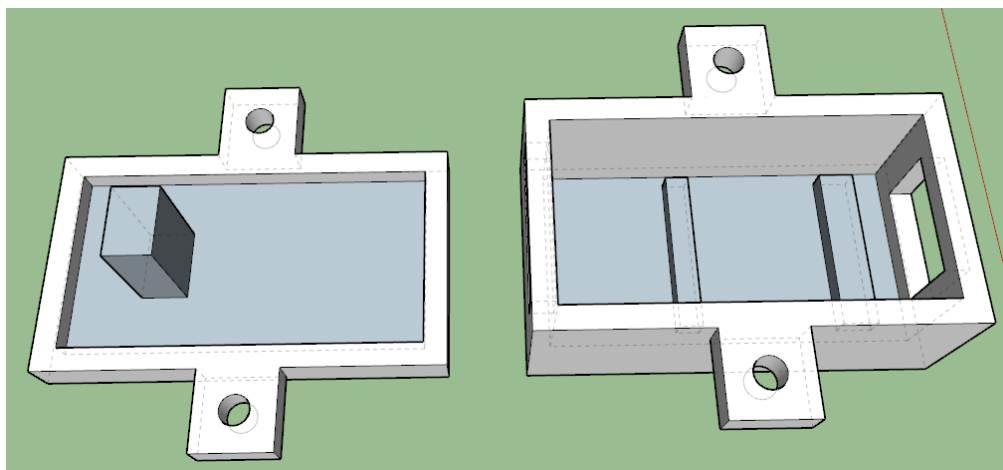


Figura 49: Diseño 3D de la carcasa del sensor HD-38.

6.3.1.7. FC-03

Para realizar el diseño de la carcasa del sensor FC-03, que es el sensor utilizado en el anemómetro, se han tenido en cuenta distintos aspectos.

En primer lugar, tenemos la parte inferior de la carcasa (Fig. 50), que contiene los topes de sujeción y la ranura para la conexión con los pines.

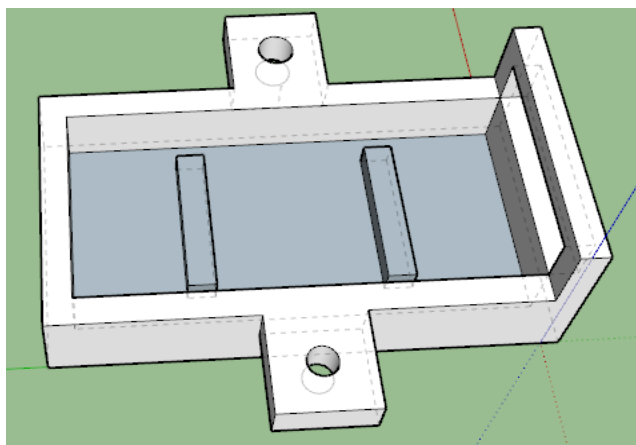


Figura 50: Diseño 3D de la parte inferior del FC-03.

Por otra parte, la parte superior de la carcasa posee una ranura que permitirá colocar el encoder entre el emisor y el receptor del sensor infrarrojo de forma que se detecten adecuadamente las interrupciones. Además, uno de los lados tiene unas dimensiones mayores que permitirán fijar el sensor a la tapa de la caja estanca a la distancia exacta a la que se encuentra el encoder del anemómetro (Fig. 51).

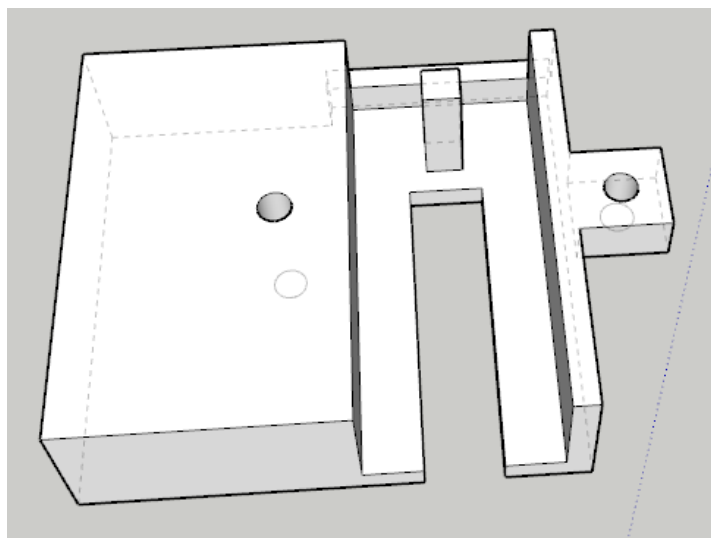


Figura 51: Diseño 3D de la parte superior del FC-03.

6.3.1.8. Pantalla LCD

Además de incluir las características generales de las otras envolventes, en el caso de la LCD se ha diseñado la abertura necesaria para la visualización de la pantalla (Fig. 52).

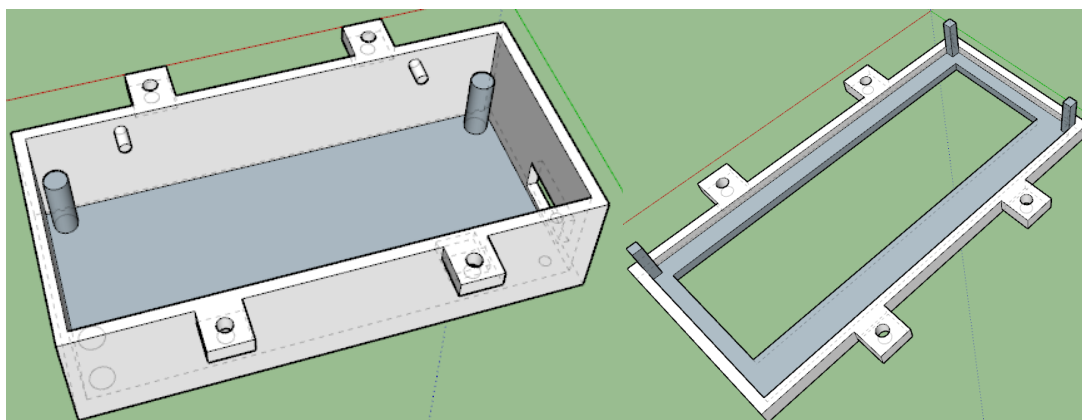


Figura 52: Diseño 3D de la carcasa de la LCD.

6.3.1.9. Módulo para tarjeta micro SD

En cuanto al diseño 3D de la protección del módulo para tarjeta micro SD, se destaca que se ha incluido una ranura con el tamaño suficiente para retirar la tarjeta micro SD sin necesidad de desmontar la carcasa (Fig. 53).

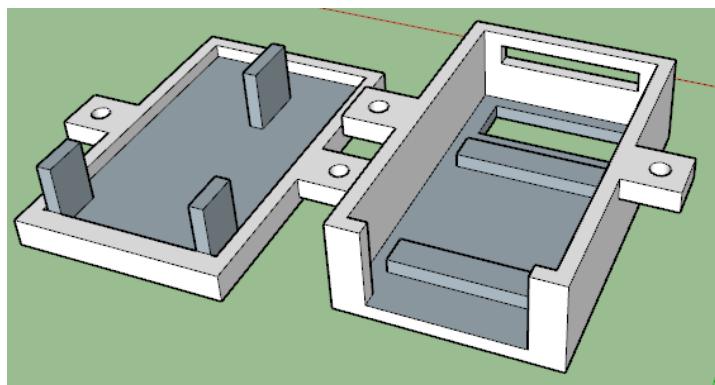


Figura 53: Diseño 3D de la carcasa del módulo para tarjeta micro SD.

6.3.1.10. RTC

En la parte superior de la carcasa, se ha diseñado un soporte con la función de mantener fija la alimentación secundaria, consiguiendo que, en caso de movimientos de la misma, no se produzca la desconexión. La parte inferior contiene los topes necesarios para evitar los movimientos del componente (Fig. 54).

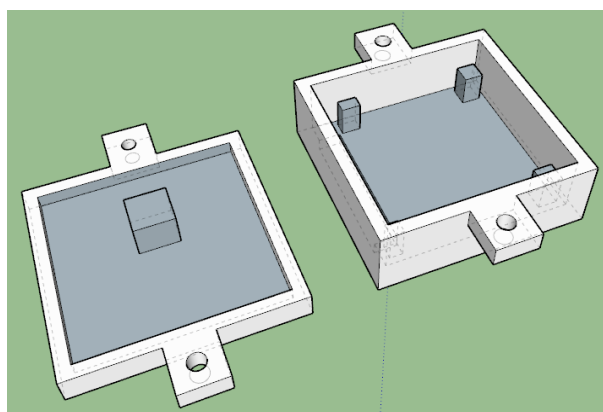


Figura 54: Diseño 3D de la carcasa del RTC.

6.3.2. Pluviómetro y anemómetro

Pluviómetro

Para medir la cantidad de precipitaciones se ha construido un recipiente de metacrilato de 20 cm de largo, 20 cm de ancho y 20 cm de alto, en cuyo interior se almacenarán las diferentes cantidades de agua, granizo o nieve (Fig. 55). Las juntas del envase han sido fijadas con un pegamento especial para metacrilato y a continuación, se han sellado con un adhesivo transparente a base de polímero MS resistente al agua.

Una de las caras del cubo es más elevada, lo que permite colocar el sensor de ultrasonidos HC-SR04 a cierta altura, evitando así ser alcanzado por el agua si esta llega al borde del recipiente. Es necesario destacar que el borde del sensor se encuentra elevado 4,75 cm con respecto al borde superior del cubo.

Las dimensiones del cubo permiten acumular un volumen de 8 litros de precipitaciones. Al dividirlo entre la superficie de la base, que son 0,04 m², obtenemos que se pueden medir hasta 200 L/m². Se ha decidido que el recipiente posea dichas dimensiones tras consultar los valores de precipitaciones acumuladas estacionales en las Islas Canarias [57].

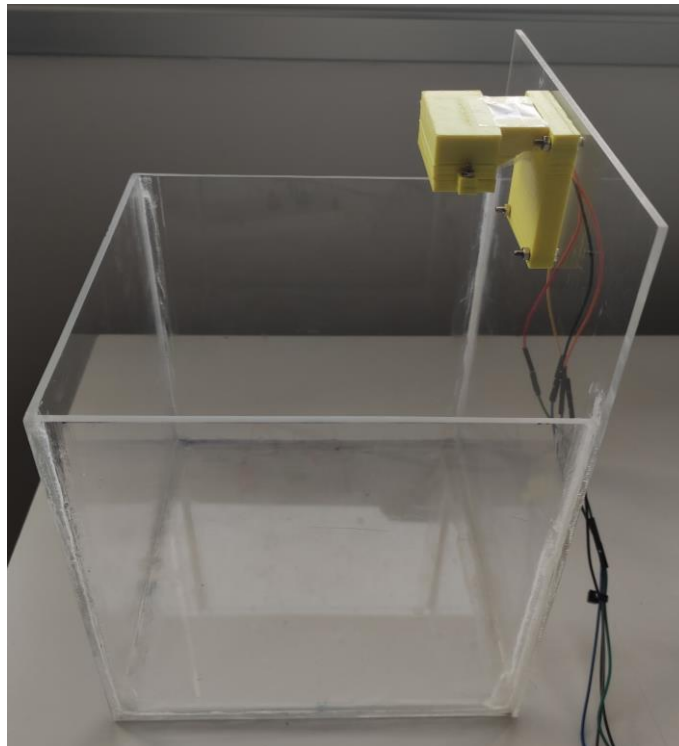


Figura 55: Estructura del pluviómetro.

Anemómetro

La estructura del anemómetro está compuesta por diferentes piezas con funciones específicas, diseñadas con el programa ya mencionado, *SketchUp*, e impresas posteriormente por las impresoras 3D disponibles en la BiblioLab. De esta manera, las partes que conforman la estructura del anemómetro son:

1. Base del anemómetro.
2. Eje con los rodamientos y el encoder.
3. Cúpula.
4. Aspas.

Por un lado, se ha desarrollado la base del mismo, compuesta por una superficie cuadrada y un tubo hueco. En dicha superficie, se han elaborado cuatro orificios con el objetivo de fijar el anemómetro a la caja estanca mediante la inserción de tornillos de 3 mm de diámetro.

En el interior del tubo hueco se introduce el eje con el encoder de 12 ranuras, encargados de girar a una velocidad angular proporcional a la velocidad lineal del viento. Para reducir al máximo cualquier posible rozamiento, se ha adherido el eje a las paredes del tubo utilizando rodamientos, los cuales permiten el correcto movimiento además de asegurar la sujeción (Fig. 56).



Figura 56: Eje unido al tubo mediante los rodamientos.

La cúpula es una pieza elaborada con el propósito de permitir la unión del eje con las tres aspas del anemómetro. Dicha pieza se ha diseñado con 4 aberturas, una en la parte inferior, donde se coloca el eje, y las otras 3 en la zona lateral de la misma, separadas a la misma distancia unas de otras, en las cuales irán introducidas las aspas.

Finalmente, se han fijado las diferentes piezas con un pegamento especial para plásticos rígidos, consolidando la estructura y permitiendo el giro libre de las aspas del anemómetro para

obtener una medida fiable de la velocidad del viento. Se muestra la estructura del anemómetro en la Figura 57:

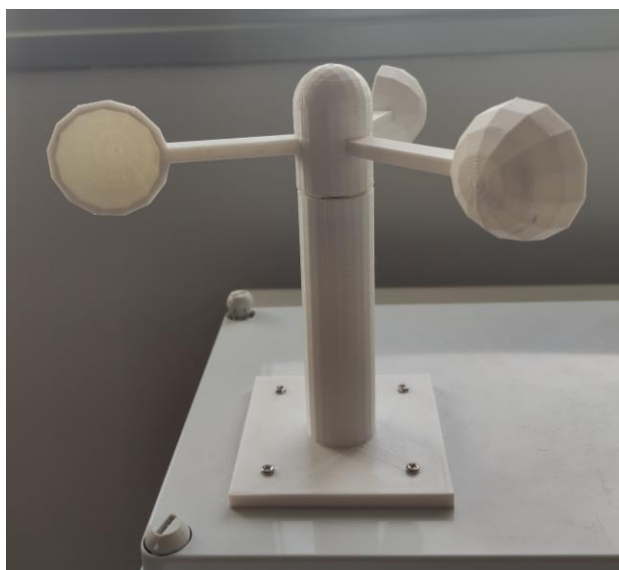


Figura 57: Estructura del anemómetro.

6.3.3. Estructura del prototipo

Una vez finalizada la conexión y programación de los componentes y el diseño de las carcasas de los mismos, se ha procedido a realizar el montaje final del prototipo.

Para la estructura principal, es decir, la zona donde van situados los diferentes componentes se ha seleccionado una caja estanca rectangular de dimensiones 240 x 190 x 90 mm. Con estas dimensiones se posee la capacidad suficiente para la introducción de los elementos en ella. Además, está dotada con un grado de protección IP55, que incluye protección contra la penetración del polvo y protección ante chorros de agua en todas las direcciones. Para poder colocar correctamente los elementos, se ha añadido al fondo de la misma una plancha de metacrilato capaz de cubrir la superficie (Fig. 58).

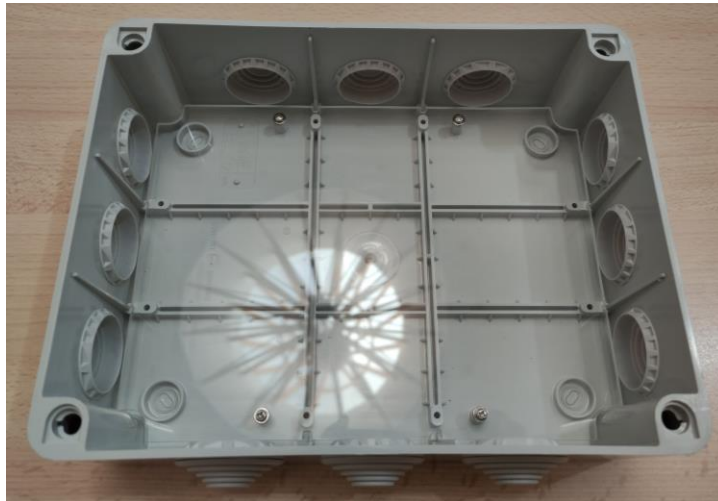


Figura 58: Caja estanca con la plancha de metacrilato.

En la parte superior de la estructura, se ha llevado a cabo un agujero con el tamaño suficiente para poder introducir el encoder del anemómetro. También se realizaron los orificios necesarios para instalar el anemómetro. Una vez fijado el instrumento, se adhirió el sensor FC-03 a la parte interna de la tapa de la caja, a la altura exacta a la que se encuentra el encoder para poder medir correctamente las interrupciones.

Además, en la parte frontal, se efectuó una ranura aprovechando los orificios de los que está provista la caja estanca. La utilidad de la ranura es colocar la pantalla LCD para observar los datos medidos desde el exterior. El agujero está protegido por una pieza de metacrilato con las juntas selladas para evitar la entrada de elementos externos.

Finalmente, en el interior de la caja (Fig. 59), se fijó con velcro los diferentes componentes:

- Arduino Mega 2560.
- Protoboard.
- Módulo para tarjeta micro SD.
- HC-38.
- RTC.

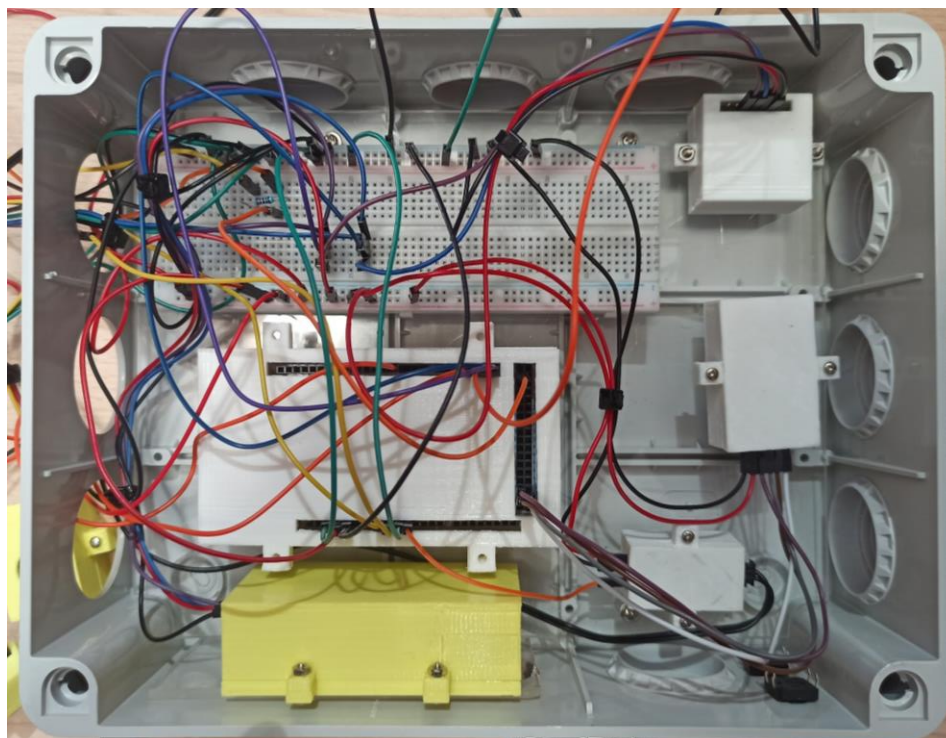


Figura 59: Interior de la caja estanca con los componentes.

6.4. MANUAL DE USUARIO

Para el correcto funcionamiento de la estación meteorológica, se ha elaborado un protocolo con las acciones a realizar en cada momento.

1. Asegurarse de que la tarjeta micro SD se encuentra en el interior del módulo para tarjetas micro SD de la caja estanca.
2. Conecte la alimentación, asegurándose de que el *jack* está introducido en la entrada correspondiente del Arduino.
3. Cuando en la pantalla aparezca “Pulse botón verde para ON”, accione el pulsador verde ubicado en la parte derecha de la caja y espere 5 segundos (Fig. 60).
4. La estación está en funcionamiento.
5. Si desea detener la estación para recoger la tarjeta micro SD con los datos, pulse botón rojo, ubicado en la parte derecha, únicamente cuando la pantalla se encuentra apagada.
6. Una vez pulsado el botón de parada, desconecte la alimentación.

7. Retire la tarjeta micro SD y, una vez recogidas las mediciones, elimine el archivo ".json" de la tarjeta.
8. Repita este proceso para la puesta en marcha.



Figura 60: Pantalla LCD y botones de inicio y parada.

7. RESULTADOS

Una vez completado el diseño de la estación meteorológica, la programación de los componentes, el diseño e impresión de las carcasas, la adecuación de la caja estanca y la inclusión de los distintos elementos en la misma, se ha procedido a verificar su funcionamiento en condiciones reales.

7.1. CONDICIONES DE LA MEDICIÓN

La estación meteorológica fue situada durante 23h en una terraza, donde estaba protegida del agua, pero a la vez mantenía contacto con el exterior. La zona contaba con un techo, pero los laterales se encontraban al descubierto, permitiendo el paso del aire. Las mediciones se hicieron entre las 20:07 h. del día 29/06/2022 y las 19:35 h. del día siguiente.

Durante el paso del tiempo, el prototipo fue expuesto directamente al sol, debido a que las condiciones meteorológicas lo permitían en ese momento. A continuación, se exponen las diferentes franjas horarias en las que se alteró la situación de la estación:

- A las 20:07 h. del día 29/06/2022 hasta las 14:14 h. del día 30/06/2022, la estación se encontraba en la zona cubierta.
- Desde las 14:14 h. del día 30/06/2022 hasta las 16:46 h. del día 30/06/2022, se produjo el movimiento de la misma a la zona exterior, expuesta directamente al sol.
- Finalmente, desde las 16:46 h. hasta las 19:35 h. del día 30/06/2022, volvió a colocarse en el área protegida de la lluvia.

7.2. INCIDENTES EN LA MEDICIÓN

Mientras se realizaron las mediciones, se produjeron una serie de fallos en el sistema, que se manifiestan a continuación:

- Desde las 21:06 h. del 29/06/2022 hasta las 7:49 h. del 30/06/2022, se produjo una parada en la estación meteorológica, posiblemente ocasionada por un problema en la alimentación. De cara al futuro, se sugiere instalar electrónica de regulación sencilla a la entrada de la Arduino que permita mantener constante el voltaje de alimentación. Otra de las soluciones es la incorporación de una fuente conmutada.
- Otra dificultad encontrada fueron las falsas pulsaciones de los botones, es decir, el sistema detectaba una activación del pulsador de parada cuando realmente no se producían. Esto se sabe porque en el archivo *json* se indica un corchete cuando la acción de parada ha sido realizada. Para solucionarlo, se recomienda aislar las tierras de los pulsadores respecto a los demás elementos del sistema. A continuación, se muestran las diferentes paradas ocasionadas:
 - De 7:49 h. a 8:46 h. del 30/06/2022.
 - De 16:12 h. a 16:46 h. del 30/06/2022.
 - De 18:32 h. a 19:35 h. del 30/06/2022.

La reanudación se produjo manualmente siguiendo el protocolo de actuación que se describe en el apartado 6.4 Manual de usuario.

7.3. MANIPULACIÓN DE LOS DATOS

Debido a los fallos ocurridos en la estación durante el periodo completo de medición, para la representación y el estudio de los datos recogidos se ha decidido escoger los valores medidos en el periodo comprendido entre las 8:46 h. y las 18:32 h. del día 30/06/2022.

Las condiciones climáticas sufrieron variaciones importantes durante la recopilación de los datos. Cuando la estación comenzó las mediciones, en la zona de medición había niebla o nubes bajas y viento. A medida que pasaron las horas, las nubes desaparecieron y disminuyó el viento, quedando un día soleado. Sin embargo, al avanzar la tarde, aparecieron de nuevo las condiciones adversas. Estos cambios en las condiciones meteorológicas se pueden apreciar en los resultados que se han obtenido.

Los resultados que se han consultado para realizar el análisis de los mismos, han sido recogidos en el fichero “.csv”. Este formato ha permitido elaborar una tabla con todos los datos y obtener las gráficas. En cuanto al fichero “.json”, no se ha utilizado para analizar los datos, pero se ha generado con el objetivo de incluirlos en la aplicación de las estaciones. A continuación, se muestra un ejemplo del formato *json* de un ciclo de medidas (Fig. 61):

```
[
{
"Fecha": 30/6/2022,
"Hora": 19:35:11,
"Temperatura(°C)": 18.45,
"Humedad relativa(%)": 65.78,
"Calidad del aire": "Buena calidad",
"Precipitaciones acumuladas(L/m2)": 20.47,
"Intensidad UV(mw/cm2)": 0.26,
"Temperatura BMP(°C)": 18.38,
"Presion(mbar)": 954.85,
"Altitud(m)": 497.92,
"Humedad del suelo(%)": 86.79,
"Velocidad del viento(km/h)": 35.65,
"Observaciones": "Sin errores"
}
]
```

Figura 61: Extracto del archivo generado en formato *json*.

En la figura que se muestra a continuación (Fig. 62) se representa la variación de temperatura producida en los diferentes tramos del día. Se han tenido en cuenta las temperaturas

registradas tanto por el sensor DHT22 como por el BMP280, pudiendo observar la pequeña variación entre los valores obtenidos por uno y otro sensor.

En el rango de tiempo donde la estación se colocó en el exterior, se registraron mayores valores de temperatura. Esto coincide con la mejora de las condiciones climáticas. Sin embargo, ambos sensores registraron valores demasiado altos respecto a los valores reales. Se deduce entonces que dichos sensores no pueden estar expuestos directamente al sol.

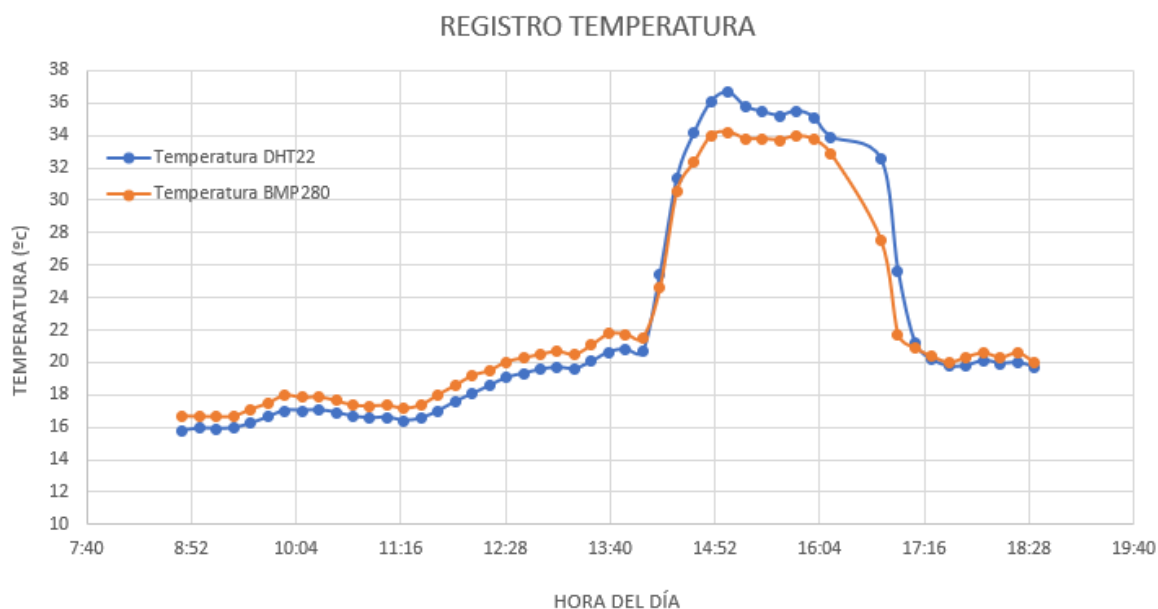


Figura 62: Gráfica comparativa de la temperatura de los sensores BMP280 y DHT22.

A continuación, la gráfica humedad relativa que se expone en la Figura 63, hace referencia al vapor de agua que se encuentra en un volumen de aire determinado. Como se ha comentado anteriormente, durante el inicio del periodo de medición, las condiciones climáticas eran adversas: nublado, con niebla y viento. Con el avance de la mañana, las condiciones mejoraron bastante, resultando en un día soleado y caluroso.

Dicho comportamiento se puede visualizar en la gráfica de humedad relativa, obteniendo valores altos de humedad relativa durante las primeras horas de medición, situados entre el 80 y el 90 %. Sobre las 14:00 h. aproximadamente, la humedad relativa sufrió una disminución progresiva hasta el 60 %. Fue entonces cuando las condiciones climáticas permitieron situar la estación en el exterior. Al realizar este cambio, la humedad relativa sufrió una disminución bastante relevante, llegando a valores por debajo del 30 %.



Figura 63: Gráfica del registro humedad relativa del aire.

Para determinar si la velocidad del viento obtenido por nuestro anemómetro es correcta, se ha incorporado una gráfica de la página web *Weather Spark*, donde se muestra la velocidad real del viento el día 30/06/2022 (Fig. 64). Comparando los valores reales con los valores obtenidos con nuestra estación (Fig. 65), podemos observar que desde que comenzaron las mediciones hasta que se situó la estación en el exterior, los valores recogidos de velocidad del viento se encuentran entre 15 y 25 km/h aproximadamente. Los valores que muestra *Weather Spark* se encuentran entre 20 y 27,5 km/h, aunque aportando un intervalo de variación de entre 15 y 30 km/h. El ligero aumento que sufrió la velocidad del viento registrado por *Weather Spark* entre las 16:00 h. y las 20:00 h. también se registró mediante la estación en el mismo periodo de tiempo.

Se puede concluir que, salvo las variaciones que surgieron al colocar la estación directamente al exterior, el anemómetro mide la velocidad del viento de forma óptima.

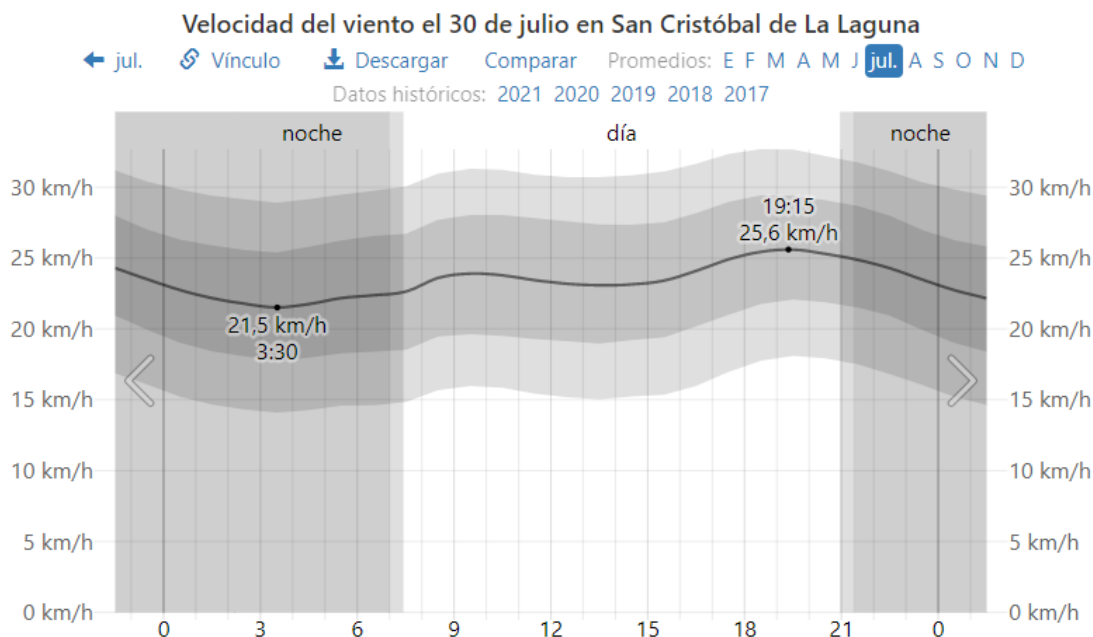


Figura 64: Velocidad del viento el día de la medición [57].



Figura 65: Gráfica del registro de la velocidad del viento.

La figura que se encuentra a continuación (Fig. 66), hace referencia a la radiación UV. Como se puede examinar en los resultados obtenidos, la colocación del sensor es de suma importancia, debiendo estar situado directamente al sol, donde se ocasiona una mejora significativa de sus capacidades. Únicamente se recogieron valores de radiación ultravioleta en

el periodo de tiempo en el que la estación meteorológica se situó en el exterior, como cabía de esperar. Para este periodo se obtuvieron valores de radiación ligeramente superiores a los 3 mW/cm².

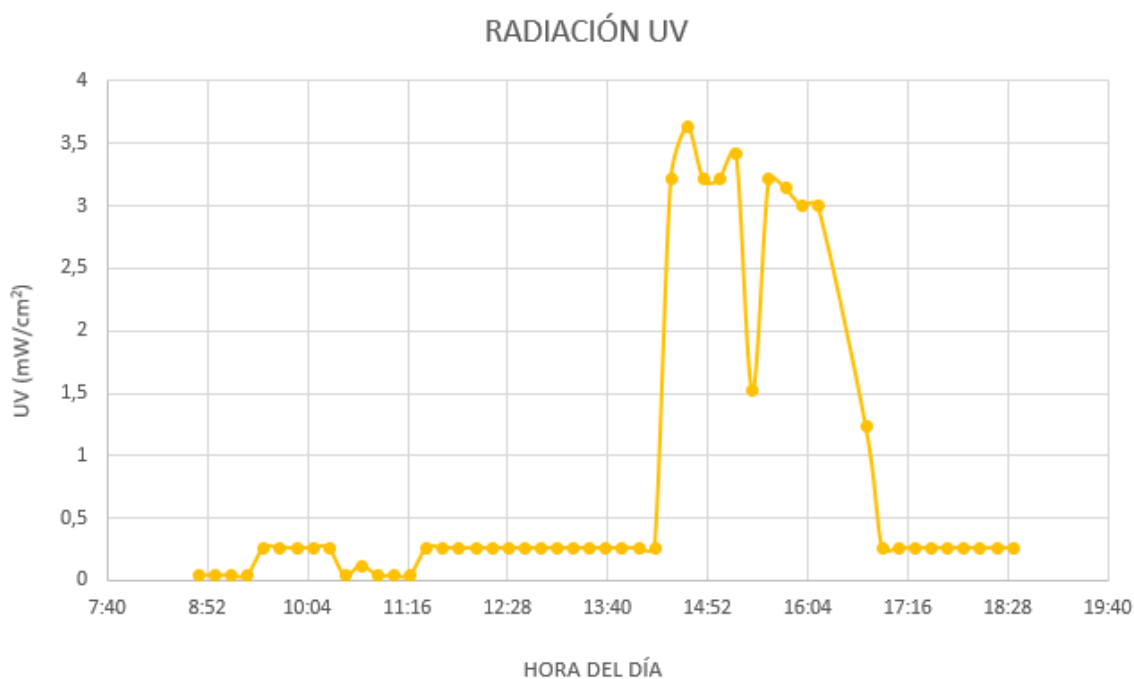


Figura 66: Registro radiación UV.

Respecto a las demás variables medidas no se ve necesaria su representación gráfica, debido a que no presentaron cambios importantes. Así pues, se hace un resumen:

- Calidad del aire: en la calidad del aire no se registraron variaciones, únicamente se obtuvo una “buena calidad”, que es el rango que se encuentra entre los 0 y los 250 ppm. Esto se debe a que la zona de medición estaba alejada de focos de contaminación como la autopista o polígonos industriales.
- Presión atmosférica: los valores que se registraron de presión atmosférica apenas sufrieron variaciones durante el periodo de medición, resultando de escaso interés.
- Altitud: la altitud es calculada por el sensor BMP280 en función de los valores que detecta de presión atmosférica, por lo que tampoco se ha incluido su representación gráfica.
- Precipitaciones: se registró un valor nulo de precipitaciones, ya que en el periodo de tiempo en el que se realizó la medición no se produjeron.
- Humedad del suelo: Como el sensor de humedad del suelo es anticorrosivo, fue el único que en todo momento estuvo colocado en una maceta con tierra a la intemperie. Se han

recogido valores entre el 80 y el 87 % de humedad del suelo, lo que demuestra que la niebla mantenía la tierra bastante húmeda.

8. FUTURAS MEJORAS

El diseño presenta las bases para desarrollar el prototipo final, pudiéndose modificar y mejorar diferentes aspectos, y, sobre todo, la inclusión de nuevos elementos que amplíen las funcionalidades de la estación.

8.1. SOLUCIÓN A LOS FALLOS OBTENIDOS

Como se comentó en el apartado de los resultados, durante las mediciones se produjeron errores tanto en la alimentación como en los pulsadores, por tanto, es necesario aportar soluciones a dichos problemas.

- Para evitar picos de voltajes o de corriente que produzcan falsas pulsaciones de los botones de parada y de inicio, se recomienda aislar las tierras de los pulsadores respecto a los demás elementos del sistema.
- Como solución al fallo de la alimentación, si no se dispone de una fuente conmutada, se necesita la incorporación de electrónica de regulación, basada en un diodo Zener o similar, para que, cuando la corriente en la entrada del Arduino cambie, el voltaje de salida permanezca constante.

8.2. NUEVAS MAGNITUDES

Para una mejora en la aplicación de la estación en el sector agrícola, sería posible incluir dos variables al diseño del prototipo.

8.2.1. PH del agua

El pH es una medida que se realiza al agua para determinar el grado de acidez o alcalinidad. Normalmente, los valores se establecen en una escala del 0 al 14, donde los valores por debajo del 7 establecen que el agua es más ácida y valores superiores, que es más alcalina.

Para medir el pH se puede utilizar un electrodo sensible a los H⁺ (pH - metro) o también por el uso de indicadores colorimétricos de pH, que permiten conocer el nivel de pH según el color [59].

Utilidad de medir el pH

Uno de los principales efectos que afecta notablemente al cultivo es la lluvia ácida, debido a que un bajo nivel de pH, es decir, mayor porcentaje de ácido influye en la disponibilidad de los nutrientes para las plantas. Esto puede ocasionar deficiencia, toxicidad o que las plantas no se encuentren en niveles adecuados impidiendo la correcta absorción de los nutrientes. Se puede mejorar el nivel de acidificación con la aplicación de cal.

Por otro lado, valores extremos del pH, superiores a 7, afectan a la estructura del suelo. Finalmente, para poder obtener el máximo rendimiento y productividad de la planta, se recomienda mantener un pH entre 5.5 y 7 [59].

8.2.2. Humedad de las Hojas

La humedad de las hojas es el parámetro que da cuenta de la humedad presente en la superficie de las hojas de una planta. La humedad presente en las hojas puede darse por diferentes causas, como la lluvia o el rocío, y está relacionado de forma muy estrecha con la existencia de enfermedades en las plantas [3].

Es posible medir la humedad de las hojas mediante sensores específicos para ello, conocidos como sensores de humectación de hoja. Estos sensores se colocan tanto en la epidermis superior como en la epidermis inferior de la hoja, simulando tanto la forma como la textura de una hoja real. Detectan la presencia de rocío y su duración, además permiten obtener la cantidad de agua en la superficie de la hoja. Como ejemplo de sensor que permite medir la humectación de las hojas se puede mencionar el modelo PHYTOS 31, que permite obtener las medidas de forma precisa y sencilla [60].

Utilidad de medir la humectación de las hojas

Medir el grado de humectación en las hojas de las plantas resulta de especial interés debido a la relación existente entre este parámetro y la presencia de enfermedades e infecciones debidas a hongos en las plantas. De esta forma, teniendo un control de dicha variable, es posible predecir las posibles enfermedades que pudieran aparecer y atajarlas mediante los distintos tratamientos existentes [3].

La presencia de enfermedades puede mermar gravemente el rendimiento de los cultivos, pudiendo llevar a las plantas a su muerte, y además, los tratamientos fitosanitarios tienen un elevado coste económico. Es por ello que medir la humedad en las hojas es de gran utilidad, ayudando a predecir y reducir el riesgo de que surjan las enfermedades [60].

8.3. COMUNICACIÓN CON EL SERVIDOR Y ENTRE LOS DISPOSITIVOS

8.3.1. Módulo wifi ESP8266 12F y ThingSpeak

La parte de comunicación con el servidor es de gran relevancia, consiguiendo transmitir los datos recogidos a la plataforma *ThingSpeak*, para su posterior exposición en la página web del Trabajo de Fin de Grado, “Aplicación fullstack Mi Finquita”, desarrollada por Don Javier Martín de León, del Grado en Ingeniería Informática.

En primer lugar, se necesita el módulo wifi ESP8266 12F, que requiere una programación concreta. Sin embargo, es relevante conocer la necesidad de utilizar otros componentes para que el módulo wifi funcione correctamente.

- Incorporación de una alimentación externa, debido a que el ESP8266 12F utiliza una alimentación de 3,3 V con una corriente de 500 mA. Esta demanda de corriente impide conectar el módulo directamente a la alimentación que proporciona la Arduino, que puede proporcionar un máximo de 50 mA.
- Adquisición de un convertidor de nivel lógico de 3,3 a 5 V. Como en la placa de desarrollo Arduino Mega 2560 los pines digitales tienen un nivel lógico de 5 V y el módulo wifi de 3,3 V, es imprescindible la utilización de dicho elemento.

Finalmente, se procederá a la creación de un canal en la plataforma *ThingSpeak*, aplicación que nos permite almacenar y recopilar los datos recogidos de la estación, sin

necesidad de crear servidores o desarrollar software web. Además, se puede utilizar Matlab¹ para interpretar los valores.

8.3.2. ZigBee

La comunicación inalámbrica es un punto de interés a desarrollar en un futuro. Dicha tecnología conseguiría que extensiones agrarias con una superficie considerable, o fincas a diferentes niveles o con diferentes cultivos, se pudieran colocar sensores a distintas distancias con respecto al módulo principal con el fin de recopilar información necesaria para una parcela específica.

El dispositivo ideal para dicha función es el *Xbee*, módulo de comunicación inalámbrico que utiliza el protocolo de comunicación de radio *ZigBee*. Su bajo consumo y su longitud de hasta 1000 metros hace idóneo al componente para futuras mejoras [61].

9. BIBLIOGRAFÍA

- [1] www.inia.org.uy/disciplinas/agroclima/le/cc_tempe.htm
- [2] M. Carmen Casas Castillo y Marta Alarcón Jordán, Ediciones UPC, Meteorología y clima, 1999.
- [3] <https://docplayer.es/13303312-Uso-de-estaciones-meteorologicas-en-la-agricultura.html>
- [4] [HF Riego Excel: Evapotranspiración - Evapotranspiración de referencia \(ETO\) \(hidraulicafacil.com\)](http://HF.Riego.Excel:Evapotranspiración-Evapotranspiración-de-referencia(ETO)(hidraulicafacil.com))
- [5] [Deteccion Oasis Orograficos.pdf \(juntadeandalucia.es\)](http://Deteccion.Oasis.Orograficos.pdf(juntadeandalucia.es))
- [6] [Presión atmosférica - Netatmo](http://Presión.atmosférica-Netatmo)
- [7] [Meteorología - Presión atmosférica - Taller Virtual de Meteorología y Clima. \(ucm.es\)](http://Meteorología-Presión.atmosférica-Taller.Virtual.de.Meteorología.y.Clima.(ucm.es))
- [8] [Presión atmosférica - Brioagro](http://Presión.atmosférica-Brioagro)
- [9] [La humedad del aire y su medida \(tiempo.com\)](http://La.humedad.del.aire.y.su.medida(tiempo.com))
- [10] [7. La humedad del aire \(cenicafe.org\)](http://7.La.humedad.del.aire(cenicafe.org))
- [11] [HUMEDAD DEL SUELO | Biología de suelos \(wordpress.com\)](http://HUMEDAD.DEL.SUELO|Biología.de.suelos(wordpress.com))

¹ En el siguiente vídeo se puede encontrar información del funcionamiento de la plataforma: <https://es.mathworks.com/videos/iot-from-data-to-action-part-1-getting-started-with-thingspeak-1633414543881.html>

- [12] [Humedad del suelo. Cómo se comporta y su importancia - Agromática \(agromatica.es\)](#)
- [13] [Medición del contenido de humedad - Wikipedia, la enciclopedia libre](#)
- [14] [Viento, Definición y Características - BioEnciclopedia](#)
- [15] <https://www.agronewscastillayleon.com/beneficios-de-una-estacion-meteorologica-en-la-agricultura>
- [16] <http://www.mambiente.munimadrid.es/opencms/calair/ContAtmosferica/Contaminantes/Efectos/plantas.html>
- [17] <https://www.iberdrola.com/compromiso-social/radiacion-solar>
- [18] http://www.ece.uprm.edu/~m_goyal/agroclimatologia.pdf
- [19] <https://www.intagri.com/articulos/horticultura-prottegida/importancia-de-la-radiacion-solar-en-la-produccion-bajo-invernadero#:~:text=La%20radiaci%C3%B3n%20solar%20se%20puede,crecimiento%20y%20desarrollo%20del%20cultivo>
- [20] [Precipitación \(meteorología\) - Wikipedia, la enciclopedia libre](#)
- [21] [Cómo se miden las precipitaciones en meteorología \(aristasur.com\)](#)
- [22] <https://www.aprendiendoarduino.com/cursos/historico-de-cursos/arduino-avanzado-2016/>
- [23] [▷ Tutorial UART para Arduino y ESP8266 \(2022\) \(elosciloscopio.com\)](#)
- [24] [▷ Tutorial I2C para Arduino, ESP8266 y ESP32 \(2022\) \(elosciloscopio.com\)](#)
- [25] [▷ Tutorial SPI para Arduino y ESP8266 \(2022\) \(elosciloscopio.com\)](#)
- [26] [SketchUp Make 2017 - Free download and software reviews - CNET Download](#)
- [27] [Ultimaker Cura: software de impresión 3D potente y fácil de usar | Ultimaker](#)
- [28] [Fritzing 0.9.3b - Descargar para PC Gratis \(malavida.com\)](#)
- [29] [Velleman VMA101 ATmega2560 Mega - moduł Botland - Sklep dla robotyków](#)
- [30] Datasheet de la placa Arduino Mega. Velleman. Fecha de consulta: 20/03/2022.
- [31] [Esquema de patillaje \(pinout\) de Arduino Uno, Nano, Mini y Mega \(luisllamas.es\)](#)
- [32] [Alimentar el Arduino: La guía definitiva - Geek Factory](#)
- [33] Datasheet del sensor DHT22. Aosong. Fecha de consulta: 21/03/2022.
- [34] <https://naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html>
- [35] [Cómo utilizar el sensor de gas MQ-135 - Codrey Electronics](#)
- [36] Datasheet del sensor MQ135. Hanwei Electronics CO. Fecha de consulta: 25/03/2022.
- [37] <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- [38] Datasheet del sensor HC-SR07. Elec Freaks. Fecha de consulta: 19/03/2022.
- [39] [Sensor luz ultravioleta GY-ML8511 | MCI Electronics](#)
- [40] Datasheet del sensor ML8511. Lapis. Fecha de consulta: 20/03/2022.
- [41] [SPI con Arduino y sensor BMP280 temperatura y presión \(programarfacil.com\)](#)
- [42] <https://www.luisllamas.es/medir-temperatura-y-presion-barometrica-con-arduino-y-bmp280/>
- [43] Datasheet del sensor BMP280. Bosch. Fecha de consulta: 20/03/2022.
- [44] <https://www.vistronica.com/sensores/humedad/modulo-sensor-de-humedad-del-suelo-hd-38-detail.html>

- [45] Datasheet del sensor HD-38. Texas Instruments. Fecha de consulta: 21/03/2022.
- [46] <https://electronilab.co/tienda/sensor-de-humedad-de-suelo-higrometro/#:~:text=Este%20sensor%20de%20humedad%20puede,seguimiento%20de%20un%20jard%C3%ADn%20urbano>
- [47] <https://naylampmechatronics.com/sensores-proximidad/240-sensor-encoder-infrarrojo-fc-03-moch22a.html>
- [48] Datasheet del sensor FC-08. Junye. Fecha de consulta: 5/04/2022.
- [49] <https://controlautomaticoeducacion.com/arduino/lcd/>
- [50] Datasheet de la pantalla LCD. Hitachi. Fecha de consulta: 28/03/2022.
- [51] <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>
- [52] Datasheet del módulo para tarjeta micro SD. Sunrom. Fecha de consulta: 30/03/2022.
- [53] <https://programarfacil.com/blog/arduino-blog/reloj-con-arduino-rtc/>
- [54] <https://www.geekfactory.mx/tutoriales-arduino/ds1307-en-tinyrtc-con-arduino/>
- [55] Datasheet RTC DS1307. Maxim. Fecha de consulta: 5/04/2022.
- [56] [Arduino y el modo Sleep | Tienda y Tutoriales Arduino \(prometec.net\)](#)
- [57] [Valores climatológicos normales - Agencia Estatal de Meteorología - AEMET. Gobierno de España](#)
- [58] <https://es.weatherspark.com/d/31608/7/30/Tiempo-promedio-el-30-de-julio-en-San-Crist%C3%B3bal-de-La-Laguna-Espa%C3%B1a#Figures-CloudCover>
- [59] http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0568-25172009000300003#:~:text=En%20general%2C%20el%20pH%20%C3%B3ptimo,et%20al.%2C%202005
- [60] [Sensor de humectación de hoja PHYTOS 31 \(lab-ferrer.com\)](#)
- [61] <https://aprendiendoarduino.wordpress.com/2016/11/16/zigbeexbee/>



ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Grado en Ingeniería Electrónica Industrial y Automática

IV. PRESUPUESTO Y MEDICIONES

TRABAJO FIN DE GRADO

Título:

**ESTACIÓN METEOROLÓGICA DE BAJO
COSTE PARA EL ENTORNO AGRÍCOLA**

Presupuesto y mediciones

Para la realización de los presupuestos, se ha dividido en dos capítulos. Por un lado, “Capítulo I: Instalación Electrónica”, donde únicamente se encuentra una unidad de obra, “UD Componentes”. El segundo capítulo, “Capítulo II: Diseño exterior de la estación meteorológica”, con 4 unidades de obra: “UD Carcasas en impresión 3D”, “UD: Anemómetro”, “UD: Pluviómetro” y “UD: Caja estanca”.

Capítulo I: Instalación electrónica

UD COMPONENTES

Los componentes se encuentran conectados a la placa de desarrollo Atmega 2560 WPB101 a través de los cables para Arduino y protoboard. Posteriormente se realiza la programación de los mismos.

Código	Unidad	Descripción	Rendimiento	Precio Unitario	Importe
1		<u>Materiales</u>			
	Ud	Sensor de temperatura y humedad relativa DHT22	1	5,35 €	5,35 €
	Ud	Sensor calidad del aire MQ135	1	3,50 €	3,50 €
	Ud	Sensor ultrasonido HC-SR07	1	2,70 €	2,70 €
	Ud	Sensor radiación UV ML8511	1	7,50 €	7,50 €
	Ud	Sensor presión, temperatura y altitud BMP280	1	4,00 €	4,00 €
	Ud	Sensor humedad del suelo HD-38	1	7,60 €	7,60 €
	Ud	Sensor infrarojos FC-03	1	2,20 €	2,20 €
	Ud	Pantalla LCD	1	5,35 €	5,35 €
	Ud	Módulo adaptador para tarjeta micro SD	1	2,20 €	2,20 €
	Ud	RTC DS1307 con batería incluida	1	4,80 €	4,80 €
	Ud	Juego 40 cables 30 cm M-M uso Arduino	1	4,15 €	4,15 €
	Ud	Juego 40 cables 30 cm macho-hembra uso Arduino	1	4,15 €	4,15 €
	Ud	Protoboard 830 agujeros	1	3,90 €	3,90 €
	Ud	Cable USB para arduino	1	1,00 €	1,00 €
	Ud	Pulsadores	2	0,50 €	1,00 €
	Ud	Placa desarrollo Atmega 2560 WPB101	1	21,40 €	21,40 €
	Ud	Resistencia de 10 kΩ	3	0,22 €	0,66 €
	Ud	Tarjeta micro SD 32 GB	1	8,99 €	8,99 €

				Subtotal materiales:	90,45 €
2		<u>Manos de obra</u>			
	h	Ingeniero técnico industrial	110	25,00 €	2.750,00 €
	h	Ingeniero técnico industrial	110	25,00 €	2.750,00 €
				Subtotal mano de obra:	5.500,00 €
3		<u>Costes directos complementarios</u>			
	%		2	5.590,45 €	111,81 €
				Costes directos (1+2+3):	5.702,26 €

Capítulo II: Diseño exterior de la estación meteorológica

UD: Carcasas en impresión 3D

Se diseñan las protecciones en el programa SketchUp y se imprimen en las impresoras 3D

Código	Unidad	Descripción	Rendimiento	Precio Unitario	Importe
1		<u>Materiales</u>			
	Ud	Filamento PLA Starfil	0,7	16,70 €	11,69 €
	Ud	Laca en spray	0,7	2,00 €	1,40 €
	Ud	Líquido limpiacristal	0,7	1,10 €	0,77 €
	Ud	Tornillos M2,5 x 16	20	0,04 €	0,80 €
	Ud	Tuercas M2,5	20	0,04 €	0,80 €
				Subtotal materiales:	15,46 €
2		<u>Equipamiento y maquinaria</u>			
	h	Impresora 3D BQ	19	5,00 €	95,00 €
				Subtotal equipamiento y maquinaria:	95,00 €
3		<u>Mano de obra</u>			
	h	Ingeniero técnico industrial	60	25,00 €	1.500,00 €
	h	Ingeniero técnico industrial	60	25,00 €	1.500,00 €
				Subtotal mano de obra:	3.000,00 €

4	<u>Costes directos complementarios</u>			
	%		2	3.110,46 €
				62,21 €
			Costes directos	3.172,67
			(1+2+3+4):	€

**Capítulo II: Diseño exterior de la estación
meteorológica**

UD: Anemómetro

Se diseña las partes del anemómetro (aspas, cúpula, tubo y soporte, y, eje con encoder) y se imprimen con las impresoras 3D.

Código	Unidad	Descripción	Rendimiento	Precio Unitario	Importe
1		<u>Materiales</u>			
	Ud	Filamento PLA Starfil	0,3	16,70 €	5,01 €
	Ud	Laca en spray	0,3	2,00 €	0,60 €
	Ud	Líquido limpiacristal	0,3	1,10 €	0,33 €
				Subtotal materiales:	5,94 €
2		<u>Equipamiento y maquinaria</u>			
	h	Impresora 3D BQ	9	5,00 €	45,00 €
	h	Impresora 3D Creality	5,5	5,00 €	27,50 €
				Subtotal equipamiento y maquinaria:	72,50 €
3		<u>Mano de obra</u>			
	h	Ingeniero técnico industrial	10	25,00 €	250,00 €
	h	Ingeniero técnico industrial	10	25,00 €	250,00 €
				Subtotal mano de obra:	500,00 €
4		<u>Costes directos complementarios</u>			
	%		2	578,44 €	11,57 €
			Costes directos		590,01
			(1+2+3+4):		€

**Capítulo II: Diseño exterior de la estación
meteorológica**

UD: Pluviómetro

Construcción, pegado y sellado del recipiente de metacrilato.

Código	Unidad	Descripción	Rendimiento	Precio Unitario	Importe
1		<u>Materiales</u>			
	Ud	Planca metacrilato de 100 cm x 30 cm x 3 mm	1	21,52 €	21,52 €
	Ud	Pegamento especial metacrilato	1	2,30 €	2,30 €
	Ud	Silicona transparente (polimero MS)	2	5,71 €	11,42 €
	Ud	España para lijar	1	1,10 €	1,10 €
		<u>Subtotal materiales:</u>			36,34 €
2		<u>Mano de obra</u>			
	h	Ingeniero técnico industrial	5	25,00 €	125,00 €
	h	Ingeniero técnico industrial	5	25,00 €	125,00 €
		<u>Subtotal mano de obra:</u>			250,00 €
3		<u>Costes directos complementarios</u>			
	%		2	286,34 €	5,73 €
		Costes directos (1+2+3):			292,07 €

Capítulo II: Diseño exterior de la estación meteorológica

UD: Caja estanca

Construcción, pegado y sellado del recipiente de metacrilato.

Código	Unidad	Descripción	Rendimiento	Precio Unitario	Importe
1		<u>Materiales</u>			
	Ud	Caja Rect. GW 4408 GEWIS 240x190x90 mm IP55	1	10,45 €	10,45 €
	Ud	Velcro	2	1,30 €	2,60 €

	Ud	Plancha metacrilato 20x20 cm	1	2,90 €	2,90 €
	Ud	Alimentación (AC/DC) 9V a 1A	1	12,85 €	12,85 €
	Ud	Tornillos M3 x 20	4	0,60 €	2,40 €
				Subtotal materiales:	31,20 €
2		Mano de obra			
	h	Ingeniero técnico industrial	15	25,00 €	375,00 €
	h	Ingeniero técnico industrial	15	25,00 €	375,00 €
				Subtotal mano de obra:	750,00 €
3		Costes directos complementarios			
	%		2	781,20 €	15,62 €
				Costes directos (1+2+3+4):	796,82 €

RESUMEN PRESUPUESTO

Capítulo I: Instalación electrónica	5.702,26 €
Capítulo II: Diseño exterior de la estación meteorológica	4.856,57 €
Presupuesto de Ejecución Material	10.558,83 €
16 % Gastos Generales	1.689,41 €
6 % Beneficio Industrial	633,53 €
Presupuesto de Contrata	12.881,77 €
7 % IGIC	901,72 €
Presupuesto Total	13.783,50 €



ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Grado en Ingeniería Electrónica Industrial y Automática

V. CONCLUSIONES

TRABAJO FIN DE GRADO

Título:

**ESTACIÓN METEOROLÓGICA DE BAJO
COSTE PARA EL ENTORNO AGRÍCOLA**

Conclusiones

Se ha logrado elaborar un prototipo funcional de estación meteorológica que ha permitido registrar las variables agroclimáticas deseadas, permitiendo la manipulación y el análisis posterior de los datos recogidos.

Una vez finalizado el prototipo, comprobando su correcto funcionamiento en condiciones de laboratorio, se ha puesto en marcha en condiciones reales en un terreno privado. Tras cierto periodo de tiempo en el que la estación ha recabado los valores de todas las magnitudes, se ha procedido a analizar dichos datos. Cabe señalar la ocurrencia de distintos incidentes durante el periodo de medición, produciéndose paradas indeseadas. A pesar de ello, se ha obtenido datos suficientes para poder estudiar el comportamiento de los sensores y verificar la fiabilidad de los valores recogidos.

El desarrollo del proyecto ha dado lugar a un prototipo de estación meteorológica, lo que implica la posibilidad de ampliar y mejorar el funcionamiento del mismo. Son numerosos los progresos que se pueden lograr implementando las distintas mejoras mencionadas, perfeccionando el funcionamiento actual del prototipo, permitiendo la medición de otros parámetros de interés o mejorando el sistema de registro y transmisión de datos.

Este proyecto nos ha servido para poner en práctica todas las capacidades adquiridas a lo largo del grado en Ingeniería Electrónica Industrial y Automática, además de aplicar las competencias específicas, transversales, generales y básicas de las haciendo hincapié en el trabajo en equipo, la resolución de problemas de forma autónoma, la toma de decisiones y el desarrollo de las habilidades necesarias para efectuar futuros proyectos técnicos.



ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Grado en Ingeniería Electrónica Industrial y Automática

VI. CONCLUSIONS

TRABAJO FIN DE GRADO

Título:

**ESTACIÓN METEOROLÓGICA DE BAJO
COSTE PARA EL ENTORNO AGRÍCOLA**

Conclusions

A functional prototype of a meteorological station has been developed which has been able to record the desired agro-climatic variables, allowing the subsequent manipulation and analysis of the data collected.

Once the prototype was completed and had been checked for its correct operation in laboratory conditions, it was put into operation in real conditions on private property. After a certain period in which the station collected the values of all the magnitudes, the data were analysed. It should be noted that various incidents occurred during the measurement period, leading to undesired stoppages. Nevertheless, sufficient data was obtained to study the behaviour of the sensors and verify the reliability of the values collected.

The development of the project has resulted in a prototype meteorological station, which implies the possibility of extending and improving its operation. Numerous advances can be achieved by implementing the variations mentioned during the work, improving the current operation of the prototype, allowing the measurement of other parameters of interest, and the quality of the data recording and transmission system.

This project has allowed us to put into practice all the skills acquired throughout the degree in Industrial Electronics and Automation Engineering, as well as to apply the specific, transversal, general and basic competences of the degree, emphasising teamwork, autonomous problem solving, decision making and the development of the necessary skills to carry out future technical projects.



ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Grado en Ingeniería Electrónica Industrial y Automática

VII. ANEXOS

TRABAJO FIN DE GRADO

Título:

**ESTACIÓN METEOROLÓGICA DE BAJO
COSTE PARA EL ENTORNO AGRÍCOLA**

Anexo I: Archivos de los diseños 3D

En el siguiente enlace se encuentra el repositorio con los diseños de los elementos en 3D en formato “.gcode”, “.obj”, “.stl” y “.skp”:

https://drive.google.com/drive/u/1/folders/1hUAuXqsy7buNVE0tAAIMvrw7_EKR_4Jo

Anexo II: Programación

```

// TFG ESTACIÓN METEOROLÓGICA

// Autores: Hugo Hilario Reyes
//          Luis Vallejo Segovia

// Tutor: Santiago Torres Álvarez

// LIBRERÍAS:

#include <RTCLib.h>           // Librería RTC DS1307 --> Fecha y hora.
#include <Adafruit_BMP280.h> // Librería BMP280 --> Presión, altitud y temperatura.
#include <SD.h>               // Librería SD.
#include <SPI.h>              // Librería comunicaciones SPI.
#include <LowPower.h>         // Librería Modo Sleep.
#include <LiquidCrystal_I2C.h> // Librería pantalla LCD con módulo I2C.
#include <Wire.h>             // Librería comunicaciones I2C.
#include <DHT.h>              // Librería DHT22 --> Temperatura y humedad relativa del aire.
#include <MQ135.h>           // Librería MQ-135 --> Calidad del aire.

// DHT22
#define DHTTYPE DHT22

// BMP280
Adafruit_BMP280 sensorBMP;

// RTC DS1307
RTC_DS1307 rtc;

// Declaramos las "variables" de tipo archivo (File).
File archivoCSV;
File archivoJSON;

// DEFINICIÓN DE PINES:

/* Pines módulo SD:
   Pin 50 digital para MISO (morado).
   Pin 51 digital para MOSI (gris).
   Pin 52 digital para SCK (marrón).
   Pin 53 digital para CS (blanco).
*/

const int Trigger = 47; //Pin digital 47 para el Trigger del sensor HC-SR04. -- NARANJA
const int Echo = 45;   //Pin digital 45 para el Echo del sensor HC-SR04. -- AMARILLO

int UVOUT = A1;        //Pin analógico A1 ML8511 (salida del sensor).
int REF_3V3 = A2;     //Pin analógico A2 ML8511 (3.3v de la arduino). -- VERDE

const int MQPin = A0; //Pin analógico A0 para el sensor MQ-135.
MQ135 sensor_MQ135(MQPin);

const int DHTPin = 22; //Pin digital 22 para el sensor DHT22.
DHT dht(DHTPin, DHTTYPE);

const int lectura_HD38 = A3; //Pin analógico A3 para el sensor HD-38.

// Se crea el objeto lcd con dirección 0x27 y 16 columnas x 2 filas.
LiquidCrystal_I2C lcd(0x27,16,2);

// DECLARACIÓN DE VARIABLES:

// Declaración de los errores.
String errores;

// Variable para registrar el número de interrupciones del anemómetro.
volatile byte num_int = 0;

// Variable para apagar y encender la estación.
int parada = 2;

// Variable para evitar repetición del corchete final en fichero JSON.
int memoria_corchete = 0;

void setup() {

  Serial.begin(9600);

  // Inicio de pantalla LCD:
  lcd.init(); // Inicializar el LCD.
  lcd.backlight(); // Encender la luz de fondo.
  lcd.clear();

  // Inicio de RTC y configuración de fecha y hora:
  rtc.begin();
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  Serial.print(__DATE__);
  Serial.print(" ");
  Serial.print(__TIME__);

  // Inicio del sensor dht:
  dht.begin();

```

```

// CONFIGURACIÓN PINES DIGITALES:

// Ultrasonidos HC-SR04:
pinMode(Trigger, OUTPUT); //pin configurado como salida.
pinMode(Echo, INPUT); //pin configurado como entrada.
digitalWrite(Trigger, LOW); //Inicializamos el pin a 0.

// ML8511:
pinMode(UVOUT, INPUT);
pinMode(REF_3V3, INPUT);

// GENERACIÓN COLUMNAS DEL FICHERO CSV EN LA TARJETA SD:

// SD (Si no se puede conectar, se detiene la medición):
Serial.println("Iniciando tarjeta micro SD...");
if (!SD.begin(53)) {
  Serial.println("No se pudo iniciar la tarjeta");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("No se pudo");
  lcd.setCursor(0, 1);
  lcd.print("iniciar tarjeta.");
  lcd.clear();
  delay(2000);
  lcd.setCursor(0, 0);
  lcd.print("Estacion parada");
  while(1);
}
Serial.println("Inicio de SD exitoso.");

if(SD.exists("Parte.csv") == false){
  Serial.println("Abriendo archivo para generar columnas...");
  archivoCSV = SD.open("Parte.csv", FILE_WRITE); //abrimos el archivo
  if (archivoCSV) {
    delay(1000);
    Serial.println("Generando columnas...");
    archivoCSV.print("Fecha,Hora,Temperatura(°C),Humedad relativa(%),Calidad del aire,Precipitaciones Acumuladas(L/m2),
      Intensidad UV(mw/cm2),Temperatura BMP(°C),Presion(mbar),Altitud(m),Humedad del suelo(%),Velocidad del Viento(km/h),Observaciones");
    archivoCSV.close(); //cerramos el archivo
    delay(1000);
  } else {
    Serial.println("Error al abrir el archivo.");
  }
} else {
  Serial.println("El archivo ya existe.");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.println("Archivo SD: OK");
  delay(2000);
}

// GENERACIÓN CORCHETE DEL FICHERO JSON:

if(SD.exists("ParteJ.txt") == false){
  Serial.println("Abriendo archivo para generar corchete...");
  archivoJSON = SD.open("ParteJ.txt", FILE_WRITE); //abrimos el archivo
  if (archivoJSON) {
    delay(1000);
    Serial.println("Generando corchete...");
    archivoJSON.println("[");
    archivoJSON.close();
    delay(1000);
  } else {
    Serial.println("Error al abrir el archivo.");
  }
} else {
  Serial.println("El archivo JSON ya existe.");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.println("Archivo SD: OK");
  delay(2000);
}
}

void loop() {

  // Habilitamos la interrupción de inicio:
  attachInterrupt(digitalPinToInterrupt(3),PULSADOR_INICIO,FALLING);
  delay(5000);

  // Reseteamos la variable para evitar efecto rebote:
  memoria_corchete = 0;

  // Inicio de pantalla LCD:
  lcd.init();
  lcd.backlight();
  lcd.clear();

  if (parada != 0){
    Serial.println("Pulse botón VERDE para inicializar.");
    lcd.setCursor(0, 0);
    lcd.print("Pulse boton");
    lcd.setCursor(0, 1);
    lcd.print("verde para ON.");
  }
}

```

```

    delay(5000);
}

while(parada == 0) {

    // Deshabilitamos las interrupciones:
    detachInterrupt(digitalPinToInterrupt(3));
    detachInterrupt(digitalPinToInterrupt(18));

    // Inicio de pantalla LCD:
    lcd.init();
    lcd.backlight();
    lcd.clear();

    // COMPROBACIÓN CONEXIÓN CON SENSORES:

    Serial.println("Realizando comprobaciones...");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Realizando");
    lcd.setCursor(0, 1);
    lcd.print("comprobaciones.");
    delay(3000);

    // RTC DS1307:
    if(!rtc.begin()){
        Serial.println("No se detecta el RTC.");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.println("Error EC.1");
        delay(2000);
        String errorrtc = "EC.1";
        errores = errores + errorrtc;
    }

    // DHT22:
    float hum = dht.readHumidity(); // Necesitamos declarar hum y temp para comprobar la comunicación.
    float temp = dht.readTemperature();
    if (isnan(hum) || isnan(temp)) {
        Serial.println("No se detecta el sensor DHT22.");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.println("Error EC.2");
        delay(2000);
        String errordht22 = "EC.2";
        errores = errores + errordht22;
    }

    // BMP280:
    if(!sensorBMP.begin()){
        Serial.println("No se detecta el sensor BMP280.");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.println("Error EC.3");
        delay(2000);
        String errorbmp = "EC.3";
        errores = errores + errorbmp;
    }

    // SIN ERRORES:
    if(errores.length() == 0){
        errores = "Sin errores";
    }

    // REALIZANDO MEDICIONES

    Serial.println("Realizando mediciones...");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Realizando");
    lcd.setCursor(0, 1);
    lcd.print("mediciones...");
    delay(3000);

    // LLAMADA A LAS FUNCIONES DE LOS SENSORES:

    // RTC:
    DateTime now = rtc.now();

    // DHT22:
    float resultado_temperatura = funcion_DHT22_temperatura ();
    float resultado_humedad = funcion_DHT22_humedad ();

    // MQ-135:
    //int adc_MQ = analogRead(MQPin);
    String resultado_calidad = funcion_MQ135 ();

    // Ultrasonidos HC-SR04:
    float resultado_precipitaciones = funcion_HCSR04 ();

    // ML8511:
    float resultado_radiacion = funcion_ML8511 ();

    // BMP280:
    float resultado_temperatura_BMP = funcion_BMP280_temperatura ();

```

```

float resultado_presion = funcion_BMP280_presion ();
float resultado_altitud = funcion_BMP280_altitud ();

// HD-38:
float resultado_humedad_suelo = funcion_HD38 ();

// FC-03:
float resultado_velocidad_viento = funcion_FC03 ();

// MOSTRAMOS TODOS LOS RESULTADOS POR SERIAL Y LCD:

// RTC:
Serial.print(now.day());
Serial.print('/');
Serial.print(now.month());
Serial.print('/');
Serial.println(now.year());
Serial.print(now.hour());
Serial.print(':');
Serial.print(now.minute());
Serial.print(':');
Serial.print(now.second());
Serial.println();

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Fecha: ");
lcd.setCursor(0, 1);
lcd.print(now.day());
lcd.print("/");
lcd.print(now.month());
lcd.print("/");
lcd.print(now.year());
delay(3000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Hora:");
lcd.setCursor(0, 1);
lcd.print(now.hour());
lcd.print(":");
lcd.print(now.minute());
lcd.print(":");
lcd.print(now.second());
delay(3000);

// DHT22:
Serial.print("Temperatura: ");
Serial.print(resultado_temperatura);
Serial.println(" °C");
Serial.print("Humedad: ");
Serial.print(resultado_humedad);
Serial.println(" %");

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Temp: ");
lcd.print(resultado_temperatura);
lcd.setCursor(0, 1);
lcd.print("Humedad: ");
lcd.print(resultado_humedad);
lcd.print(" %");
delay(3000);

// MQ-135:
Serial.print("Estado del aire: ");
Serial.println(resultado_calidad);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Estado del aire:");
lcd.setCursor(0, 1);
lcd.print(resultado_calidad);
delay(3000);

// Ultrasonidos HC-SR04:
Serial.print("Precipitaciones acumuladas: ");
Serial.print(resultado_precipitaciones);
Serial.println(" L/m2");

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Precip. acum.:");
lcd.setCursor(0, 1);
lcd.print(resultado_precipitaciones);
lcd.print(" L/m2");
delay(3000);

// ML8511:
Serial.print("Intensidad UV: ");
Serial.print(resultado_radiacion);
Serial.println(" mW/cm2");

lcd.clear();
lcd.setCursor(0, 0);

```

```

lcd.print("Intensidad UV:");
lcd.setCursor(0, 1);
lcd.print(resultado_radiacion);
lcd.print(" mW/cm2");
delay (3000);

// BMP280:
Serial.print("Temperatura: ");
Serial.print(resultado_temperatura_BMP);
Serial.println(" °C");
Serial.print("Presión: ");
Serial.print(resultado_presion);
Serial.println(" mbar");
Serial.print("Altitud: ");
Serial.print(resultado_altitud);
Serial.println(" m");

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Temperatura: ");
lcd.setCursor(0, 1);
lcd.print(resultado_temperatura_BMP);
delay(3000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Presion: ");
lcd.setCursor(0, 1);
lcd.print(resultado_presion);
lcd.print(" mbar");
delay(3000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Altitud: ");
lcd.setCursor(0, 1);
lcd.print(resultado_altitud);
lcd.print(" m");
delay(3000);

// HD-38:
Serial.print("Estado del suelo: ");
Serial.print(resultado_humedad_suelo);
Serial.println(" % húmedo.");

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Estado suelo:");
lcd.setCursor(0, 1);
lcd.print(resultado_humedad_suelo);
lcd.print(" %");
lcd.print(" húmedo");
delay(3000);

// FC-03:
Serial.print("Velocidad del viento: ");
Serial.print(resultado_velocidad_viento);
Serial.println(" km/h");

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Vel. del viento:");
lcd.setCursor(0, 1);
lcd.print(resultado_velocidad_viento);
lcd.print(" km/h");
delay(3000);

// Observaciones:
Serial.print("Observaciones: ");
Serial.printlnerrores);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Observaciones:");
lcd.setCursor(0, 1);
lcd.print(errores);
delay(3000);

// GUARDAMOS LOS DATOS MOSTRADOS EN LA TARJETA MICRO SD:
archivoCSV = SD.open("Parte.csv", FILE_WRITE); //abrimos el archivo

if (archivoCSV) {
  Serial.println("Guardando datos en la tarjeta micro SD... ");
  archivoCSV.println();
  archivoCSV.print(now.day());
  archivoCSV.print('/');
  archivoCSV.print(now.month());
  archivoCSV.print('/');
  archivoCSV.print(now.year());
  archivoCSV.print(",");
  archivoCSV.print(now.hour());
  archivoCSV.print(':');
  archivoCSV.print(now.minute());
  archivoCSV.print(':');
  archivoCSV.print(now.second());
}

```

```

archivoCSV.print(",");
archivoCSV.print(resultado_temperatura);
archivoCSV.print(",");
archivoCSV.print(resultado_humedad);
archivoCSV.print(",");
archivoCSV.print(resultado_calidad);
archivoCSV.print(",");
archivoCSV.print(resultado_precipitaciones);
archivoCSV.print(",");
archivoCSV.print(resultado_radiacion);
archivoCSV.print(",");
archivoCSV.print(resultado_temperatura_BMP);
archivoCSV.print(",");
archivoCSV.print(resultado_presion);
archivoCSV.print(",");
archivoCSV.print(resultado_altitud);
archivoCSV.print(",");
archivoCSV.print(resultado_humedad_suelo);
archivoCSV.print(",");
archivoCSV.print(resultado_velocidad_viento);
archivoCSV.print(",");
archivoCSV.print(errores);
archivoCSV.close();
delay(1000);
Serial.println("...datos guardados en micro SD ");
delay(2000);
} else {
Serial.println("Error al abrir el archivo");
}

// GENERAMOS EL ARCHIVO JSON:

archivoJSON = SD.open("ParteJ.txt", FILE_WRITE);

if (archivoJSON) {
Serial.println("Generando archivo json...");
archivoJSON.println();
archivoJSON.println("{}");
archivoJSON.print("\\"Fecha\": ");
archivoJSON.print(now.day());
archivoJSON.print('/');
archivoJSON.print(now.month());
archivoJSON.print('/');
archivoJSON.print(now.year());
archivoJSON.println(",");
archivoJSON.print("\\"Hora\": ");
archivoJSON.print(now.hour());
archivoJSON.print(':');
archivoJSON.print(now.minute());
archivoJSON.print(':');
archivoJSON.print(now.second());
archivoJSON.println(",");
archivoJSON.print("\\"Temperatura(°C)\": ");
archivoJSON.print(resultado_temperatura);
archivoJSON.println(",");
archivoJSON.print("\\"Humedad relativa(%)\": ");
archivoJSON.print(resultado_humedad);
archivoJSON.println(",");
archivoJSON.print("\\"Calidad del aire\": ");
archivoJSON.print("");
archivoJSON.print(resultado_calidad);
archivoJSON.print("");
archivoJSON.println(",");
archivoJSON.print("\\"Precipitaciones acumuladas(L/m2)\": ");
archivoJSON.print(resultado_precipitaciones);
archivoJSON.println(",");
archivoJSON.print("\\"Intensidad UV(mw/cm2)\": ");
archivoJSON.print(resultado_radiacion);
archivoJSON.println(",");
archivoJSON.print("\\"Temperatura BMP(°C)\": ");
archivoJSON.print(resultado_temperatura_BMP);
archivoJSON.println(",");
archivoJSON.print("\\"Presion(mbar)\": ");
archivoJSON.print(resultado_presion);
archivoJSON.println(",");
archivoJSON.print("\\"Altitud(m)\": ");
archivoJSON.print(resultado_altitud);
archivoJSON.println(",");
archivoJSON.print("\\"Humedad del suelo(%)\": ");
archivoJSON.print(resultado_humedad_suelo);
archivoJSON.println(",");
archivoJSON.print("\\"Velocidad del viento(km/h)\": ");
archivoJSON.print(resultado_velocidad_viento);
archivoJSON.println(",");
archivoJSON.print("\\"Observaciones\": ");
archivoJSON.print("");
archivoJSON.print(errores);
archivoJSON.println("");
archivoJSON.print("{}");
archivoJSON.close();
delay(1000);
Serial.println("...archivo generado.");
delay(2000);
} else {
Serial.println("Error al abrir el archivo");
}

```



```

// Habilitamos la interrupción de parada:
attachInterrupt(digitalPinToInterrupt(18),PULSADOR_PARADA,FALLING);

// Reseteamos la variable para evitar efecto rebote:
memoria_corchete = 0;

// ACTIVACIÓN MODO SLEEP (10 min):
for (int i = 0; i < 150; i++){
  LowPower.powerDown(SLEEP_4S, ADC_OFF, BOD_ON);
  lcd.noDisplay();
  lcd.noBacklight();
}
}
}

// INTERRUPTACIONES DE LOS PULSADORES:
// Pulsador inicio de mediciones:
void PULSADOR_INICIO (){

  delay(80);
  parada = 0;

}

// Pulsador fin de mediciones:
void PULSADOR_PARADA (){
  delay(80);
  parada = 1;
  archivoJSON = SD.open("ParteJ.txt", FILE_WRITE);

  if (archivoJSON && (memoria_corchete == 0)) {
    archivoJSON.println();
    archivoJSON.println("]");
    archivoJSON.close();
    memoria_corchete = 1;
    delay(2000);
  }
}

// IMPLEMENTACIÓN DE LAS FUNCIONES DE LOS SENSORES:

// DHT22:
// Temperatura:
float funcion_DHT22_temperatura (){

  float temperatura = 0;

  for(int i = 0; i < 4; i++){
    float temp = dht.readTemperature();
    temperatura = temperatura + temp;
    delay(2100);
  }
  temperatura = temperatura / 4;
  return (temperatura);
}

// Humedad:
float funcion_DHT22_humedad (){

  float humedad = 0;

  for(int i = 0; i < 4; i++){
    float hum = dht.readHumidity();
    humedad = humedad + hum;
    delay(2100);
  }
  humedad = humedad / 4;
  return (humedad);
}

// MQ135:
String funcion_MQ135 (){

  float valor_ppm = 0;
  String calidad;

  for(int i = 0; i < 4; i++){
    float ppm = sensor_MQ135.getPPM();
    valor_ppm = valor_ppm + ppm;
    delay(100);
  }
  valor_ppm = valor_ppm / 4;

  if(valor_ppm > 0 && valor_ppm <= 250){
    calidad = "Buena calidad";
  }
}

```

```

if(valor_ppm > 250 && valor_ppm <= 450){
    calidad = "Calidad normal";
}
if(valor_ppm > 450 && valor_ppm <= 800){
    calidad = "Mala calidad";
}
if(valor_ppm >= 800){
    calidad = "Muy mala calidad";
}
return (calidad);
}

// Ultrasonidos HC-SR04:

float funcion_HCSR04 (){

    float tiempoHC;
    float distanciaHC;

    const float base_recipiente = 0.04; // La base del recipiente tiene 0,04 m2 (20cm x 20cm).
    const float alt_recipiente = 0.20; // La altura del recipiente es de 0,2 m (20cm).
    const float litros = 1000.0;

    float volumen = 0.0;
    float altura_agua = 0.0;
    float distancia_media = 0.0;
    float precipitaciones = 0.0;

    for(int i = 0; i < 8; i++){
        digitalWrite(Triple, HIGH);
        delayMicroseconds(10); // Enviamos un pulso de 10us.
        digitalWrite(Triple, LOW);

        tiempoHC = pulseIn(Echo, HIGH); // Obtenemos el ancho del pulso.
        distanciaHC = (tiempoHC * 0.0172); // Escalamos el tiempo a una distancia en cm.

        float distanciaHC_metros = (distanciaHC - 4.75)/100.0; // Le restamos la distancia de más que está elevado el sensor.
        distancia_media = distancia_media + distanciaHC_metros;
        delay(250);
    }

    distancia_media = distancia_media / 8;
    altura_agua = alt_recipiente - distancia_media;

    volumen = base_recipiente * altura_agua * litros; // Multiplicamos por 1000 para pasar de m3 a litros.
    precipitaciones = volumen / base_recipiente;

    return (precipitaciones);
}

// ML8511:

float funcion_ML8511 (){

    int nivel_uv = 0;
    int nivel_ref = 0;
    float voltaje_salida = 0;
    float intensidad_uv = 0;
    float radiacion_uv = 0;

    for(int i = 0; i < 4; i++){
        nivel_uv = averageAnalogRead(UVOUT);
        nivel_ref = averageAnalogRead(REF_3V3);
        voltaje_salida = 3.3 / nivel_ref * nivel_uv;
        intensidad_uv = mapfloat(voltaje_salida, 1.0, 2.85, 0.0, 15.0);
        radiacion_uv = radiacion_uv + intensidad_uv;
        delay(100);
    }
    radiacion_uv = radiacion_uv / 4;
    return(radiacion_uv);
}

int averageAnalogRead(int pinToRead){

    byte numberOfReadings = 8;
    unsigned int runningValue = 0;

    for (int i = 0; i < numberOfReadings; i++){
        runningValue += analogRead(pinToRead);
        runningValue /= numberOfReadings;
    }
    return (runningValue);
}

float mapfloat(float x, float in_min, float in_max, float out_min, float out_max) {
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

// BMP280:
// Temperatura:

float funcion_BMP280_temperatura (){

    float temperatura = 0;

    for(int i = 0; i < 4; i++){

```

```

    float temp = sensorBMP.readTemperature();
    temperatura = temperatura + temp;
    delay(100);
}
temperatura = temperatura / 4;
return (temperatura);
}

// Presión:
float funcion_BMP280_presion (){

    float presion = 0;

    for(int i = 0; i < 4; i++){
        float pres = sensorBMP.readPressure()/100;
        presion = presion + pres;
        delay(100);
    }
    presion = presion / 4;
    return (presion);
}

// Altitud:
float funcion_BMP280_altitud (){

    float altitud = 0;

    for(int i = 0; i < 4; i++){
        float alt = sensorBMP.readAltitude();
        altitud = altitud + alt;
        delay(100);
    }
    altitud = altitud / 4;
    return (altitud);
}

// HD-38:
float funcion_HD38 (){

    float humedad_suelo = 0;

    for(int i = 0; i < 8; i++){
        float voltaje = analogRead(lectura_HD38) * (5.0 / 1023.0);
        humedad_suelo = humedad_suelo + voltaje;
        delay(100);
    }

    humedad_suelo = humedad_suelo / 8;
    float porcentaje_real =(1-(humedad_suelo/5))*100;
    int porcentaje_mostrado = 0;

    if(porcentaje_real <= 5){
        porcentaje_mostrado = 0;
        return (porcentaje_mostrado);
    } else if (porcentaje_real >= 100){
        porcentaje_mostrado = 100;
        return (porcentaje_mostrado);
    } else {
        return(porcentaje_real);
    }
}

// FC-03:
float funcion_FC03 (){

    float num_vueltas = 0.0;
    long tiempo_anterior = 0;
    float w_angular = 0.0;
    float v_viento = 0.0;
    float v_viento_kmh = 0.0;
    float r_encoder = 0.015;
    float velocidad_viento = 0.0;

    for(int i = 0; i < 30; i++){

        attachInterrupt(digitalPinToInterrupt(19),ANEMOMETRO,FALLING); // Interrupción 4 va en el pin 19.
        delay(1000);
        detachInterrupt(digitalPinToInterrupt(19));

        num_vueltas = 5 * 1000 / (millis() - tiempo_anterior) * num_int; // 1 vuelta del encoder son 12 interrupciones, que son sus 12 ranuras.
        tiempo_anterior = millis();
        num_int = 0;
        w_angular = num_vueltas * (2*3.1415927);
        v_viento = w_angular * r_encoder;
        v_viento_kmh = v_viento * 3.6;
        velocidad_viento = velocidad_viento + v_viento_kmh;
    }

    velocidad_viento = velocidad_viento / 30;
    return(velocidad_viento);
}

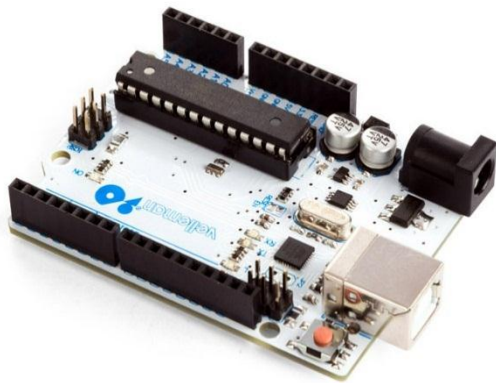
void ANEMOMETRO(){
    num_int++;
}

```

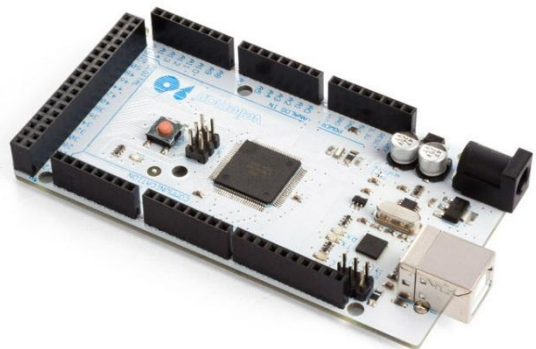
Anexo III: Fichas técnicas de los componentes

VMA100/VMA101/VMA102/VMA103

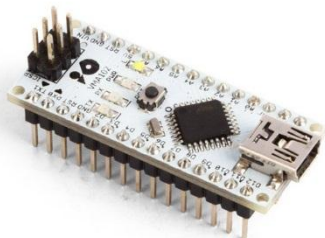
DEVELOPMENT BOARD



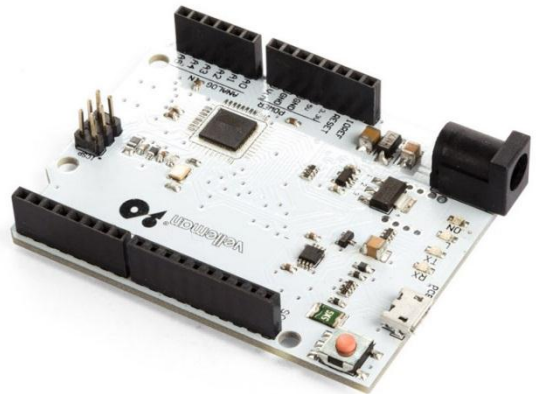
VMA100



VMA101



VMA102



VMA103



USER MANUAL

1. Introduction

To all residents of the European Union

Important environmental information about this product



This symbol on the device or the package indicates that disposal of the device after its lifecycle could harm the environment. Do not dispose of the unit (or batteries) as unsorted municipal waste; it should be taken to a specialized company for recycling. This device should be returned to your distributor or to a local recycling service. Respect the local environmental rules.

■ If in doubt, contact your local waste disposal authorities.

Thank you for choosing Velleman®! Please read the manual thoroughly before bringing this device into service. If the device was damaged in transit, do not install or use it and contact your dealer.

2. Safety Instructions



- This device can be used by children aged from 8 years and above, and persons with reduced physical, sensory or mental capabilities or lack of experience and knowledge if they have been given supervision or instruction concerning the use of the device in a safe way and understand the hazards involved. Children shall not play with the device. Cleaning and user maintenance shall not be made by children without supervision.



- Indoor use only.
Keep away from rain, moisture, splashing and dripping liquids.

3. General Guidelines



- Refer to the Velleman® Service and Quality Warranty on the last pages of this manual.
- Familiarise yourself with the functions of the device before actually using it.
- All modifications of the device are forbidden for safety reasons. Damage caused by user modifications to the device is not covered by the warranty.
- Only use the device for its intended purpose. Using the device in an unauthorised way will void the warranty.
- Damage caused by disregard of certain guidelines in this manual is not covered by the warranty and the dealer will not accept responsibility for any ensuing defects or problems.
- Nor Velleman nv nor its dealers can be held responsible for any damage (extraordinary, incidental or indirect) – of any nature (financial, physical...) arising from the possession, use or failure of this product.
- Due to constant product improvements, the actual product appearance might differ from the shown images.
- Product images are for illustrative purposes only.
- Do not switch the device on immediately after it has been exposed to changes in temperature. Protect the device against damage by leaving it switched off until it has reached room temperature.
- Keep this manual for future reference.

4. What is Arduino®

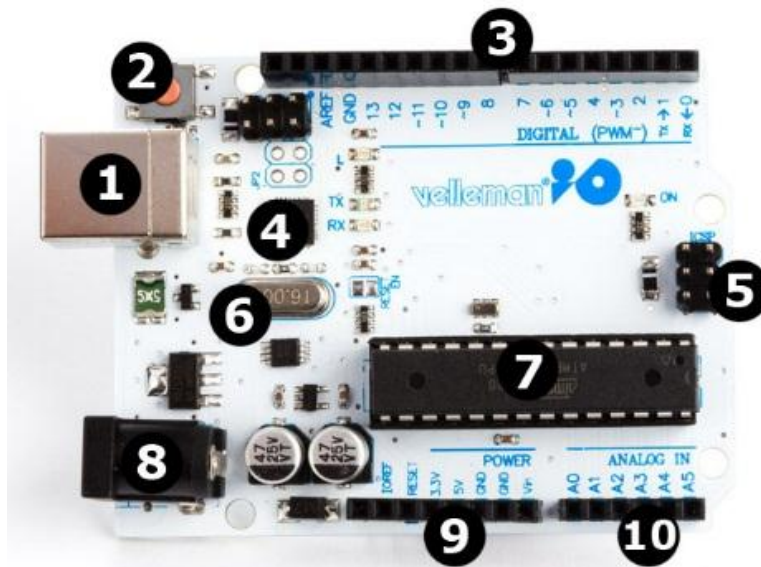
Arduino® is an open-source prototyping platform based in easy-to-use hardware and software. Arduino® boards are able to read inputs – light-on sensor, a finger on a button or a Twitter message – and turn it into an output – activating of a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so, you use the Arduino programming language (based on Wiring) and the Arduino® software IDE (based on Processing).

Surf to www.arduino.cc and www.arduino.org for more information.

5. Overview

VMA100

The VMA100 (Arduino® Uno compatible) is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. Connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

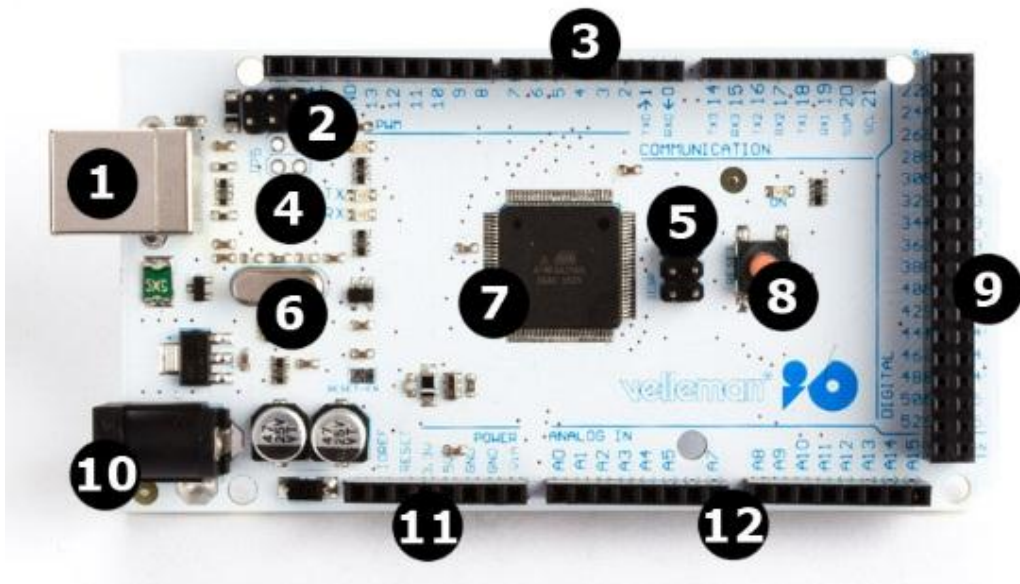


1	USB interface	6	16 MHz clock
2	reset button	7	Atmel mega328p (DIL)
3	digital I/O	8	7-12 VDC power input
4	Atmel mega16U2	9	power and ground pins
5	ICSP	10	analogue input pins

microcontroller	ATmega328
operating voltage.....	5 VDC
input voltage (recommended)	7-12 VDC
input voltage (limits).....	6-20 VDC
digital I/O pins	14 (of which 6 provide PWM output)
analogue input pins.....	6
DC current per I/O pin.....	40 mA
DC current for 3.3 V pin.....	50 mA
flash memory	32 kB (ATmega328) of which 0.5 kB used by bootloader
SRAM	2 kB (ATmega328)
EEPROM.....	1 kB (ATmega328)
clock speed	16 MHz
dimensions	
length	68.6 mm
width	53.4 mm
weight	25 g

VMA101

The VMA101 (Arduino® compatible) Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analogue inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller. Connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino® Duemilanove or Diecimila.



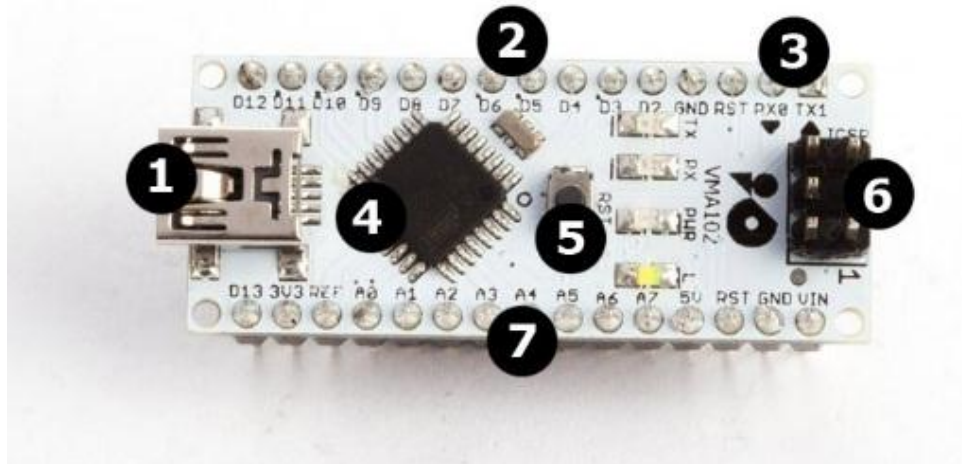
1	USB interface
2	ICSP for 16U2
3	digital I/O
4	Atmel mega16U2
5	ICSP for mega2560
6	16 MHz clock

7	Atmel mega2560
8	reset button
9	digital I/O
10	7-12 VDC power input
11	power and ground pins
12	analogue input pins

microcontroller	ATmega2560
operating voltage.....	5 VDC
input voltage (recommended)	7-12 VDC
input voltage (limits).....	6-20 VDC
digital I/O pins	54 (of which 15 provide PWM output)
analogue input pins.....	16
DC current per I/O pin.....	40 mA
DC current for 3.3 V pin.....	50 mA
flash memory	256 kB of which 8 kB used by bootloader
SRAM	8 kB
EEPROM.....	4 kB
clock speed	16 MHz
dimensions	
length	112 mm
width	55 mm
weight	62 g

VMA102

The VMA102 (Arduino® compatible Nano 3.0) is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino® Duemilanove, but in a different package. It lacks only a DC power jack, and works with a mini-B USB cable instead of a standard one.



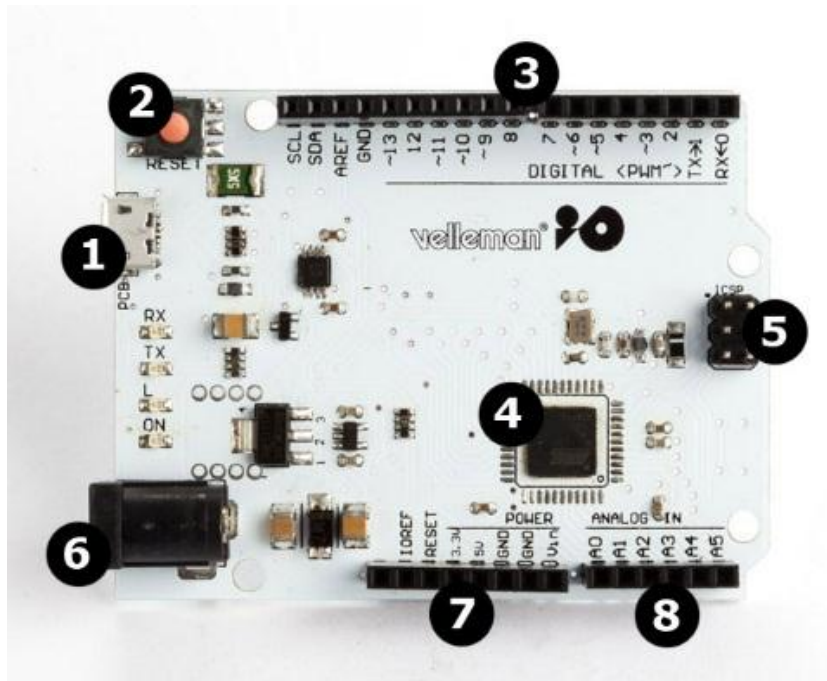
1	mini USB interface
2	digital I/O
3	serial communication pins
4	Atmel mega328p

5	reset button
6	ICSP
7	analogue input pins

microcontroller	Atmel ATmega168 or ATmega328
operating voltage.....	5 VDC
input voltage (recommended)	7-12 VDC
input voltage (limits).....	6-20 VDC
digital I/O pins	14 (of which 6 provide PWM output)
analogue input pins.....	8
DC current per I/O pin.....	40 mA
flash memory	16 kB (ATmega168) or 32 kB (ATmega328)
SRAM	1 kB (ATmega168) or 2 kB (ATmega328)
EEPROM.....	512 bytes (ATmega168) or 1 kB (ATmega328)
clock speed	16 MHz
dimensions	
length	45 mm
width	18 mm
weight	5 g

VMA103

The VMA103 (Arduino® compatible Leonardo) is a microcontroller board based on the ATmega32u4. It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analogue inputs), a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller. Connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.



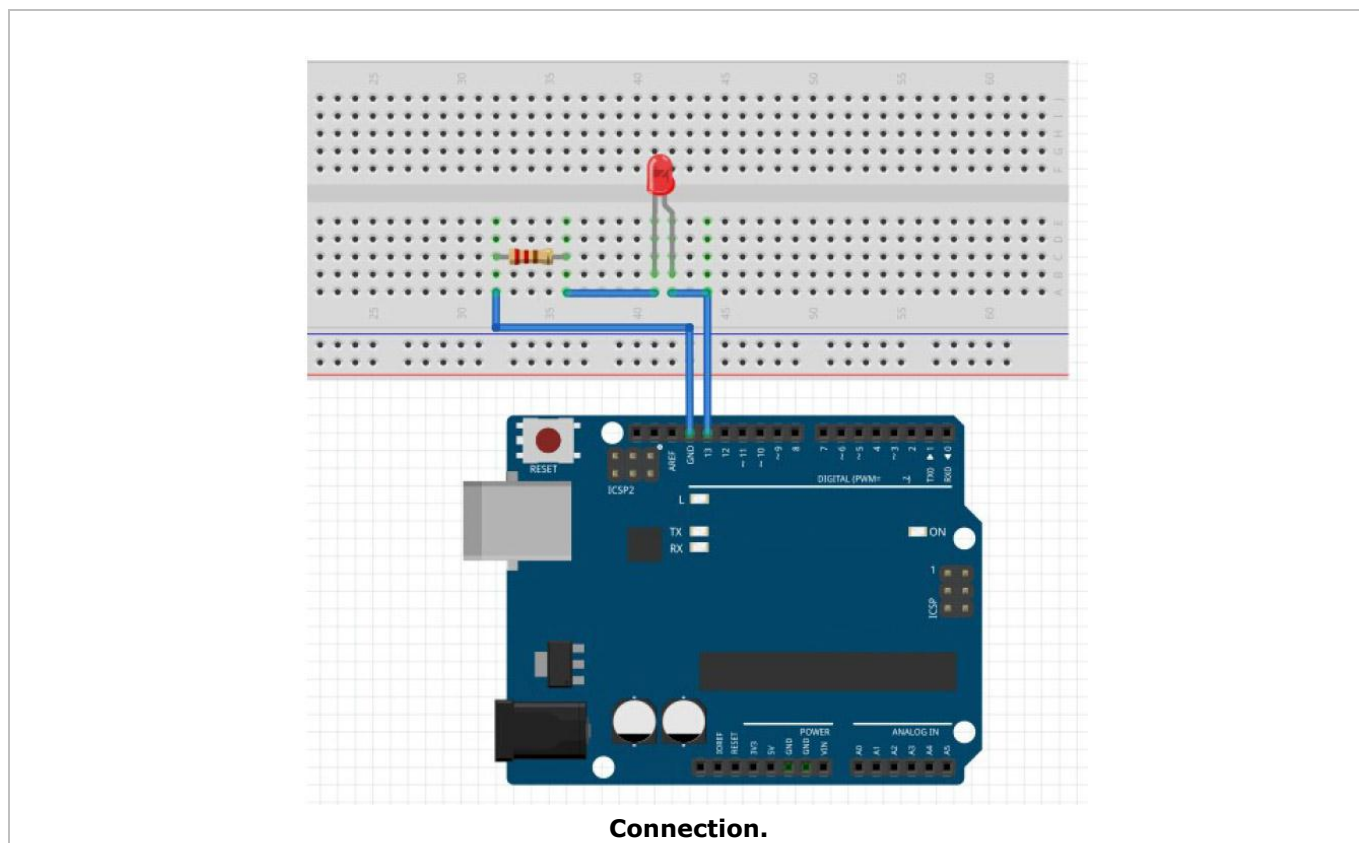
1	USB interface
2	reset button
3	digital I/O
4	Atmel mega32u4

5	ICSP
6	7-12 VDC power input
7	power and ground pins
8	analogue input pins

microcontroller	ATmega32u4
operating voltage.....	5 VDC
input voltage (recommended)	7-12 VDC
input voltage (limits).....	6-20 VDC
digital I/O pins	20
analogue input pins.....	12
DC current per I/O pin.....	40 mA
DC current for 3.3 V pin.....	50 mA
flash memory	32 kB (ATmega32u4) of which 4 kB used by bootloader
SRAM	2.5 kB (ATmega32u4)
EEPROM.....	1 kB (ATmega32u4)
clock speed	16 MHz
dimensions	
length	68.6 mm
width	53.3 mm
weight	20 g

6. How to Use

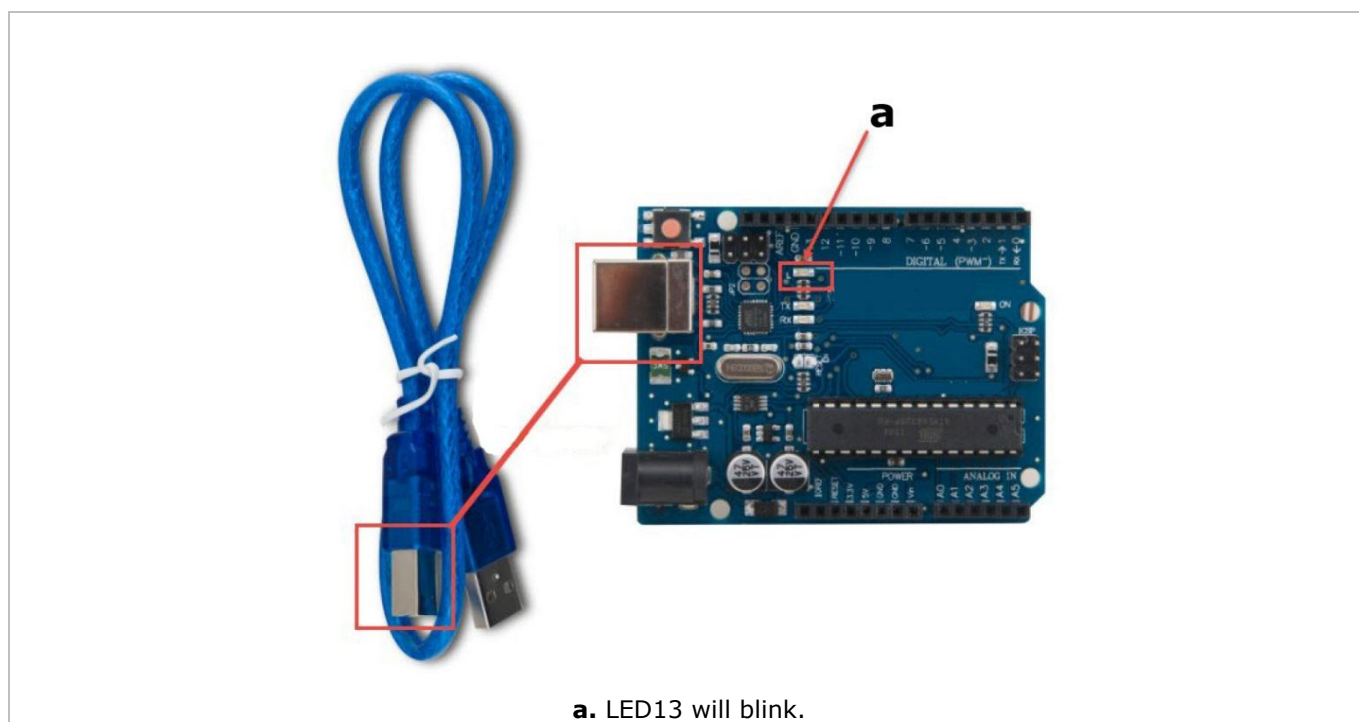
In the example below, we use the VMA100 to make an LED blink.



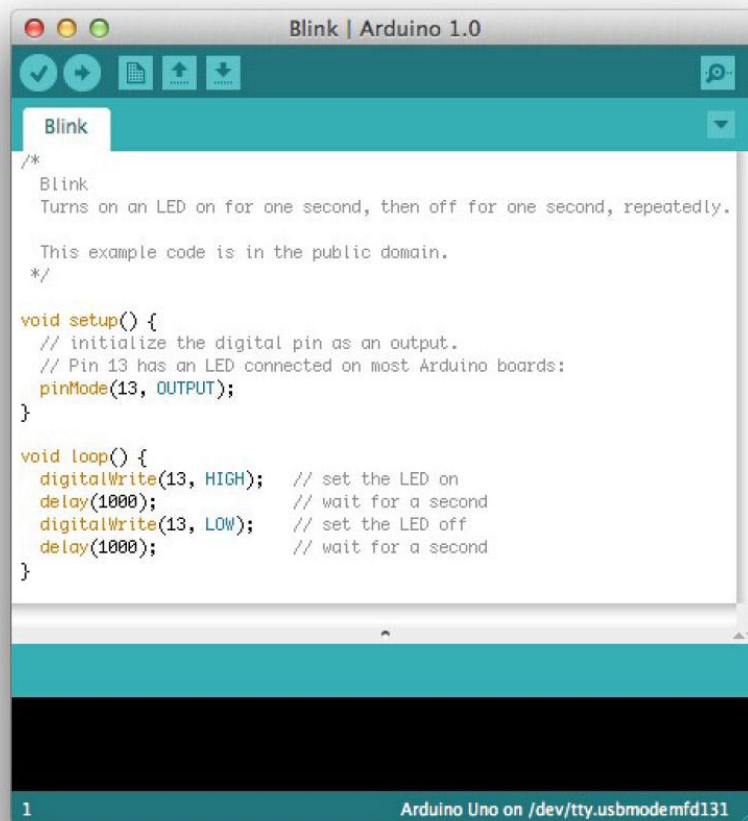
First, download the latest Arduino® Integrated Development Environment (IDE) from the Arduino® page.

<https://www.arduino.cc/en/Main/Software>

Install the software and use a USB cable to connect your VMA board to your computer.



Next, launch the Arduino® application and click File > Examples > 01. Basics > Blink.



```

Blink | Arduino 1.0
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

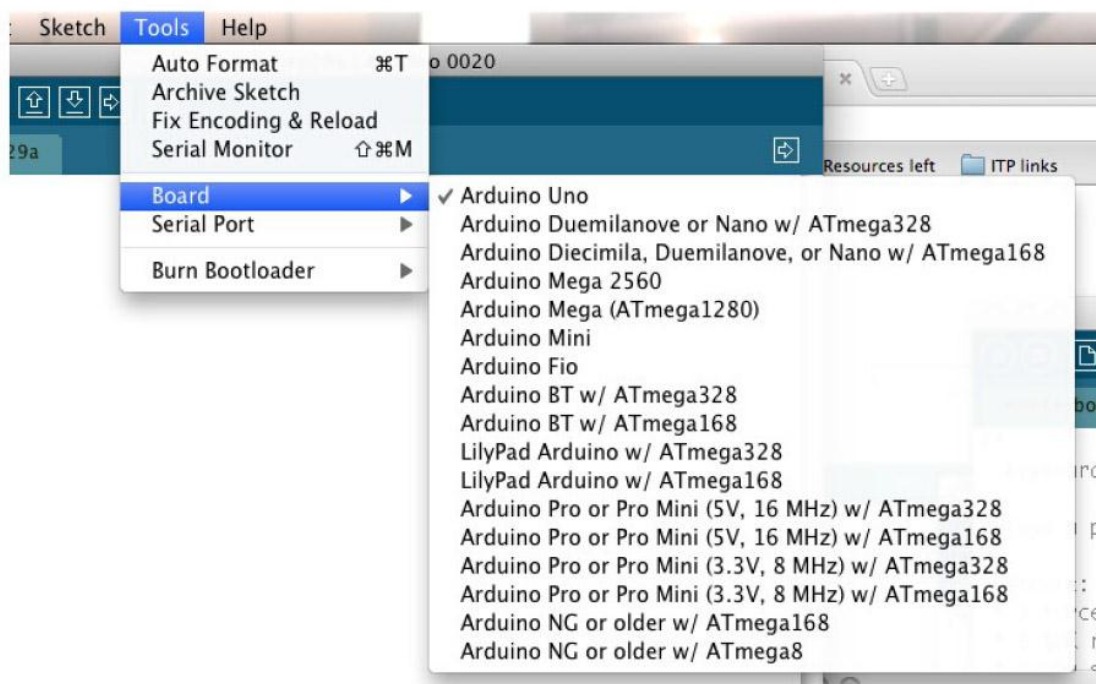
This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

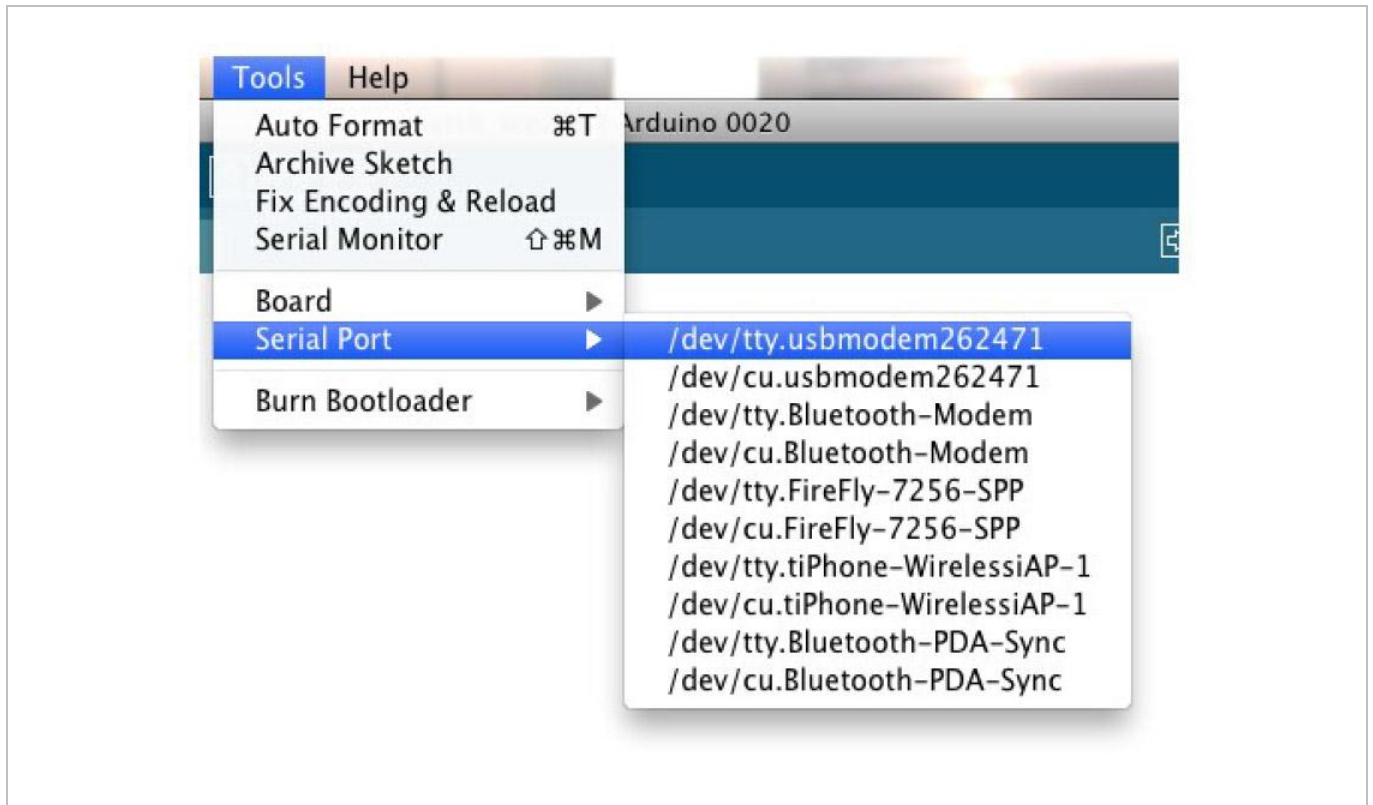
void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);          // wait for a second
}
  
```

1 Arduino Uno on /dev/tty.usbmodemfd131

You will need to select the used board. To do so, click Tools > Board.



Select the serial device of the Arduino board. To do so, click Tools > Serial Port.



The LED should now be blinking.

Use this device with original accessories only. Velleman nv cannot be held responsible in the event of damage or injury resulting from (incorrect) use of this device. For more info concerning this product and the latest version of this manual, please visit our website www.velleman.eu. The information in this manual is subject to change without prior notice.

© COPYRIGHT NOTICE

The copyright to this manual is owned by Velleman nv. All worldwide rights reserved. No part of this manual may be copied, reproduced, translated or reduced to any electronic medium or otherwise without the prior written consent of the copyright holder.

Velleman® Service and Quality Warranty

Since its foundation in 1972, Velleman® acquired extensive experience in the electronics world and currently distributes its products in over 85 countries.

All our products fulfil strict quality requirements and legal stipulations in the EU. In order to ensure the quality, our products regularly go through an extra quality check, both by an internal quality department and by specialized external organisations. If, all precautionary measures notwithstanding, problems should occur, please make appeal to our warranty (see guarantee conditions).

General Warranty Conditions Concerning Consumer Products (for EU):

- All consumer products are subject to a 24-month warranty on production flaws and defective material as from the original date of purchase.
- Velleman® can decide to replace an article with an equivalent article, or to refund the retail value totally or partially when the complaint is valid and a free repair or replacement of the article is impossible, or if the expenses are out of proportion.

You will be delivered a replacing article or a refund at the value of 100% of the purchase price in case of a flaw occurred in the first year after the date of purchase and delivery, or a replacing article at 50% of the purchase price or a refund at the value of 50% of the retail value in case of a flaw occurred in the second year after the date of purchase and delivery.

• Not covered by warranty:

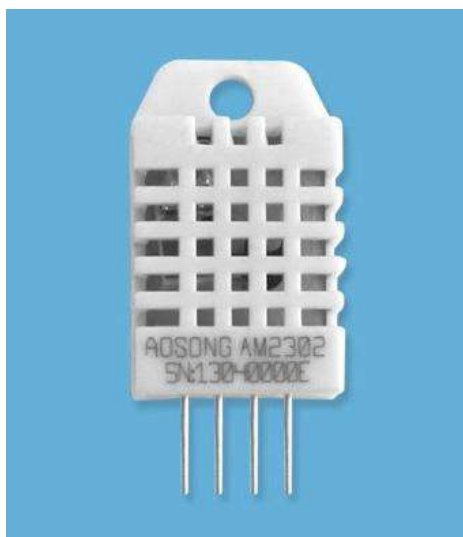
- all direct or indirect damage caused after delivery to the article (e.g. by oxidation, shocks, falls, dust, dirt, humidity...), and by the article, as well as its contents (e.g. data loss), compensation for loss of profits;
- consumable goods, parts or accessories that are subject to an aging process during normal use, such as batteries (rechargeable, non-rechargeable, built-in or replaceable), lamps, rubber parts, drive belts... (unlimited list);
- flaws resulting from fire, water damage, lightning, accident, natural disaster, etc....;
- flaws caused deliberately, negligently or resulting from improper handling, negligent maintenance, abusive use or use contrary to the manufacturer's instructions;
- damage caused by a commercial, professional or collective use of the article (the warranty validity will be reduced to six (6) months when the article is used professionally);
- damage resulting from an inappropriate packing and shipping of the article;
- all damage caused by modification, repair or alteration performed by a third party without written permission by Velleman®.
- Articles to be repaired must be delivered to your Velleman® dealer, solidly packed (preferably in the original packaging), and be completed with the original receipt of purchase and a clear flaw description.
- Hint: In order to save on cost and time, please reread the manual and check if the flaw is caused by obvious causes prior to presenting the article for repair. Note that returning a non-defective article can also involve handling costs.
- Repairs occurring after warranty expiration are subject to shipping costs.
- The above conditions are without prejudice to all commercial warranties.

The above enumeration is subject to modification according to the article (see article's manual).

AOSONG

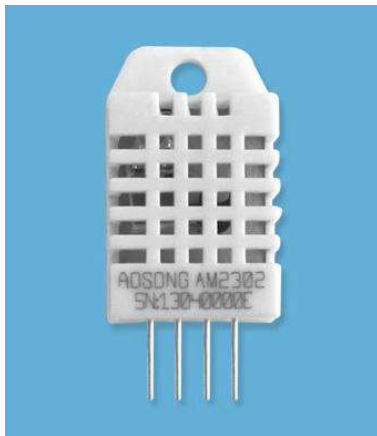
Temperature and humidity module

AM2302 Product Manual

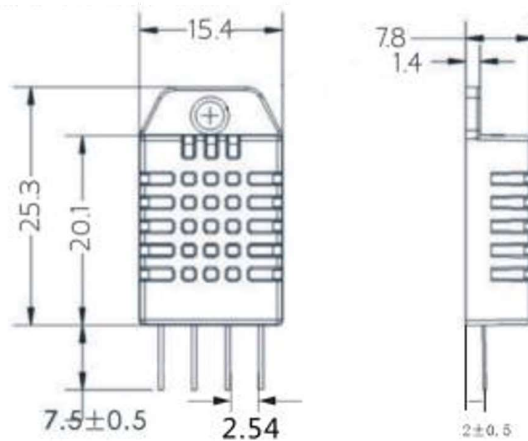


1、 Product Overview

AM2302 capacitive humidity sensing digital temperature and humidity module is one that contains the compound has been calibrated digital signal output of the temperature and humidity sensors. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a capacitive sensor wet components and a high-precision temperature measurement devices, and connected with a high-performance 8-bit microcontroller. The product has excellent quality, fast response, strong anti-jamming capability, and high cost. Each sensor is extremely accurate humidity calibration chamber calibration. The form of procedures, the calibration coefficients stored in the microcontroller, the sensor within the processing of the heartbeat to call these calibration coefficients. Standard single-bus interface, system integration quick and easy. Small size, low power consumption, signal transmission distance up to 20 meters, making it the best choice of all kinds of applications and even the most demanding applications. Products for the 3-lead (single-bus interface) connection convenience. Special packages according to user needs.



Physical map



Dimensions (unit: mm)

2、 Applications

HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, home appliances, humidity regulator, medical, weather stations, and other humidity measurement and control and so on.

3、 Features

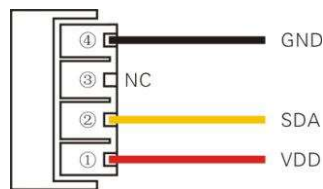
Ultra-low power, the transmission distance, fully automated calibration, the use of capacitive humidity sensor, completely interchangeable, standard digital single-bus output, excellent long-term stability, high accuracy temperature measurement devices.

4、The definition of single-bus interface

4.1 AM2302 Pin assignments

Table 1: AM2302 Pin assignments

Pin	Name	Description
①	VDD	Power (3.3V–5.5V)
②	SDA	Serial data, bidirectional port
③	NC	Empty
④	GND	Ground



PIC1: AM2302 Pin Assignment

4.2 Power supply pins (VDD GND)

AM2302 supply voltage range 3.3V – 5.5V, recommended supply voltage is 5V.

4.3 Serial data (SDA)

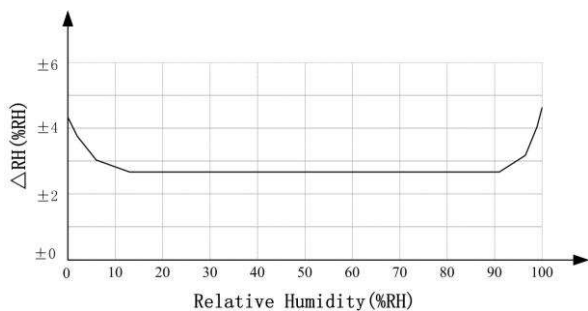
SDA pin is tri structure for reading, writing sensor data. Specific communication timing, see the detailed description of the communication protocol.

5、Sensor performance

5.1 Relative humidity

Table 2: AM2302 Relative humidity performance table

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		%RH
Range		0		99.9	%RH
Accuracy ^[1]	25°C		± 2		%RH
Repeatability			± 0.3		%RH
Exchange		Completely interchangeable			
Response ^[2]	1/e(63%)		<5		S
Sluggish			<0.3		%RH
Drift ^[3]	Typical		<0.5		%RH/yr

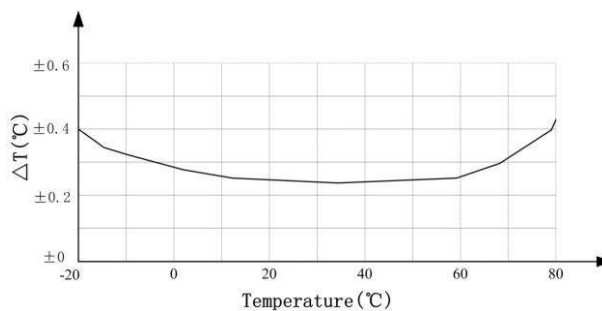


Pic2: At25°C The error of relative humidity

5.2 Temperature

Table 3: AM2302 Relative temperature performance

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		°C
n			16		bit
Accuracy			± 0.5	± 1	°C
Range		-40		80	°C
Repeat			± 0.2		°C
Exchange		Completely interchangeable			
Response	1/e(63%)		<10		S
Drift			± 0.3		°C/yr



Pic3: The maximum temperature error

6、Electrical Characteristics

Electrical characteristics, such as energy consumption, high, low, input, output voltage, depending on the power supply. Table 4 details the electrical characteristics of the AM2302, if not identified, said supply voltage of 5V. To get the best results with the sensor, please design strictly in accordance with the conditions of design in Table 4.

Table 4: AM2302 DC Characteristics

Parameter	Condition	min	typ	max	Unit
Voltage		3.3	5	5.5	V
Power consumption ^[4]	Dormancy	10	15		μA
	Measuring		500		μA
	Average		300		μA
Low level output voltage	I _{OL} ^[5]	0		300	mV
High output voltage	R _p <25 kΩ	90%		100%	VDD
Low input voltage	Decline	0		30%	VDD
Input High Voltage	Rise	70%		100%	VDD
R _{pu} ^[6]	VDD = 5V VIN = VSS	30	45	60	kΩ
Output current	turn on		8		mA
	turn off	10	20		μA
Sampling period		2			S

[1] the accuracy of the factory inspection, the sensor 25°C and 5V, the accuracy specification of test conditions, it does not include hysteresis and nonlinearity, and is only suitable for non-condensing environment.

[2] to achieve an order of 63% of the time required under the conditions of 25°C and 1m / s airflow.

[3] in the volatile organic compounds, the values may be higher. See the manual application to store information.

[4] this value at VDD = 5.0V when the temperature is 25°C, 2S / time, under the conditions of the average.

[5] low output current.

[6] that the pull-up resistor.

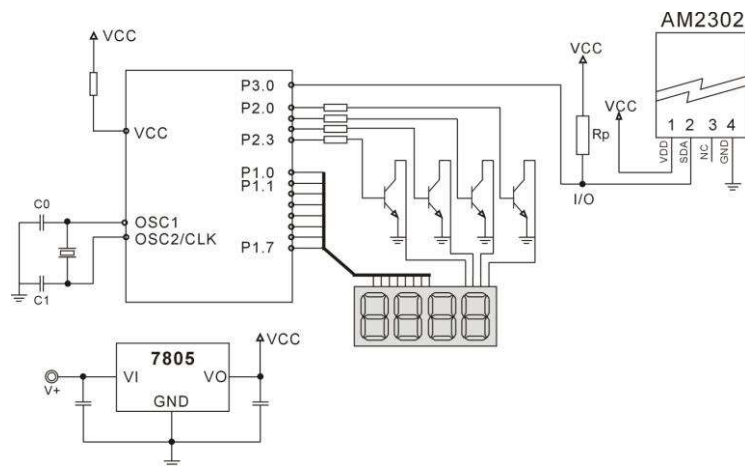
7、Single-bus communication (ONE-WIRE)

7.1 Typical circuits for single bus

Microprocessor and AM2302 connection typical application circuit is shown in Figure 4. Single bus communication mode, pull the SDA microprocessor I / O port is connected.

Special instructions of the single-bus communication :

1. Typical application circuit recommended in the short cable length of 30 meters on the 5.1K pull-up resistor pullup resistor according to the actual situation of lower than 30 m.
2. With 3.3V supply voltage, cable length shall not be greater than 100cm. Otherwise, the line voltage drop will lead to the sensor power supply, resulting in measurement error.
3. Read the sensor minimum time interval for the 2S; read interval is less than 2S, may cause the temperature and humidity are not allowed or communication is unsuccessful, etc..
4. Temperature and humidity values are each read out the results of the last measurement For real-time data that need continuous read twice, we recommend repeatedly to read sensors, and each read sensor interval is greater than 2 seconds to obtain accuratethe data.



Pic4: AM2302 Typical circuits for single bus

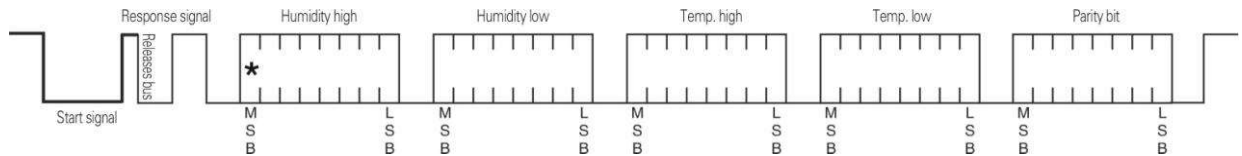
7.2、Single-bus communication protocol

◎Single bus Description

AM2302 device uses a simplified single-bus communication. Single bus that only one data line, data exchange system, controlled by the data line to complete. Equipment (microprocessor) through an open-drain or tri-state port connected to the data line to allow the device does not send data to release the bus, while other devices use the bus; single bus usually require an external about 5.1kΩ pull-up resistor, so when the bus is idle, its status is high. Because they are the master-slave structure, only the host calls the sensor, the sensor will answer, so the hosts to access the sensor must strictly follow the sequence of single bus, if there is a sequence of confusion, the sensor will not respond to the host.

◎Single bus to send data definition

SDA For communication and synchronization between the microprocessor and the AM2302, single-bus data format, a transmission of 40 data, the high first-out. Specific communication timing shown in Figure 5, the communication format is depicted in Table 5.



Pic5: AM2302 Single-bus communication protocol

Table 5: AM2302 Communication format specifier

Name	Single-bus format definition
Start signal	Microprocessor data bus (SDA) to bring down a period of time (at least 800μ s) [1] notify the sensor to prepare the data.
Response signal	Sensor data bus (SDA) is pulled down to 80μ s, followed by high-80μ s response to host the start signal.
Data format	Host the start signal is received, the sensor one-time string from the data bus (SDA) 40 data, the high first-out.
Humidity	Humidity resolution of 16Bit, the previous high; humidity sensor string value is 10 times the actual humidity values.
Temp.	Temperature resolution of 16Bit, the previous high; temperature sensor string value is 10 times the actual temperature value; The temperature is the highest bit (Bit15) is equal to 1 indicates a negative temperature, the temperature is the highest bit (Bit15) is equal to 0 indicates a positive temperature; Temperature in addition to the most significant bit (Bit14 ~ bit 0) temperature values.
Parity bit	Parity bit = humidity high + humidity low + temperature high + temperature low

◎ **Single-bus data calculation example**

Example 1: 40 Data received:

0000 0010 1001 0010 0000 0001 0000 1101 1010 0010
 High humidity 8 Low humidity 8 High temp. 8 Low temp. 8 Parity bit

Calculate:

$0000\ 0010 + 1001\ 0010 + 0000\ 0001 + 0000\ 1101 = 1010\ 0010$ (Parity bit)

Received data is correct:

humidity: $0000\ 0010\ 1001\ 0010 = 0292\text{H}$ (Hexadecimal) = $2 \times 256 + 9 \times 16 + 2 = 658$
 \Rightarrow Humidity = 65.8%RH

Temp.: $0000\ 0001\ 0000\ 1101 = 10\text{DH}$ (Hexadecimal) = $1 \times 256 + 0 \times 16 + 13 = 269$
 \Rightarrow Temp. = 26.9°C

◎ **Special Instructions:**

When the temperature is below 0 °C, the highest position of the temperature data.

Example: -10.1 °C Expressed as 1 000 0000 0110 0101

Temp.: $0000\ 0000\ 0110\ 0101 = 0065\text{H}$ (Hexadecimal) = $6 \times 16 + 5 = 101$
 \Rightarrow Temp. = -10.1°C

Example 2: 40 received data:

0000 0010 1001 0010 0000 0001 0000 1101 1011 0010
 High humidity 8 Low humidity 8 High temp. 8 Low temp. 8 Parity bit

Calculate:

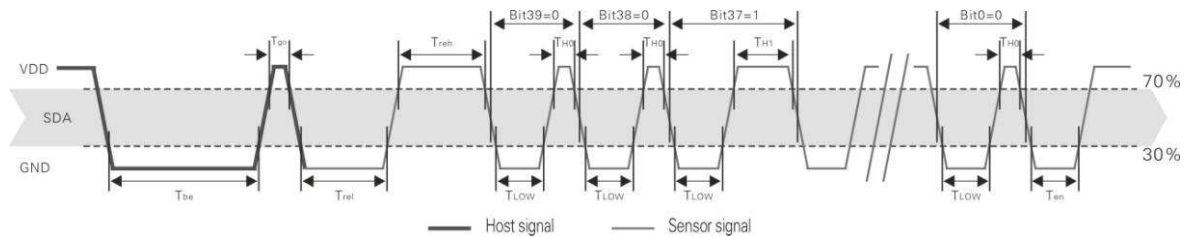
$0000\ 0010 + 1001\ 0010 + 0000\ 0001 + 0000\ 1101 = 1010\ 0010 \neq 1011\ 0010$ (Validation error)

The received data is not correct, give up, to re-receive data.

7.3 Single-bus communication timing

User host (MCU) to send a start signal (data bus SDA line low for at least $800\mu s$) after AM2302 from Sleep mode conversion to high-speed mode. The host began to signal the end of the AM2302 send a response signal sent from the data bus SDA serial 40Bit's data, sends the byte high; data sent is followed by: Humidity high , Humidity low , Temperature high , Temperature low , Parity bit , Send data to the end of trigger information collection, the collection end of the sensor is automatically transferred to the sleep mode, the advent until the next communication.

Detailed timing signal characteristics in Table 6 , Single-bus communication timing diagram Pic 6:



Pic 6: AM2302 Single-bus communication timing

Note: the temperature and humidity data read by the host from the AM2302 is always the last measured value, such as the two measurement interval is very long, continuous read twice to the second value of real-time temperature and humidity values, while two readtake minimum time interval be 2S.

Table 6: Single bus signal characteristics

Symbol	Parameter	min	typ	max	Unit
T_{be}	Host the start signal down time	0.8	1	20	mS
T_{go}	Bus master has released time	20	30	200	μS
T_{rel}	Response to low time	75	80	85	μS
T_{reh}	In response to high time	75	80	85	μS
T_{LOW}	Signal "0", "1" low time	48	50	55	μS
T_{H0}	Signal "0" high time	22	26	30	μS
T_{H1}	Signal "1" high time	68	70	75	μS
T_{en}	Sensor to release the bus time	45	50	55	μS

Note: To ensure the accurate communication of the sensor, the read signal, in strict accordance with the design parameters and timing in Table 6 and Figure 6.

7.4 Peripherals read step example

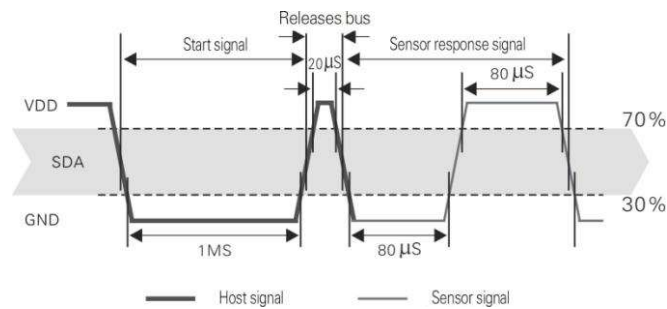
Communication between the host and the sensor can read data through the following three steps to complete.

Step 1

AM2302 have to wait for the power (on AM2302 power 2S crossed the unstable state, the device can not send any instructions to read during this period), the test environment temperature and humidity data, and record data, since the sensor into a sleep state automatically. AM2302 The SDA data line from the previous pull-up resistor pulled up is always high, the AM2302 the SDA pin is in input state, the time detection of external signal.

Step 2

Microprocessor I/O set to output, while output low, and low hold time can not be less than 800us, typical values are down 1MS, then the microprocessor I/O is set to input state, the release of the bus, due to the pull-up resistor, the microprocessor I/O AM2302 the SDA data line also will be high, the bus master has released the AM2302 send a response signal, that is, the output 80 microseconds low as the response signal, tightthen output high of 80 microseconds notice peripheral is ready to receive data signal transmission as shown to Pic7 :



Pic7: Single bus decomposition of the timing diagram

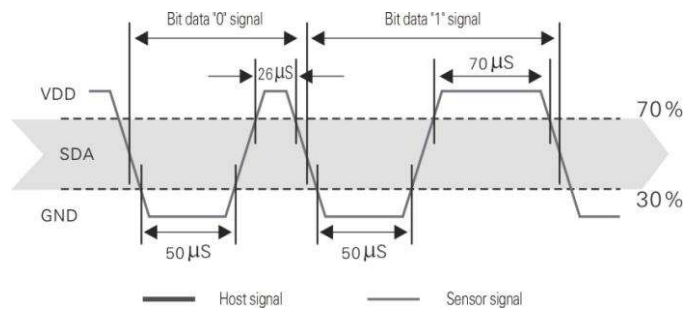
Step 3

AM2302 sending the response, followed by the data bus SDA continuous serial output 40 data, the microprocessor receives 40 data I/O level changes.

Bit data "0" format: 26–28 microseconds 50 microseconds low plus high;

Bit data "1" format: the high level of low plus, 50 microseconds to 70 microseconds;

Bit data "0" bit data "1" format signal shown to pic 8:

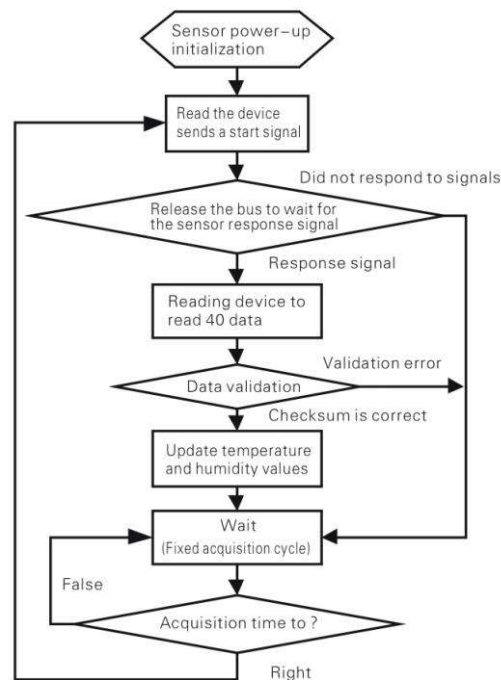


Pic 8: The single bus break down the timing diagram

AM2302 data bus SDA output 40 data continue to output the low 50 microseconds into the input state, followed by pull-up resistor goes high. AM2302 internal re-test environmental temperature and humidity data, and record the data, the end of the test records, the microcontroller automatically into hibernation. Microcontroller only after receipt of the start signal of the host wake-up sensor, into the working state.

7.5 Peripheral to read flow chart

AM2302 sensor read single bus flow chart diagram shown in Figure 9, we also provide the C51 read the code examples, customers need to download, please visit our website (www.aosong.com) related to downloadthis manual does not provide the code description.



Pic9: Single-bus to read the flow chart

8、 Application of information

1. Work and storage conditions

Outside the sensor the proposed scope of work may lead to temporary drift of the signal up to 300% RH. Return to normal working conditions, sensor calibration status will slowly toward recovery. To speed up the recovery process may refer to "resume processing". Prolonged use of non-normal operating conditions, will accelerate the aging of the product.

Avoid placing the components on the long-term condensation and dry environment, as well as the following environment.

A, salt spray

B, acidic or oxidizing gases such as sulfur dioxide, hydrochloric acid

Recommended storage environment

Temperature: 10 ~ 40 °C Humidity: 60% RH or less

2. The impact of exposure to chemicals

The capacitive humidity sensor has a layer by chemical vapor interference, the proliferation of chemicals in the sensing layer may lead to drift and decreased sensitivity of the measured values. In a pure environment, contaminants will slowly be released. Resume processing as described below will accelerate this process. The high concentration of chemical pollution (such as ethanol) will lead to the complete damage of the sensitive layer of the sensor.

3. The temperature influence

Relative humidity of the gas to a large extent dependent on temperature. Therefore, in the measurement of humidity,

should be to ensure that the work of the humidity sensor at the same temperature. With the release of heat of electronic components share a printed circuit board, the installation should be as far as possible the sensor away from the electronic components and mounted below the heat source, while maintaining good ventilation of the enclosure. To reduce the thermal conductivity sensor and printed circuit board copper plating should be the smallest possible, and leaving a gap between the two.

4. Light impact

Prolonged exposure to sunlight or strong ultraviolet radiation, and degrade performance.

5. Resume processing

Placed under extreme working conditions or chemical vapor sensor, which allows it to return to the status of calibration by the following handler. Maintain two hours in the humidity conditions of 45°C and <10% RH (dry); followed by 20–30°C and > 70% RH humidity conditions to maintain more than five hours.

6. Wiring precautions

The quality of the signal wire will affect the quality of the voltage output, it is recommended to use high quality shielded cable.

7. Welding information

Manual welding, in the maximum temperature of 300°C under the conditions of contact time shall be less than 3 seconds.

8. Product upgrades

Details, please the consultation Aosong electronics department.

9、 The license agreement

Without the prior written permission of the copyright holder, shall not in any form or by any means, electronic or mechanical (including photocopying), copy any part of this manual, nor shall its contents be communicated to a third party. The contents are subject to change without notice.

The Company and third parties have ownership of the software, the user may use only signed a contract or software license.

10、 Warnings and personal injury

This product is not applied to the safety or emergency stop devices, as well as the failure of the product may result in injury to any other application, unless a particular purpose or use authorized. Installation, handling, use or maintenance of the product refer to product data sheets and application notes. Failure to comply with this recommendation may result in death and serious personal injury. The Company will bear all damages resulting personal injury or death, and waive any claims that the resulting subsidiary company managers and employees and agents, distributors, etc. that may arise, including: a variety of costs, compensation costs, attorneys' fees, and so on.

11、Quality Assurance

The company and its direct purchaser of the product quality guarantee period of three months (from the date of delivery). Publishes the technical specifications of the product data sheet shall prevail. Within the warranty period, the product was confirmed that the quality is really defective, the company will provide free repair or replacement. The user must satisfy the following conditions:

- ① The product is found defective within 14 days written notice to the Company;
- ② The product shall be paid by mail back to the company;
- ③ The product should be within the warranty period.

The Company is only responsible for those used in the occasion of the technical condition of the product defective product. Without any guarantee, warranty or written statement of its products used in special applications. Company for its products applied to the reliability of the product or circuit does not make any commitment.

TECHNICAL DATA

MQ-135 GAS SENSOR

FEATURES

Wide detecting scope
Stable and long life

Fast response and High sensitivity
Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH₃,NO_x, alcohol, Benzene, smoke,CO₂,etc.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 800mw	

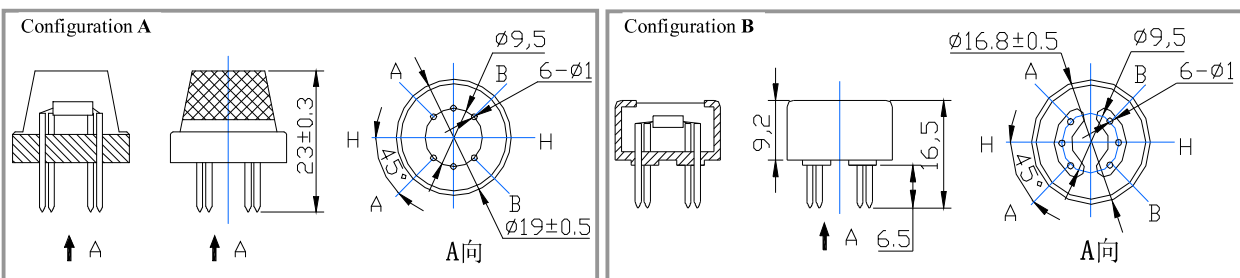
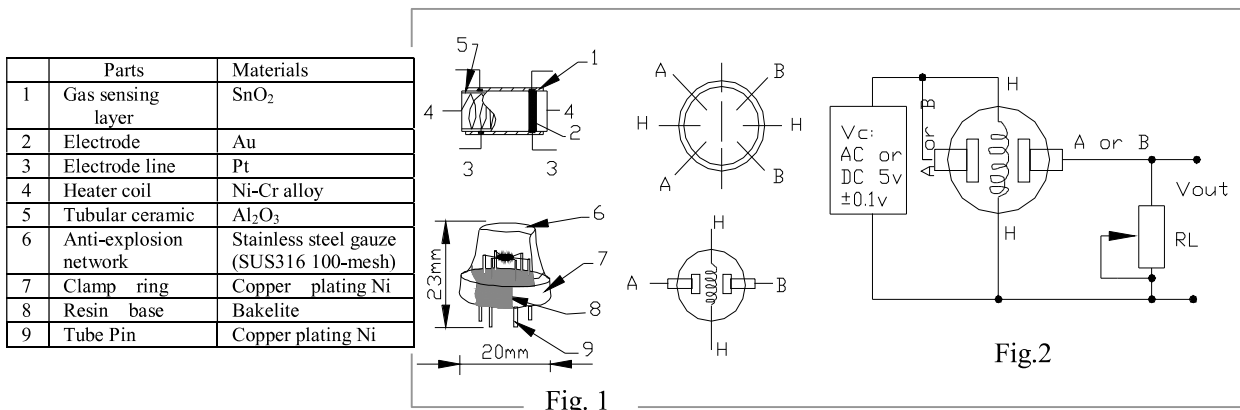
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-10°C-45°C	
Tas	Storage Tem	-20°C-70°C	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
R _s	Sensing Resistance	30K Ω -200K Ω (100ppm NH ₃)	Detecting concentration scope: 10ppm-300ppm NH ₃ 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH ₃	Concentration Slope rate	≤0.65	
Standard Detecting Condition	Temp: 20°C ±2°C Humidity: 65%±5%	V _c :5V±0.1 V _H : 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro AL₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of

sensitive components. The enveloped MQ-135 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-135

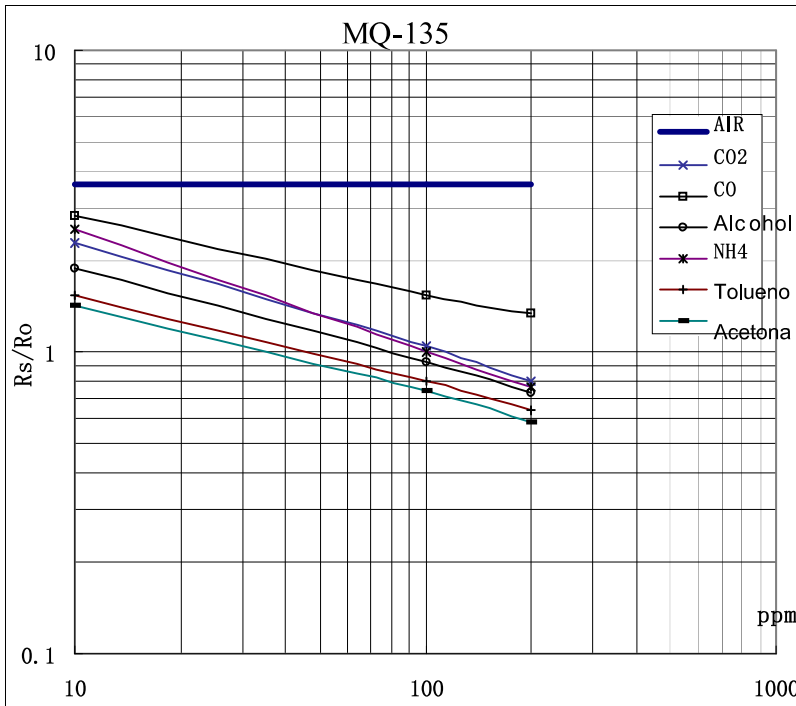


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases.

in their: Temp: 20°C、
Humidity: 65%、
O₂ concentration 21%
RL=20k Ω

Ro: sensor resistance at 100ppm of NH₃ in the clean air.

Rs:sensor resistance at various concentrations of gases.

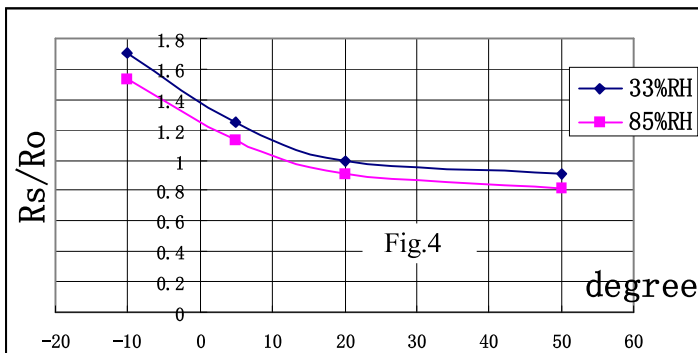


Fig.4 is shows the typical dependence of the MQ-135 on temperature and humidity.

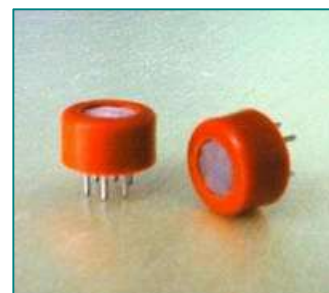
Ro: sensor resistance at 100ppm of NH₃ in air at 33%RH and 20 degree.

Rs: sensor resistance at 100ppm of NH₃ at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-135 is difference to various kinds and various concentration gases. So,When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 100ppm NH₃ or 50ppm Alcohol concentration in air and use value of Load resistancethat(R_L) about 20 K Ω (10K Ω to 47 K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.



Notification

1 Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixture, silicon latex, putty or plastic contain silicon environment

1.2 High Corrosive gas

If the sensors exposed to high concentration corrosive gas (such as H_2S , SO_x , Cl_2 , HCl etc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin.

1.4 Touch water

Sensitivity of the sensors will be reduced when splattered or dipped in water.

1.5 Freezing

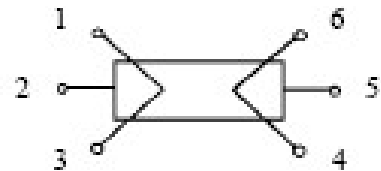
Do avoid icing on sensor's surface, otherwise sensor would lose sensitivity.

1.6 Applied voltage higher

Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, if apply voltage on 1、3 pins or 4、6 pins, it will make lead broken, and without signal when apply on 2、4 pins



2 Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor' sensitivity will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if long time placed in high gas concentration, it will affect sensors characteristic.

2.3 Long time storage

The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof without silicon gel bag with clean air. For the sensors with long time storage but no electrify, they need long aging time for stability before using.

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then reapture. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage

For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions:

2.7.1 Soldering flux: Rosin soldering flux contains least chlorine

2.7.2 Speed: 1-2 Meter/ Minute

2.7.3 Warm-up temperature: $100 \pm 20^\circ C$

2.7.4 Welding temperature: $250 \pm 10^\circ C$

2.7.5 1 time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will be reduced.



Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

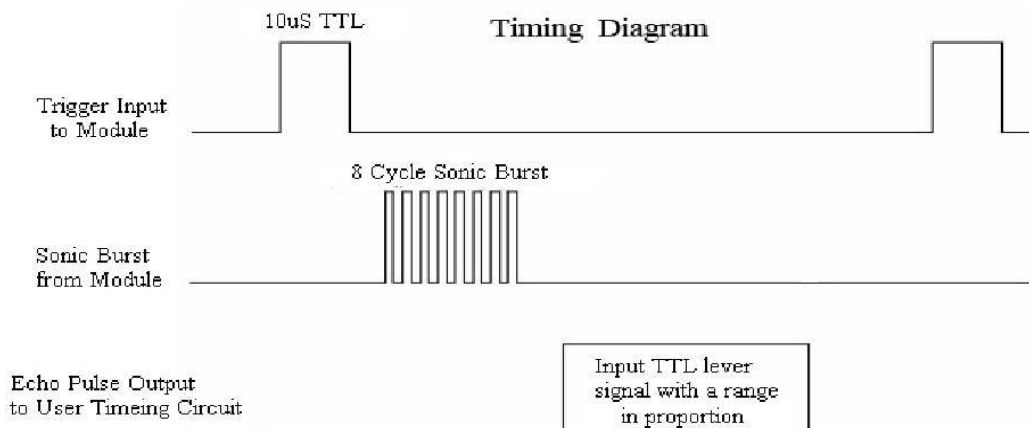
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

www.ElecFreaks.com



ML8511

UV Sensor with Voltage Output

GENERAL DISCRIPTION

The ML8511 is a UV sensor, which is suitable for acquiring UV intensity indoors or outdoors. The ML8511 is equipped with an internal amplifier, which converts photo-current to voltage depending on the UV intensity. This unique feature offers an easy interface to external circuits such as ADC. In the power down mode, typical standby current is 0.1 μ A, thus enabling a longer battery life.

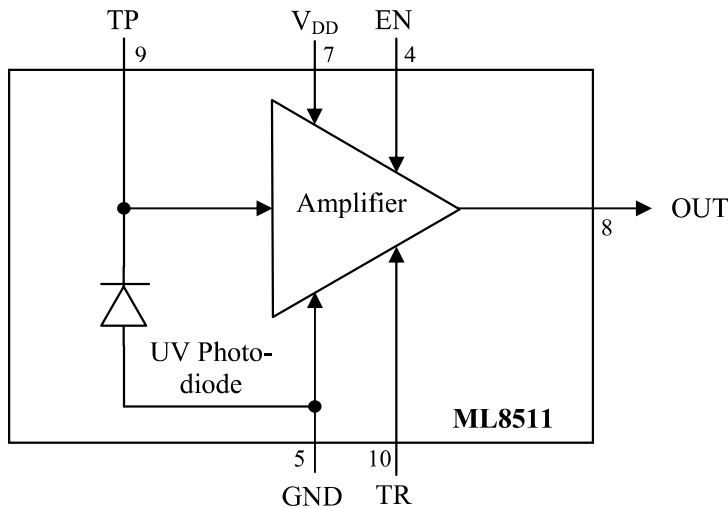
FEATURES

- Photodiode sensitive to UV-A and UV-B
- Embedded operational amplifier
- Analog voltage output
- Low supply current (300 μ A typ.) and low standby current (0.1 μ A typ.)
- Small and thin surface mount package (4.0mm x 3.7mm x 0.73mm, 12-pin ceramic QFN)

APPLICATIONS

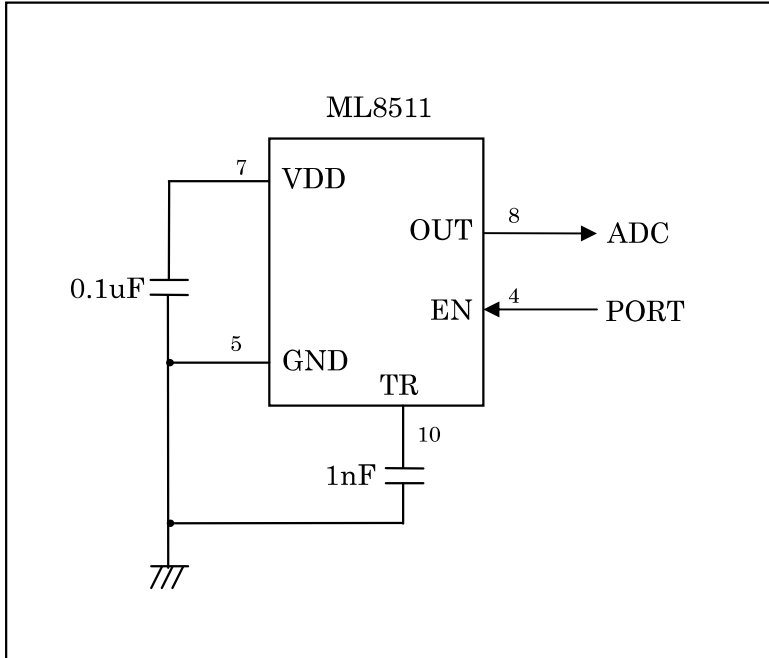
- Smart phone, Watch, Weather station, Bicycle navigation, Accessary, Gaming

BLOCK DAIAGRAM



PIN CONFIGURATIONS

Pin	Symbol	I/O	Function
7	VDD	PW	Supply voltage. Decouple this pin to ground with 0.1 μ F capacitor.
5	GND	PW	Ground
4	EN	I	Active high enable pin. (High: Active mode, Low: Standby mode)
8	OUT	O	Output (Low in power down or standby mode)
9	TP	I/O	Test pin. Do not connect.
10	TR	I/O	Internal reference voltage. Decouple this pin to ground with 1 nF capacitor.
1,2,3, 6,11,12	NC	-	No Connection. Do not connect.

EXAMPLE OF CONNCTING DIAGRAM

* Load resistance of OUT port is recommended more than $100\text{ k}\Omega$.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Condition	Rating	unit
Supply Voltage	V_{DD}	Ta=25 °C	-0.3 to +4.6	V
Input Voltage	V_I	Ta=25 °C	-0.3 to +4.6	V
Output Short Current	I_{OS}	Ta=25 °C	5	mA
Power Dissipation	P_D	Ta=25 °C	30	mW
Storage Temperature	T_{stg}	-	-30 to +85	°C

RECOMENDED OPERATION CONDITIONS

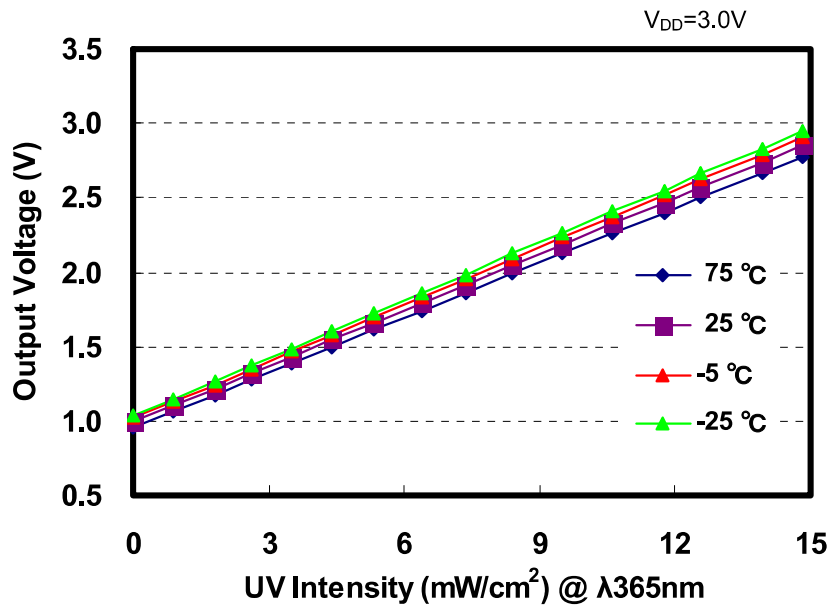
Parameter	Symbol	Min.	Typ.	Max.	unit
Operating Voltage	V_{DD}	2.7	3.3	3.6	V
Operating Temperature	Ta	-20	-	70	°C

ELECTRO-OPTICAL CHARACTERISTICS(V_{DD}=+2.7V to +3.6V, Ta= -20°C to +70°C)

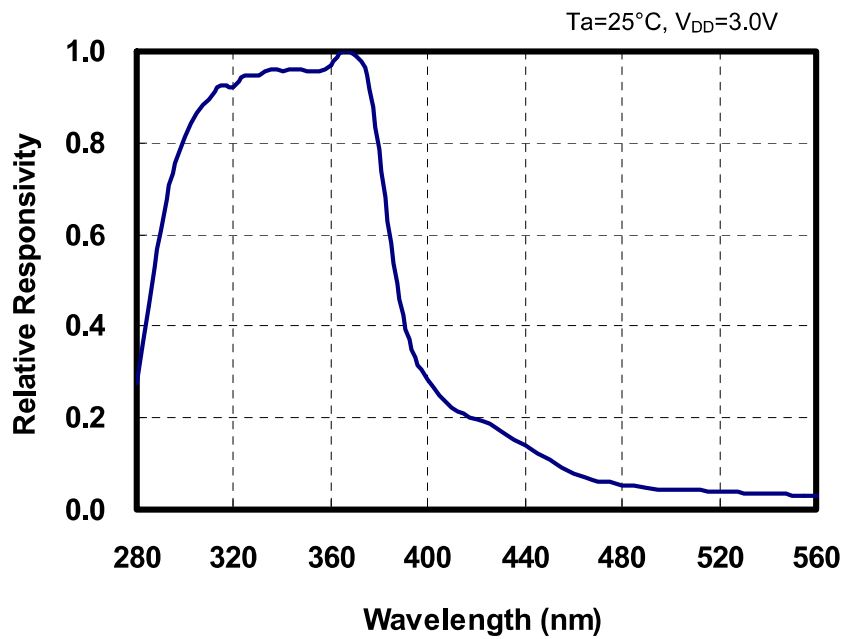
Parameter	Symbol	Condition	Min.	Typ.	Max.	unit
Supply Current (active mode)	I_{DDA}	$V_{EN}=V_{DD}$	-	300	500	μA
Supply Current (standby mode)	I_{DDs}	$V_{EN}=0$	-	0.1	1	μA
Input Voltage (High level)	V_{IH}	-	$V_{DD} \times 0.8$	-	$V_{DD} + 0.3$	V
Input Voltage (Low level)	V_{IL}	-	-0.2	-	0.72	V
High level input current	I_{IH}	$V_{EN}=V_{DD}$	-	-	1	μA
Low level input current	I_{IL}	$V_{EN}=0$	-1	-	-	μA
Wavelength of maximum sensitivity	λ_p	Ta=25°C	-	365	-	nm
Output Setup Time	T_{SU}	$V_{EN}=V_{DD}$	-	-	1	ms
Output Voltage (Shading) *	V_{REF}	Ta=25°C, $V_{EN}=V_{DD}$	0.95	1.0	1.05	V
Output Voltage (10mW/cm ² at λ_p) *	V_O	Ta=25°C, $V_{EN}=V_{DD}$	2.08	2.2	2.32	V

* Load resistance of OUT port is recommended more than 100 kΩ.

OUTPUT VOLTAGE- UV INTENSITY CHARACTERISTICS



SPECTRAL RESPONSIVITY CHARACTERISTICS

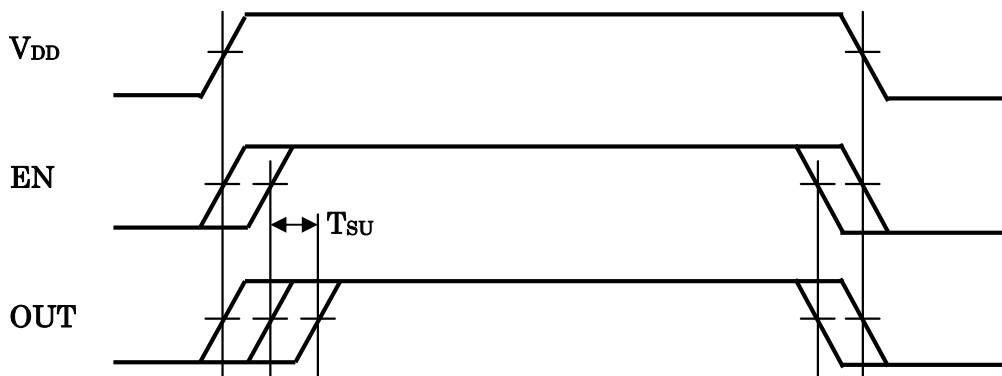


TIMING CHART

Supply voltage and EN signal state should take one of the following procedures:

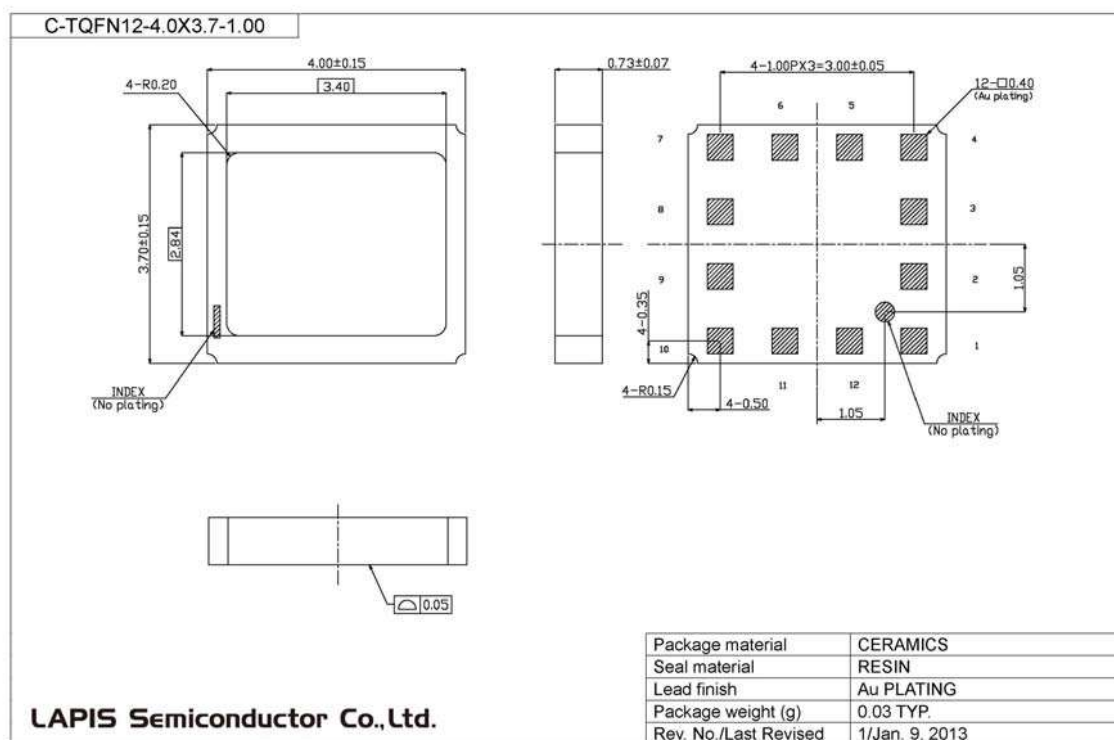
1. EN should be HIGH or LOW at the same time when V_{DD} is applied.
2. EN should be HIGH or LOW while V_{DD} is applied.

Output should be read after output voltage level becomes stable. Maximum time required until stable output voltage reaches is 1 millisecond after EN goes HIGH.



PACKAGE DIMENSIONS

(Unit: mm)



Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact our responsible sales person for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

Specifications are defined without considering the UV absorption by an external cover material. Please contact our sales representatives when the cover material is applied.

Do not press or rub the surface of the resin covering the top of the package where, which is on the UV-ray is light received.

In addition, do not apply pressure at high temperature.

REVISION HISTORY

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEDL851100FC-01	Sep. 3, 2008	-		Preliminary edition
FEDL8511-02	Jan. 24, 2008	-	1	Added disclaimer
FEDL8511-03	Nov. 7, 2008	-	-	Changed Logo OKI to OKI SEMICONDUCTOR
		1	5	The following items has been moved: PACKAGE, NOTICE, PRECAUTION and DISCLAIMER.
		2	1	The following items have been moved: BLOCK DIAGRAM and PIN CONFIGURATION
		3	2	The following tables has been partially modified: RECOMMENDED OPERATING CONDITIONS ELECTRO-OPTICAL CHARACTERISTICS
		3	2	Changed load resistance 500 kΩ to 100 kΩ
		-	3	Added graphs
		-	4	Added "TIMING CHART"
FJDL8511-04	Jan. 17, 2013	1	1	Feature description is removed. Applications section is added.
		4	4	Explanation is provided to the timing chart.
		5	5	Package drawing is updated. Descriptions in the NOTE are modified. DISCLAIMER is removed.
FJDL8511-05	Mar. 8, 2013	1	1	Descriptions in Pin configurations are modified.
		-	2	Example of connecting diagram is added.

NOTES

No copying or reproduction of this document, in part or in whole, is permitted without the consent of LAPIS Semiconductor Co., Ltd.

The content specified herein is subject to change for improvement without notice.

The content specified herein is for the purpose of introducing LAPIS Semiconductor's products (hereinafter "Products"). If you wish to use any such Product, please be sure to refer to the specifications, which can be obtained from LAPIS Semiconductor upon request.

Examples of application circuits, circuit constants and any other information contained herein illustrate the standard usage and operations of the Products. The peripheral conditions must be taken into account when designing circuits for mass production.

Great care was taken in ensuring the accuracy of the information specified in this document. However, should you incur any damage arising from any inaccuracy or misprint of such information, LAPIS Semiconductor shall bear no responsibility for such damage.

The technical information specified herein is intended only to show the typical functions of and examples of application circuits for the Products. LAPIS Semiconductor does not grant you, explicitly or implicitly, any license to use or exercise intellectual property or other rights held by LAPIS Semiconductor and other parties. LAPIS Semiconductor shall bear no responsibility whatsoever for any dispute arising from the use of such technical information.

The Products specified in this document are intended to be used with general-use electronic equipment or devices (such as audio visual equipment, office-automation equipment, communication devices, electronic appliances and amusement devices).

The Products specified in this document are not designed to be radiation tolerant.

While LAPIS Semiconductor always makes efforts to enhance the quality and reliability of its Products, a Product may fail or malfunction for a variety of reasons.

Please be sure to implement in your equipment using the Products safety measures to guard against the possibility of physical injury, fire or any other damage caused in the event of the failure of any Product, such as derating, redundancy, fire control and fail-safe designs. LAPIS Semiconductor shall bear no responsibility whatsoever for your use of any Product outside of the prescribed scope or not in accordance with the instruction manual.

The Products are not designed or manufactured to be used with any equipment, device or system which requires an extremely high level of reliability the failure or malfunction of which may result in a direct threat to human life or create a risk of human injury (such as a medical instrument, transportation equipment, aerospace machinery, nuclear-reactor controller, fuel-controller or other safety device). LAPIS Semiconductor shall bear no responsibility in any way for use of any of the Products for the above special purposes. If a Product is intended to be used for any such special purpose, please contact a ROHM sales representative before purchasing.

If you intend to export or ship overseas any Product or technology specified herein that may be controlled under the Foreign Exchange and the Foreign Trade Law, you will be required to obtain a license or permit under the Law.

BMP280

Digital Pressure Sensor

Bosch Sensortec



BOSCH
Invented for life



BMP280: Data sheet

Document revision 1.19

Document release date January 8th, 2018

Document number BST-BMP280-DS001-19

Technical reference code(s) 0273 300 436

Notes Data in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product's appearance.

BMP280

DIGITAL PRESSURE SENSOR

Key parameters

- Pressure range 300 ... 1100 hPa
(equiv. to +9000...-500 m above/below sea level)
- Package 8-pin LGA metal-lid
Footprint : 2.0 × 2.5 mm², height: 0.95 mm
- Relative accuracy ±0.12 hPa, equiv. to ±1 m
(700 ... 900hPa @ 25°C)
- Absolute accuracy typ. ±1 hPa
(950 ...1050 hPa, 0 ...+40 °C)
- Temperature coefficient offset 1.5 Pa/°C, equiv. to 12.6 cm/°C
(25 ... 40°C @900hPa)
- Digital interfaces I²C (up to 3.4 MHz)
SPI (3 and 4 wire, up to 10 MHz)
- Current consumption 2.7µA @ 1 Hz sampling rate
- Temperature range -40 ... +85 °C
- RoHS compliant, halogen-free
- MSL 1

Typical applications

- Enhancement of GPS navigation
(e.g. time-to-first-fix improvement, dead-reckoning, slope detection)
- Indoor navigation (floor detection, elevator detection)
- Outdoor navigation, leisure and sports applications
- Weather forecast
- Vertical velocity indication (e.g. rise/sink speed)

Target devices

- Handsets such as mobile phones, tablet PCs, GPS devices
- Navigation systems
- Home weather stations
- Flying toys
- Watches

General Description

Robert Bosch is the world market leader for pressure sensors in automotive and consumer applications. Bosch's proprietary APSM (Advanced Porous Silicon Membrane) MEMS manufacturing process is fully CMOS compatible and allows a hermetic sealing of the cavity in an all silicon process. The BMP280 is based on Bosch's proven Piezo-resistive pressure sensor technology featuring high EMC robustness, high accuracy and linearity and long term stability.

The BMP280 is an absolute barometric pressure sensor especially designed for mobile applications. The sensor module is housed in an extremely compact 8-pin metal-lid LGA package with a footprint of only $2.0 \times 2.5 \text{ mm}^2$ and 0.95 mm package height. Its small dimensions and its low power consumption of $2.7 \mu\text{A} @ 1\text{Hz}$ allow the implementation in battery driven devices such as mobile phones, GPS modules or watches.

As the successor to the widely adopted BMP180, the BMP280 delivers high performance in all applications that require precise pressure measurement. The BMP280 operates at lower noise, supports new filter modes and an SPI interface within a footprint 63% smaller than the BMP180.

The emerging applications of indoor navigation, fitness as well as GPS refinement require a high relative accuracy and a low TCO at the same time. BMP180 and BMP280 are perfectly suitable for applications like floor detection since both sensors feature excellent relative accuracy is $\pm 0.12 \text{ hPa}$, which is equivalent to $\pm 1 \text{ m}$ difference in altitude. The very low offset temperature coefficient (TCO) of 1.5 Pa/K translates to a temperature drift of only 12.6 cm/K .

Please contact your regional Bosch Sensortec partner for more information about software packages enhancing the calculation of the altitude given by the BMP280 pressure reading.

Table 1: Comparison between BMP180 and BMP280

Parameter	BMP180	BMP280
Footprint	$3.6 \times 3.8 \text{ mm}$	$2.0 \times 2.5 \text{ mm}$
Minimum V_{DD}	1.80 V	1.71 V
Minimum V_{DDIO}	1.62 V	1.20 V
Current consumption @3 Pa RMS noise	12 μA	2.7 μA
RMS Noise	3 Pa	1.3 Pa
Pressure resolution	1 Pa	0.16 Pa
Temperature resolution	0.1°C	0.01°C
Interfaces	I ² C	I ² C & SPI (3 and 4 wire, mode '00' and '11')
Measurement modes	Only P or T, forced	P&T, forced or periodic
Measurement rate	up to 120 Hz	up to 157 Hz
Filter options	None	Five bandwidths

Index of Contents

1. SPECIFICATIONS	7
2. ABSOLUTE MAXIMUM RATINGS	9
3. FUNCTIONAL DESCRIPTION	10
3.1 BLOCK DIAGRAM	11
3.2 POWER MANAGEMENT	11
3.3 MEASUREMENT FLOW	11
3.3.1 PRESSURE MEASUREMENT	12
3.3.2 TEMPERATURE MEASUREMENT	13
3.3.3 IIR FILTER	13
3.4 FILTER SELECTION	14
3.5 NOISE	15
3.6 POWER MODES	15
3.6.1 SLEEP MODE	16
3.6.2 FORCED MODE	16
3.6.3 NORMAL MODE	16
3.6.4 MODE TRANSITION DIAGRAM	17
3.7 CURRENT CONSUMPTION	18
3.8 MEASUREMENT TIMINGS	18
3.8.1 MEASUREMENT TIME	18
3.8.2 MEASUREMENT RATE IN NORMAL MODE	19
3.9 DATA READOUT	19
3.10 DATA REGISTER SHADOWING	20
3.11 OUTPUT COMPENSATION	20
3.11.1 COMPUTATIONAL REQUIREMENTS	20
3.11.2 TRIMMING PARAMETER READOUT	21
3.11.3 COMPENSATION FORMULA	21
3.12 CALCULATING PRESSURE AND TEMPERATURE	22
4. GLOBAL MEMORY MAP AND REGISTER DESCRIPTION	24
4.1 GENERAL REMARKS	24
4.2 MEMORY MAP	24
4.3 REGISTER DESCRIPTION	24
4.3.1 REGISTER 0xD0 "ID"	24
4.3.2 REGISTER 0xE0 "RESET"	24
4.3.3 REGISTER 0xF3 "STATUS"	25
4.3.4 REGISTER 0xF4 "CTRL_MEAS"	25
4.3.5 REGISTER 0xF5 "CONFIG"	26
4.3.6 REGISTER 0xF7...0xF9 "PRESS" ($_{MSB}$, $_{LSB}$, $_{XLSB}$)	26
4.3.7 REGISTER 0xFA...0xFC "TEMP" ($_{MSB}$, $_{LSB}$, $_{XLSB}$)	27

5. DIGITAL INTERFACES	28
5.1 INTERFACE SELECTION.....	28
5.2 I ² C INTERFACE.....	28
5.2.1 I ² C WRITE.....	29
5.2.2 I ² C READ.....	29
5.3 SPI INTERFACE.....	30
5.3.1 SPI WRITE.....	31
5.3.2 SPI READ.....	31
5.4 INTERFACE PARAMETER SPECIFICATION.....	32
5.4.1 GENERAL INTERFACE PARAMETERS.....	32
5.4.2 I ² C TIMINGS.....	32
5.4.3 SPI TIMINGS.....	33
6. PIN-OUT AND CONNECTION DIAGRAM	35
6.1 PIN-OUT.....	35
6.2 CONNECTION DIAGRAM 4-WIRE SPI.....	36
6.3 CONNECTION DIAGRAM 3-WIRE SPI.....	37
6.4 CONNECTION DIAGRAM I ² C.....	38
7. PACKAGE, REEL AND ENVIRONMENT	39
7.1 OUTLINE DIMENSIONS.....	39
7.2 LANDING PATTERN RECOMMENDATION.....	40
7.3 MARKING.....	41
7.3.1 MASS PRODUCTION DEVICES.....	41
7.3.2 ENGINEERING SAMPLES.....	41
7.4 SOLDERING GUIDELINES.....	42
7.5 TAPE AND REEL SPECIFICATION.....	43
7.5.1 DIMENSIONS.....	43
7.5.2 ORIENTATION WITHIN THE REEL.....	43
7.6 MOUNTING AND ASSEMBLY RECOMMENDATIONS.....	44
7.7 ENVIRONMENTAL SAFETY.....	44
7.7.1 RoHS.....	44
7.7.2 HALOGEN CONTENT.....	44
7.7.3 INTERNAL PACKAGE STRUCTURE.....	44
8. APPENDIX 1: COMPUTATION FORMULAE FOR 32 BIT SYSTEMS	44
8.1 COMPENSATION FORMULA IN FLOATING POINT.....	44
8.2 COMPENSATION FORMULA IN 32 BIT FIXED POINT.....	45
9. LEGAL DISCLAIMER	47
9.1 ENGINEERING SAMPLES.....	47



9.2 PRODUCT USE 47

9.3 APPLICATION EXAMPLES AND HINTS 47

10. DOCUMENT HISTORY AND MODIFICATION 48

1. Specification

If not stated otherwise,

- All values are valid over the full voltage range
- All minimum/maximum values are given for the full accuracy temperature range
- Minimum/maximum values of drifts, offsets and temperature coefficients are $\pm 3\sigma$ values over lifetime
- Typical values of currents and state machine timings are determined at 25 °C
- Minimum/maximum values of currents are determined using corner lots over complete temperature range
- Minimum/maximum values of state machine timings are determined using corner lots over 0...+65 °C temperature range

The specification tables are split into pressure and temperature part of BMP280

Table 2: Parameter specification

Parameter	Symbol	Condition	Min	Typ	Max	Units
Operating temperature range	T_A	operational	-40	25	+85	°C
		full accuracy	0		+65	
Operating pressure range	P	full accuracy	300		1100	hPa
Sensor supply voltage	V_{DD}	ripple max. 50mVpp	1.71	1.8	3.6	V
Interface supply voltage	V_{DDIO}		1.2	1.8	3.6	V
Supply current	$I_{DD,LP}$	1 Hz forced mode, pressure and temperature, lowest power		2.8	4.2	μA
Peak current	I_{peak}	during pressure measurement		720	1120	μA
Current at temperature measurement	I_{DDT}			325		μA
Sleep current ¹	I_{DDSL}	25 °C		0.1	0.3	μA
Standby current (inactive period of normal mode) ²	I_{DDSB}	25 °C		0.2	0.5	μA
Relative accuracy pressure $V_{DD} = 3.3V$	A_{rel}	700 ... 900hPa 25 ... 40 °C		± 0.12		hPa
				± 1.0		m

¹ Typical value at $V_{DD} = V_{DDIO} = 1.8 V$, maximal value at $V_{DD} = V_{DDIO} = 3.6 V$.

² Typical value at $V_{DD} = V_{DDIO} = 1.8 V$, maximal value at $V_{DD} = V_{DDIO} = 3.6 V$.

Offset temperature coefficient	TCO	900hPa 25 ... 40 °C		±1.5		Pa/K
				12.6		cm/K
Absolute accuracy pressure	A ^P _{ext}	300... 1100 hPa -20 ... 0 °C		±1.7		hPa
	A ^P _{full}	300 ... 1100 hPa 0 ... 65 °C		±1.0		hPa
Resolution of output data in ultra high resolution mode	R ^P	Pressure		0.0016		hPa
	R ^T	Temperature		0.01		°C
Noise in pressure	V _{p,full}	Full bandwidth, ultra high resolution See chapter 3.5		1.3		Pa
				11		cm
	V _{p,filtered}	Lowest bandwidth, ultra high resolution See chapter 3.5		0.2		Pa
				1.7		cm
Absolute accuracy temperature ³	A ^T	@ 25 °C		±0.5		°C
		0 ... +65 °C		±1.0		°C
PSRR (DC)	PSRR	full V _{DD} range			±0.005	Pa/ mV
Long term stability ⁴	ΔP _{stab}	12 months		±1.0		hPa
Solder drifts		Minimum solder height 50 μm	-0.5		+2	hPa
Start-up time	t _{startup}	Time to first communication after both V _{DD} > 1.58V and V _{DDIO} > 0.65V			2	ms
Possible sampling rate	f _{sample}	osrs_t = osrs_p = 1; See chapter 3.8	157	182	tbd ⁵	Hz
Standby time accuracy	Δt _{standby}			±5	±25	%

³ Temperature measured by the internal temperature sensor. This temperature value depends on the PCB temperature, sensor element self-heating and ambient temperature and is typically above ambient temperature.

⁴ Long term stability is specified in the full accuracy operating pressure range 0 ... 65 °C

⁵ Depends on application case, please contact Application Engineer for further questions

2. Absolute maximum ratings

The absolute maximum ratings are provided in Table 3.

Table 3: Absolute maximum ratings

Parameter	Condition	Min	Max	Unit
Voltage at any supply pin	V_{DD} and V_{DDIO} Pin	-0.3	4.25	V
Voltage at any interface pin		-0.3	$V_{DDIO} + 0.3$	V
Storage Temperature	$\leq 65\%$ rel. H.	-45	+85	°C
Pressure		0	20 000	hPa
ESD	HBM, at any Pin		± 2	kV
	CDM		± 500	V
	Machine model		± 200	V

3. Functional description

The BMP280 consists of a Piezo-resistive pressure sensing element and a mixed-signal ASIC. The ASIC performs A/D conversions and provides the conversion results and sensor specific compensation data through a digital interface.

BMP280 provides highest flexibility to the designer and can be adapted to the requirements regarding accuracy, measurement time and power consumption by selecting from a high number of possible combinations of the sensor settings.

BMP280 can be operated in three power modes (see chapter 3.6):

- sleep mode
- normal mode
- forced mode

In sleep mode, no measurements are performed. Normal mode comprises an automated perpetual cycling between an active measurement period and an inactive standby period. In forced mode, a single measurement is performed. When the measurement is finished, the sensor returns to sleep mode.

A set of oversampling settings is available ranging from ultra low power to ultra high resolution setting in order to adapt the sensor to the target application. The settings are predefined combinations of pressure measurement oversampling and temperature measurement oversampling. Pressure and temperature measurement oversampling can be selected independently from 0 to 16 times oversampling (see chapter 3.3.1 and 3.3.2):

- Temperature measurement
- Ultra low power
- Low power
- Standard resolution
- High resolution
- Ultra high resolution

BMP280 is equipped with a built-in IIR filter in order to minimize short-term disturbances in the output data caused by the slamming of a door or window. The filter coefficient ranges from 0 (off) to 16.

In order to simplify the device usage and reduce the high number of possible combinations of power modes, oversampling rates and filter settings, Bosch Sensortec provides a proven set of recommendations for common use-cases in smart-phones, mobile weather stations or flying toys (see chapter 3.4):

- Handheld device low-power (e.g. smart phones running Android)
- Handheld device dynamic (e.g. smart phones running Android)
- Weather monitoring (setting with lowest power consumption)
- Elevator / floor change detection
- Drop detection
- Indoor navigation

3.1 Block diagram

Figure 1 shows a simplified block diagram of the BMP280:

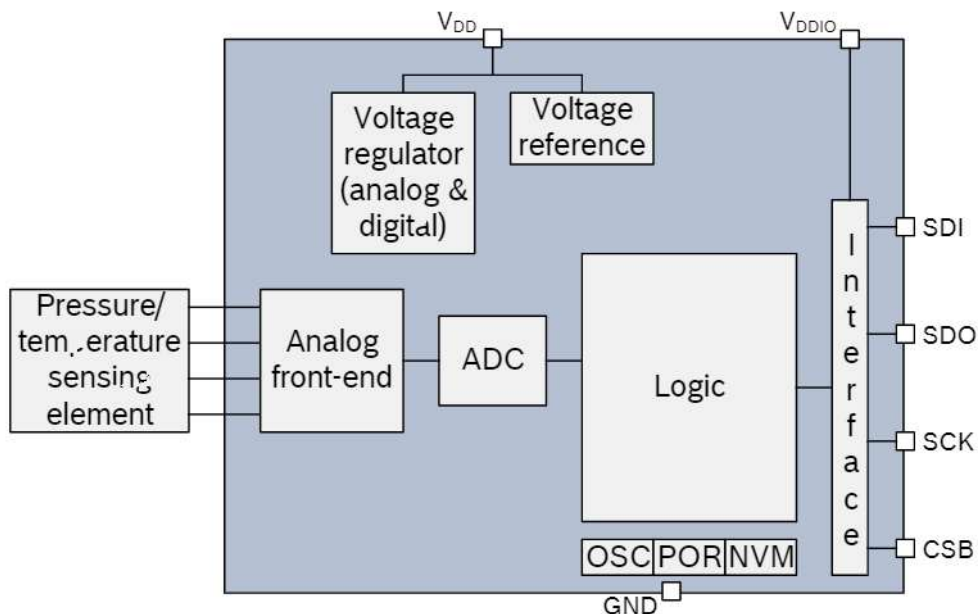


Figure 1: Block diagram of BMP280

3.2 Power management

The BMP280 has two separate power supply pins

- V_{DD} is the main power supply for all internal analog and digital functional blocks
- V_{DDIO} is a separate power supply pin, used for the supply of the digital interface

A power-on reset generator is built in which resets the logic circuitry and the register values after the power-on sequence. There are no limitations on slope and sequence of raising the V_{DD} and V_{DDIO} levels. After powering up, the sensor settles in sleep mode (see 3.6.1).

Warning. Holding any interface pin (SDI, SDO, SCK or CSB) at a logical high level when V_{DDIO} is switched off can permanently damage the device due caused by excessive current flow through the ESD protection diodes.

If V_{DDIO} is supplied, but V_{DD} is not, the interface pins are kept at a high-Z level. The bus can therefore already be used freely before the BMP280 V_{DD} supply is established.

3.3 Measurement flow

The BMP280 measurement period consists of a temperature and pressure measurement with selectable oversampling. After the measurement period, the data are passed through an optional IIR filter, which removes short-term fluctuations in pressure (e.g. caused by slamming a door). The flow is depicted in the diagram below.

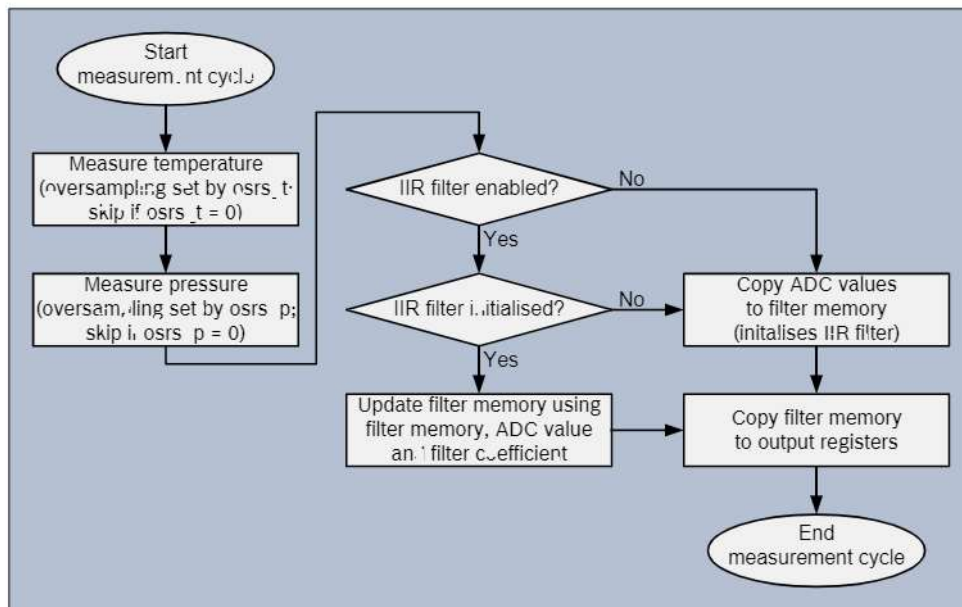


Figure 2: BMP280 measurement cycle

The individual blocks of the diagram above will be detailed in the following subchapters.

3.3.1 Pressure measurement

Pressure measurement can be enabled or skipped. Skipping the measurement could be useful if BMP280 is used as temperature sensor. When enabled, several oversampling options exist. Each oversampling step reduces noise and increases the output resolution by one bit, which is stored in the XLSB data register 0xF9. Enabling/disabling the measurement and oversampling settings are selected through the `osrs_p[2:0]` bits in control register 0xF4.

 Table 4: `osrs_p` settings

Oversampling setting	Pressure oversampling	Typical pressure resolution	Recommended temperature oversampling
Pressure measurement skipped	Skipped (output set to 0x80000)	–	As needed
Ultra low power	×1	16 bit / 2.62 Pa	×1
Low power	×2	17 bit / 1.31 Pa	×1
Standard resolution	×4	18 bit / 0.66 Pa	×1
High resolution	×8	19 bit / 0.33 Pa	×1
Ultra high resolution	×16	20 bit / 0.16 Pa	×2

In order to find a suitable setting for `osrs_p`, please consult chapter 3.4.

3.3.2 Temperature measurement

Temperature measurement can be enabled or skipped. Skipping the measurement could be useful to measure pressure extremely rapidly. When enabled, several oversampling options exist. Each oversampling step reduces noise and increases the output resolution by one bit, which is stored in the XLSB data register 0xFC. Enabling/disabling the temperature measurement and oversampling setting are selected through the `osrs_t[2:0]` bits in control register 0xF4.

Table 5: `osrs_t` settings

<code>osrs_t[2:0]</code>	Temperature oversampling	Typical temperature resolution
000	Skipped (output set to 0x80000)	–
001	×1	16 bit / 0.0050 °C
010	×2	17 bit / 0.0025 °C
011	×4	18 bit / 0.0012 °C
100	×8	19 bit / 0.0006 °C
101, 110, 111	×16	20 bit / 0.0003 °C

It is recommended to base the value of `osrs_t` on the selected value of `osrs_p` as per Table 4. Temperature oversampling above ×2 is possible, but will not significantly improve the accuracy of the pressure output any further. The reason for this is that the noise of the compensated pressure value depends more on the raw pressure than on the raw temperature noise. Following the recommended setting will result in an optimal noise-to-power ratio.

3.3.3 IIR filter

The environmental pressure is subject to many short-term changes, caused e.g. by slamming of a door or window, or wind blowing into the sensor. To suppress these disturbances in the output data without causing additional interface traffic and processor work load, the BMP280 features an internal IIR filter. It effectively reduces the bandwidth of the output signals⁶. The output of a next measurement step is filter using the following formula:

$$data_filtered = \frac{data_filtered_old \cdot (filter_coefficient - 1) + data_ADC}{filter_coefficient}$$

where `data_filtered_old` is the data coming from the previous acquisition, and `data_ADC` is the data coming from the ADC before IIR filtering.

The IIR filter can be configured using the `filter[2:0]` bits in control register 0xF5 with the following options:

⁶ Since most pressure sensors do not sample continuously, filtering can suffer from signals with a frequency higher than the sampling rate of the sensor. E.g. environmental fluctuations caused by windows being opened and closed might have a frequency <5 Hz. Consequently, a sampling rate of ODR = 10 Hz is sufficient to obey the Nyquist theorem.

Table 6: *filter* settings

Filter coefficient	Samples to reach $\geq 75\%$ of step response
Filter off	1
2	2
4	5
8	11
16	22

In order to find a suitable setting for *filter*, please consult chapter 3.4.

When writing to the register *filter*, the filter is reset. The next value will pass through the filter and be the initial memory value for the filter. If temperature or pressure measurement is skipped, the corresponding filter memory will be kept unchanged even though the output registers are set to 0x80000. When the previously skipped measurement is re-enabled, the output will be filtered using the filter memory from the last time when the measurement was not skipped.

3.4 Filter selection

In order to select optimal settings, the following use cases are suggested:

Table 7: Recommended filter settings based on use cases

Use case	Mode	Over-sampling setting	osrs_p	osrs_t	IIR filter coeff. (see 3.3.3)	I _{DD} [μ A] (see 3.7)	ODR [Hz] (see 3.8.2)	RMS Noise [cm] (see 3.5)
handheld device low-power (e.g. Android)	Normal	Ultra high resolution	$\times 16$	$\times 2$	4	247	10.0	4.0
handheld device dynamic (e.g. Android)	Normal	Standard resolution	$\times 4$	$\times 1$	16	577	83.3	2.4
Weather monitoring (lowest power)	Forced	Ultra low power	$\times 1$	$\times 1$	Off	0.14	1/60	26.4
Elevator / floor change detection	Normal	Standard resolution	$\times 4$	$\times 1$	4	50.9	7.3	6.4
Drop detection	Normal	Low power	$\times 2$	$\times 1$	Off	509	125	20.8
Indoor navigation	Normal	Ultra high resolution	$\times 16$	$\times 2$	16	650	26.3	1.6

3.5 Noise

Noise depends on the oversampling and filter settings selected. The stated values were determined in a controlled pressure environment and are based on the average standard deviation of 32 consecutive measurement points taken at highest sampling speed. This is needed in order to exclude long term drifts from the noise measurement.

Table 8: Noise in pressure

Typical RMS noise in pressure [Pa]					
Oversampling setting	IIR filter coefficient				
	off	2	4	8	16
Ultra low power	3.3	1.9	1.2	0.9	0.4
Low power	2.6	1.5	1.0	0.6	0.4
Standard resolution	2.1	1.2	0.8	0.5	0.3
High resolution	1.6	1.0	0.6	0.4	0.2
Ultra high resolution	1.3	0.8	0.5	0.4	0.2

Table 9: Noise in temperature

Typical RMS noise in temperature [°C]	
Temperature oversampling	IIR filter off
oversampling ×1	0.005
oversampling ×2	0.004
oversampling ×4	0.003
oversampling ×8	0.003
oversampling ×16	0.002

3.6 Power modes

The BMP280 offers three power modes: sleep mode, forced mode and normal mode. These can be selected using the mode[1:0] bits in control register 0xF4.

Table 10: mode settings

mode[1:0]	Mode
00	Sleep mode
01 and 10	Forced mode
11	Normal mode

3.6.1 Sleep mode

Sleep mode is set by default after power on reset. In sleep mode, no measurements are performed and power consumption (I_{DDSM}) is at a minimum. All registers are accessible; Chip-ID and compensation coefficients can be read.

3.6.2 Forced mode

In forced mode, a single measurement is performed according to selected measurement and filter options. When the measurement is finished, the sensor returns to sleep mode and the measurement results can be obtained from the data registers. For a next measurement, forced mode needs to be selected again. This is similar to BMP180 operation. Forced mode is recommended for applications which require low sampling rate or host-based synchronization.

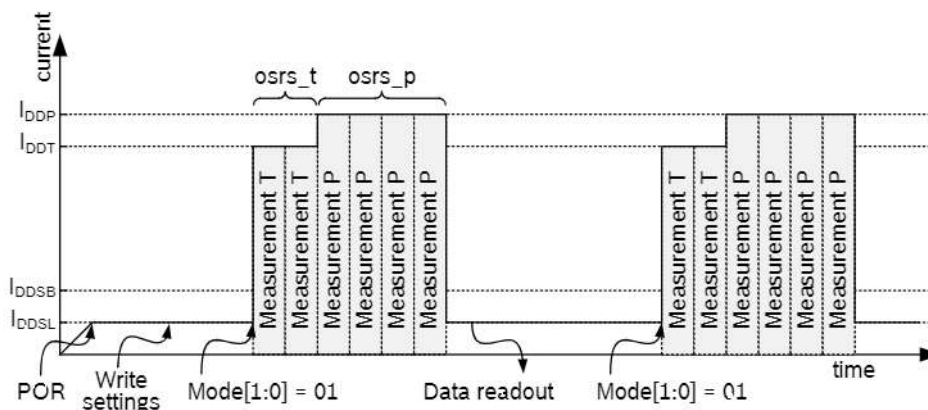


Figure 3: Forced mode timing diagram

3.6.3 Normal mode

Normal mode continuously cycles between an (active) measurement period and an (inactive) standby period, whose time is defined by $t_{standby}$. The current in the standby period (I_{DDSB}) is slightly higher than in sleep mode. After setting the mode, measurement and filter options, the last measurement results can be obtained from the data registers without the need of further write accesses. Normal mode is recommended when using the IIR filter, and useful for applications in which short term disturbances (e.g. blowing into the sensor) should be filtered.

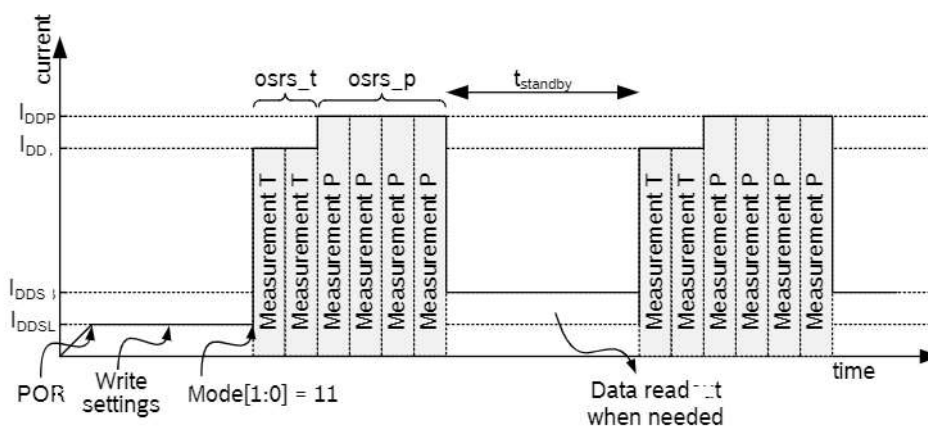


Figure 4: Normal mode timing diagram

The standby time is determined by the contents of the $t_sb[2:0]$ bits in control register 0x1F5 according to the table below:

Table 11: t_sb settings

$t_sb[1:0]$	$t_{standby}$ [ms]
000	0.5
001	62.5
010	125
011	250
100	500
101	1000
110	2000
111	4000

3.6.4 Mode transition diagram

The supported mode transitions are displayed below. If the device is currently performing a measurement, execution of mode switching commands is delayed until the end of the currently running measurement period. Further mode change commands are ignored until the last mode change command is executed. Mode transitions other than the ones shown below are tested for stability but do not represent recommended use of the device.

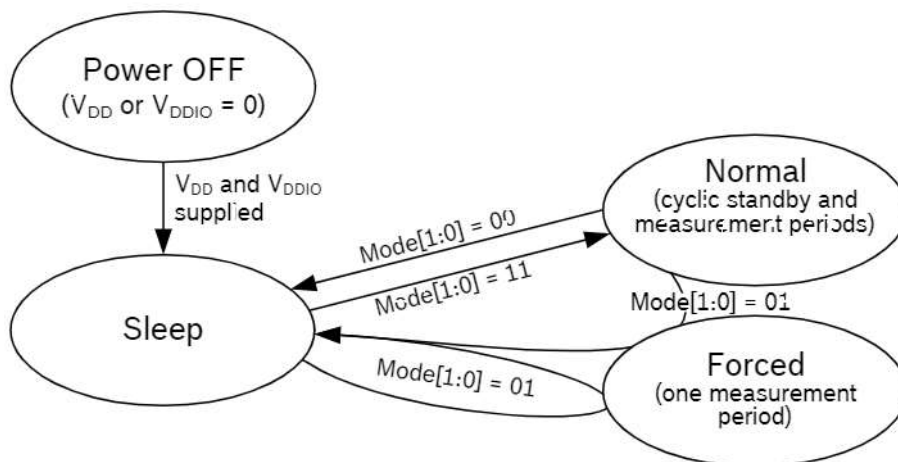


Figure 5: Mode transition diagram

3.7 Current consumption

The current consumption depends on ODR and oversampling setting. The values given below are normalized to an ODR of 1 Hz. The actual consumption at a given ODR can be calculated by multiplying the consumption in Table 12 with the ODR used. The actual ODR is defined either by the frequency at which the user sets forced measurements or by oversampling and t_{standby} settings in normal mode in Table 14.

Table 12: Current consumption

Oversampling setting	Pressure oversampling	Temperature oversampling	I _{DD} [μA] @ 1 Hz forced mode	
			Typ	Max
Ultra low power	×1	×1	2.74	4.16
Low power	×2	×1	4.17	6.27
Standard resolution	×4	×1	7.02	10.50
High resolution	×8	×1	12.7	18.95
Ultra high resolution	×16	×2	24.8	36.85

3.8 Measurement timings

The rate at which measurements can be performed in forced mode depends on the oversampling settings $osrs_t$ and $osrs_p$. The rate at which they are performed in normal mode depends on the oversampling setting settings $osrs_t$ and $osrs_p$ and the standby time t_{standby} . In the following table the resulting ODRs are given only for the suggested $osrs$ combinations.

3.8.1 Measurement time

The following table explains the typical and maximum measurement time based on selected oversampling setting. The minimum achievable frequency is determined by the maximum measurement time.

Table 13: measurement time

Oversampling setting	Pressure oversampling	Temperature oversampling	Measurement time [ms]		Measurement rate [Hz]	
			Typ	Max	Typ	Min
Ultra low power	×1	×1	5.5	6.4	181.8	155.6
Low power	×2	×1	7.5	8.7	133.3	114.6
Standard resolution	×4	×1	11.5	13.3	87.0	75.0
High resolution	×8	×1	19.5	22.5	51.3	44.4
Ultra high resolution	×16	×2	37.5	43.2	26.7	23.1

3.8.2 Measurement rate in normal mode

The following table explains which measurement rates can be expected in normal mode based on oversampling setting and t_{standby} .

Table 14: typical output data Rate (ODR) in normal mode [Hz]

Oversampling setting	t_{standby} [ms]							
	0.5	62.5	125	250	500	1000	2000	4000
Ultra low power	166.67	14.71	7.66	3.91	1.98	0.99	0.50	0.25
Low power	125.00	14.29	7.55	3.88	1.97	0.99	0.50	0.25
Standard resolution	83.33	13.51	7.33	3.82	1.96	0.99	0.50	0.25
High resolution	50.00	12.20	6.92	3.71	1.92	0.98	0.50	0.25
Ultra high resolution	26.32	10.00	6.15	3.48	1.86	0.96	0.49	0.25

Table 15: Sensor timing according to recommended settings (based on use cases)

Use case	Mode	Over-sampling setting	osrs_p	osrs_t	IIR filter coeff. (see 3.3.3)	Timing	ODR [Hz] (see 3.8.2)	BW [Hz] (see 3.3.3)
handheld device low-power (e.g. Android)	Normal	Ultra high resolution	×16	×2	4	$t_{\text{standby}} = 62.5 \text{ ms}$	10.0	0.92
handheld device dynamic (e.g. Android)	Normal	Standard resolution	×4	×1	16	$t_{\text{standby}} = 0.5 \text{ ms}$	83.3	1.75
Weather monitoring (lowest power)	Forced	Ultra low power	×1	×1	Off	1/min	1/60	full
Elevator / floor change detection	Normal	Standard resolution	×4	×1	4	$t_{\text{standby}} = 125 \text{ ms}$	7.3	0.67
Drop detection	Normal	Low power	×2	×1	Off	$t_{\text{standby}} = 0.5 \text{ ms}$	125	full
Indoor navigation	Normal	Ultra high resolution	×16	×2	16	$t_{\text{standby}} = 0.5 \text{ ms}$	26.3	0.55

3.9 Data readout

To read out data after a conversion, it is strongly recommended to use a burst read and not address every register individually. This will prevent a possible mix-up of bytes belonging to different measurements and reduce interface traffic. Data readout is done by starting a burst read from 0xF7 to 0xFC. The data are read out in an unsigned 20-bit format both for pressure and for temperature. It is strongly recommended to use the BMP280 API, available from Bosch

Sensortec, for readout and compensation. For details on memory map and interfaces, please consult chapters 3.12 and 5 respectively.

The timing for data readout in forced mode should be done so that the maximum measurement times (see chapter 3.8.1) are respected. In normal mode, readout can be done at a speed similar to the expected data output rate (see chapter 3.8.2). After the values of 'ut' and 'up' have been read, the actual pressure and temperature need to be calculated using the compensation parameters stored in the device. The procedure is elaborated in chapter 3.11.

3.10 Data register shadowing

In normal mode, measurement timing is not necessarily synchronized to readout. This means that new measurement results may become available while the user is reading the results from the previous measurement. In this case, shadowing is performed in order to guarantee data consistency. Shadowing will only work if all data registers are read in a single burst read. Therefore, the user must use burst reads if he does not synchronize data readout with the measurement cycle. Using several independent read commands may result in inconsistent data.

If a new measurement is finished and the data registers are still being read, the new measurement results are transferred into shadow data registers. The content of shadow registers is transferred into data registers as soon as the user ends the burst read, even if not all data registers were read. Reading across several data registers can therefore only be guaranteed to be consistent within one measurement cycle if a single burst read command is used. The end of the burst read is marked by the rising edge of CSB pin in SPI case or by the recognition of a stop condition in I2C case. After the end of the burst read, all user data registers are updated at once.

3.11 Output compensation

The BMP280 output consists of the ADC output values. However, each sensing element behaves differently, and actual pressure and temperature must be calculated using a set of calibration parameters. The recommended calculation in chapter 3.11.3 uses fixed point arithmetic. In high-level languages like Matlab™ or LabVIEW™, fixed-point code may not be well supported. In this case the floating-point code in appendix 8.1 can be used as an alternative. For 8-bit micro controllers, the variable size may be limited. In this case a simplified 32 bit integer code with reduced accuracy is given in appendix 8.2.

3.11.1 Computational requirements

The table below shows the number of clock cycles needed for compensation calculations on a 32 bit Cortex-M3 micro controller with GCC optimization level -O2. This controller does not contain a floating point unit, so all floating-point calculations are emulated. Floating point is only recommended for PC applications where an FPU is present.

Table 16: Computational requirements for compensation formulas

Compensation of	Number of clock cycles (ARM Cortex-M3)		
	32 bit integer	64 bit integer	Double precision
Temperature	~46	–	~2400 ⁷
Pressure	~112 ⁸	~1400	~5400 ⁷

3.11.2 Trimming parameter readout

The trimming parameters are programmed into the devices' non-volatile memory (NVM) during production and cannot be altered by the customer. Each compensation word is a 16-bit signed or unsigned integer value stored in two's complement. As the memory is organized into 8-bit words, two words must always be combined in order to represent the compensation word. The 8-bit registers are named calib00...calib25 and are stored at memory addresses 0x88...0xA1. The corresponding compensation words are named dig_T# for temperature compensation related values and dig_P# for pressure compensation related values. The mapping is shown in Table 17.

Table 17: Compensation parameter storage, naming and data type

Register Address LSB / MSB	Register content	Data type
0x88 / 0x89	dig_T1	unsigned short
0x8A / 0x8B	dig_T2	signed short
0x8C / 0x8D	dig_T3	signed short
0x8E / 0x8F	dig_P1	unsigned short
0x90 / 0x91	dig_P2	signed short
0x92 / 0x93	dig_P3	signed short
0x94 / 0x95	dig_P4	signed short
0x96 / 0x97	dig_P5	signed short
0x98 / 0x99	dig_P6	signed short
0x9A / 0x9B	dig_P7	signed short
0x9C / 0x9D	dig_P8	signed short
0x9E / 0x9F	dig_P9	signed short
0xA0 / 0xA1	reserved	reserved

3.11.3 Compensation formula

Please note that it is strongly advised to use the API available from Bosch Sensortec to perform readout and compensation. If this is not wanted, the code below can be applied at the user's risk. Both pressure and temperature values are expected to be received in 20 bit format, positive, stored in a 32 bit signed integer.

⁷ Use only recommended for high-level programming languages like Matlab™ or LabV.EW™

⁸ Use only recommended for 8-bit micro controllers

The variable `t_fine` (signed 32 bit) carries a fine resolution temperature value over to the pressure compensation formula and could be implemented as a global variable.

The data type “BMP280_S32_t” should define a 32 bit signed integer variable type and can usually be defined as “long signed int”.

The data type “BMP280_U32_t” should define a 32 bit unsigned integer variable type and can usually be defined as “long unsigned int”.

For best possible calculation accuracy, 64 bit integer support is needed. If this is not possible on your platform, please see appendix 8.2 for a 32 bit alternative.

The data type “BMP280_S64_t” should define a 64 bit signed integer variable type, which on most supporting platforms can be defined as “long long signed int”. The revision of the code is rev.1.1.

```
// Returns temperature in DegC, resolution is 0.01 DegC. Output value of "5123" equals 51.23 DegC.
// t_fine carries fine temperature as global value
BMP280_U32_t t_fine;
BMP280_S32_t bmp280_compensate_T_int32(BMP280_S32_t adc_T)
{
    BMP280_S32_t var1, var2, T;
    var1 = (((adc_T >> 3) - ((BMP280_S32_t)dig_T1 << 1)) * ((BMP280_S32_t)dig_T2) >> 11);
    var2 = (((((adc_T >> 4) - ((BMP280_S32_t)dig_T1)) * ((adc_T >> 4) - ((BMP280_S32_t)dig_T1))) >> 12) +
            ((BMP280_S32_t)dig_T3)) >> 1;
    t_fine = var1 + var2;
    T = (t_fine * 5 + 128) >> 8;
    return T;
}

// Returns pressure in Pa as unsigned 32 bit integer in Q2..8 format (24 integer bits and 8 fractional bits).
// Output value of "2167867" represents 24674867/256 = 96386.2 Pa = 963.862 hPa
BMP280_U32_t bmp280_compensate_P_int64(BMP280_S32_t adc_P)
{
    BMP280_S64_t var1, var2, p;
    var1 = ((BMP280_S64_t)t_fine) - 128000;
    var2 = var1 * var1 * (BMP280_S64_t)dig_P6;
    var2 = var2 + ((var1 * (BMP280_S64_t)dig_P5) << 17);
    var2 = var2 + (((BMP280_S64_t)dig_P4) << 35);
    var1 = ((var1 * var1 * (BMP280_S64_t)dig_P3) >> 8) + ((var1 * (BMP280_S64_t)dig_P2) << 12);
    var1 = (((((BMP280_S64_t)1) < 47) + var1)) * ((BMP280_S64_t)dig_P1) >> 3;
    if (var1 == 0)
    {
        return 0; // avoid exception caused by division by zero
    }
    p = 1048576 - adc_P;
    p = (((1 << 31) - var2) * 3125) / var1;
    var1 = (((BMP280_S64_t)dig_P9) * (p >> 13) * (p >> 13)) >> 25;
    var2 = (((BMP280_S64_t)dig_P8) * p) >> 19;
    p = ((p + var1 + var2) >> 8) + (((BMP280_S64_t)dig_P7) << 4);
    return (BMP280_U32_t)p;
}
```

3.12 Calculating pressure and temperature

The following figure shows the detailed algorithm for pressure and temperature measurement.

This algorithm is available to customers as reference C source code (“BMP28x_API”) from Bosch Sensortec and via its sales and distribution partners.

Please contact your Bosch Sensortec representative for details.

**Calculation of pressure and temperature for BMP280**

Sample trimming values			
Register Address (LSB / MSB)	Name	Value	Type
0x88 / 0x89	dig_T1	27504	unsigned short
0x8A / 0x8B	dig_T2	26435	short
0x8C / 0x8D	dig_T3	-1000	short
0x8E / 0x8F	dig_P1	36477	unsigned short
0x90 / 0x91	dig_P2	-10685	short
0x92 / 0x93	dig_P3	3024	short
0x94 / 0x95	dig_P4	2855	short
0x96 / 0x97	dig_P5	140	short
0x98 / 0x99	dig_P6	-7	short
0x9A / 0x9B	dig_P7	15500	short
0x9C / 0x9D	dig_P8	-14600	short
0x9E / 0x9F	dig_P9	6000	short
0xA0 / 0xA1			

Sample measurement values			
Register Address (MSB / LSB / XLSB)	Name	Value	Type
0xF7 / 0xF8 / 0xF9[7:4]	UT [20 bit]	519888	signed long (*)
0xFA / 0xFB / 0xFC[7:4]	UP [20 bit]	415148	signed long (*)

(*) Value is always positive, even though the compensation functions expect a signed integer as input
 (*) Value is always positive, even though the compensation functions expect a signed integer as input

```

var1 = 128793,1787
var2 = -370,8917052
tline = 128422
T = 25,08
integer result (**): 2508
Temperature [°C]
Temperature [1/100 °C]

var1 = 211,1435029
var2 = -9,523652701
var2 = 59110,65716
var2 = 187120057,7
var1 = -4,302618389
var1 = 36472,21037
p = 633428
p = 100717,8456
var1 = 28342,24444
var2 = -44875,50492
p = 100653,27
int32 result (**): 100653
int64 result (**): 25767236
Pressure [Pa]
Pressure [Pa]
Pressure [1/256 Pa]

var1 = (((double)adc_T)/6284.0 - ((double)dig_T1)/624.0) * ((double)dig_T2);
var2 = (((double)adc_T)/31072.0 - ((double)dig_T1)/312.0) * (((double)adc_T)/31072.0 - ((double)dig_T1)/312.0) * ((double)dig_T3);
t_line = (BMP280_S32_t)var1 + var2;
T = (var1 + var2) / 5120.0;

var1 = ((double)t_line) / 2.0 - 64000.0;
var2 = var1 * var1 * ((double)dig_P6) / 32768.0;
var2 = var2 + var1 * ((double)dig_P5) * 2.0;
var2 = (var2/4.0) + ((double)dig_P4) * 69536.0;
var1 = (((double)dig_P3) * var1 + var1 * 524288.0 - ((double)dig_P2) * var1) / 524288.0;
var1 = (1.0 - var1 / 32768.0) * ((double)dig_P1);
p = 1048576.0 - (double)adc_P;
p = (p - (var2 / 4096.0)) * 6250.0 / var1;
var1 = ((double)dig_P9) * p * p / 2147483648.0;
var2 = p * ((double)dig_P8) / 32768.0;
p = p + (var1 + var2 * ((double)dig_P7)) / 16.0;

```

(**) The actual result of the integer calculation may deviate slightly from the values shown here due to integer calculation rounding errors



4. Global memory map and register description

4.1 General remarks

All communication with the device is performed by reading from and writing to registers. Registers have a width of 8 bits. There are several registers which are reserved; they should not be written to and no specific value is guaranteed when they are read. For details on the interface, consult chapter 5.

4.2 Memory map

The memory map is given in Table 18 below. Reserved registers are not shown.

Table 18: Memory map

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state	
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>									0x00
temp_msb	0xFA	temp_msb<7:0>									0x80
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0	0x00
press_lsb	0xF8	press_lsb<7:0>									0x00
press_msb	0xF7	press_msb<7:0>									0x80
ctrl_reg	0xF6	t_sb[2:0]		_filter[2:0]				spi3w_en[0]		0x00	
ctrl_reg	0xF4	csrs_t[2:0]		osrs_p[2:0]				mode[1:0]		0x00	
status	0xF3			iir_asuring[0]				iir_pdate[0]		0x00	
reset	0xE0	reset[7:0]									0x00
id	0xD0	chip_id[7:0]									0x58
calib25..calib00	0xA1...0x8	calibration data									individual

Registers:	Reserved registers	Calibration data	Control registers	Data registers	Status registers	Revision	Reset
Type:	do not write	read only	read, write	read only	read only	read only	write only

4.3 Register description

4.3.1 Register 0xD0 “id”

The “id” register contains the chip identification number chip_id[7:0], which is 0x58. This number can be read as soon as the device finished the power-on-reset.

4.3.2 Register 0xE0 “reset”

The “reset” register contains the soft reset word reset[7:0]. If the value 0xB6 is written to the register, the device is reset using the complete power-on-reset procedure. Writing other values than 0xB6 has no effect. The readout value is always 0x00.

4.3.3 Register 0xF3 “status”

The “status” register contains two bits which indicate the status of the device.

Table 19: Register 0xF3 “status”

Register 0xF3 “status”	Name	Description
Bit 3	measuring[0]	Automatically set to ‘1’ whenever a conversion is running and back to ‘0’ when the results have been transferred to the data registers.
Bit 0	im_update[0]	Automatically set to ‘1’ when the NVM data are being copied to image registers and back to ‘0’ when the copying is done. The data are copied at power-on-reset and before every conversion.

4.3.4 Register 0xF4 “ctrl_meas”

The “ctrl_meas” register sets the data acquisition options of the device.

Table 20: Register 0xF4 “ctrl_meas”

Register 0xF4 “ctrl_meas”	Name	Description
Bit 7, 6, 5	osrs_t[2:0]	Controls oversampling of temperature data. See chapter 3.3.2 for details.
Bit 4, 3, 2	osrs_p[2:0]	Controls oversampling of pressure data. See chapter 3.3.1 for details.
Bit 1, 0	mode[1:0]	Controls the power mode of the device. See chapter 3.6 for details.

Table 21: register settings *osrs_p*

<i>osrs_p</i> [2:0]	Pressure oversampling
000	Skipped (output set to 0x80000)
001	oversampling ×1
010	oversampling ×2
011	oversampling ×4
100	oversampling ×8
101, Others	oversampling ×16

Table 22: register settings *osrs_t*

<i>osrs_t</i> [2:0]	Temperature oversampling
000	Skipped (output set to 0x80000)
001	oversampling ×1
010	oversampling ×2
011	oversampling ×4
100	oversampling ×8
101, 110, 111	oversampling ×16

4.3.5 Register 0xF5 “*config*”

The “*config*” register sets the rate, filter and interface options of the device. Writes to the “*config*” register in normal mode may be ignored. In sleep mode writes are not ignored.

 Table 23: Register 0xF5 “*config*”

Register 0xF5 “ <i>config</i> ”	Name	Description
Bit 7, 6, 5	<i>t_sb</i> [2:0]	Controls inactive duration <i>t_{standby}</i> in normal mode. See chapter 3.6.3 for details.
Bit 4, 3, 2	<i>filter</i> [2:0]	Controls the time constant of the IIR filter. See chapter 3.3.3 for details.
Bit 0	<i>spi3w_en</i> [0]	Enables 3-wire SPI interface when set to ‘1’. See chapter 5.3 for details.

4.3.6 Register 0xF7...0xF9 “*press*” (*_msb*, *_lsb*, *_xlsb*)

The “*press*” register contains the raw pressure measurement output data up[19:0]. For details on how to read out the pressure and temperature information from the device, please consult chapter 3.9.

 Table 24: Register 0xF7 ... 0xF9 “*press*”

Register 0xF7-0xF9 “ <i>press</i> ”	Name	Description
0xF7	<i>press_msb</i> [7:0]	Contains the MSB part up[19:12] of the raw pressure measurement output data.
0xF8	<i>press_lsb</i> [7:0]	Contains the LSB part up[11:4] of the raw pressure measurement output data.
0xF9 (bit 7, 6, 5, 4)	<i>press_xlsb</i> [3:0]	Contains the XLSB part up[3:0] of the raw pressure measurement output data. Contents depend on temperature resolution, see table 5.

4.3.7 Register 0xFA...0xFC “temp” (*_msb, _lsb, _x!sb*)

The “temp” register contains the raw temperature measurement output data $ut[19:0]$. For details on how to read out the pressure and temperature information from the device, please consult chapter 3.9.

Table 25: Register 0xFA ... 0xFC “temp”

Register 0xF7-0xF9 “press”	Name	Description
0xFA	temp_msb[7:0]	Contains the MSB part $ut[19:12]$ of the raw temperature measurement output data.
0xFB	temp_lsb[7:0]	Contains the LSB part $ut[11:4]$ of the raw temperature measurement output data.
0xFC (bit 7, 6, 5, 4)	temp_xlsb[3:0]	Contains the XLSB part $ut[3:0]$ of the raw temperature measurement output data. Contents depend on pressure resolution, see Table 4.

5. Digital interfaces

The BMP280 supports the I²C and SPI digital interfaces; it acts as a slave for both protocols. The I²C interface supports the Standard, Fast and High Speed modes. The SPI interface supports both SPI mode '00' (CPOL = CPHA = '0') and mode '11' (CPOL = CPHA = '1') in 4-wire and 3-wire configuration.

The following transactions are supported:

- Single byte write
- multiple byte write (using pairs of register addresses and register data)
- single byte read
- multiple byte read (using a single register address which is auto-incremented)

5.1 Interface selection

Interface selection is done automatically based on CSB (chip select) status. If CSB is connected to V_{DDIO}, the I²C interface is active. If CSB is pulled down, the SPI interface is activated. After CSB has been pulled down once (regardless of whether any clock cycle occurred), the I²C interface is disabled until the next power-on-reset. This is done in order to avoid inadvertently decoding SPI traffic to another slave as I²C data. Since power-on-reset is only executed when both V_{DD} and V_{DDIO} are established, there is no risk of incorrect protocol detection due to power-up sequence used. However, if I²C is to be used and CSB is not directly connected to V_{DDIO} but rather through a programmable pin, it must be ensured that this pin already outputs the V_{DDIO} level during power-on-reset of the device. If this is not the case, the device will be locked in SPI mode and not respond to I²C commands.

5.2 I²C Interface

The I²C slave interface is compatible with Philips I²C Specification version 2.1. For detailed timings refer to Table 27. All modes (standard, fast, high speed) are supported. SDA and SCL are not pure open-drain. Both pads contain ESD protection diodes to V_{DDIO} and GND. As the device does not perform clock stretching, the SCL structure is a high-Z input without drain capability.

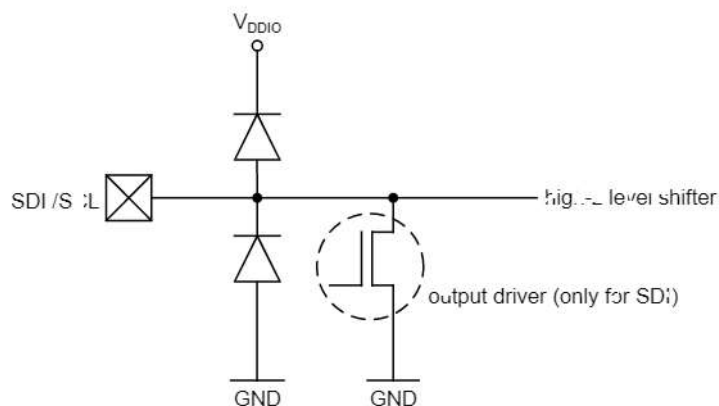


Figure 6: SDI/SCK ESD drawing

The 7-bit device address is 111011x. The 6 MSB bits are fixed. The last bit is changeable by SDO value and can be changed during operation. Connecting SDO to GND results in slave address 1110110 (0x76); connection it to V_{DDIO} results in slave address 1110111 (0x77), which is the same

as BMP180's I²C address. The SDO pin cannot be left floating; if left floating, the I²C address will be undefined.

The I²C interface uses the following pins:

- SCK: serial clock (SCL)
- SDI: data (SDA)
- SDO: Slave address LSB (GND = '0', V_{DDIO} = '1')

CSB must be connected to V_{DDIO} to select I²C interface. SDI is bi-directional with open drain to GND: it must be externally connected to V_{DDIO} via a pull up resistor. Refer to chapter 6 for connection instructions.

The following abbreviations will be used in the I²C protocol figures:

- S Start
- P Stop
- ACKS Acknowledge by slave
- ACKM Acknowledge by master
- NACKM Not acknowledge by master

5.2.1 I²C write

Writing is done by sending the slave address in write mode (RW = '0'), resulting in slave address 111011X0 ('X' is determined by state of SDO pin). Then the master sends pairs of register addresses and register data. The transaction is ended by a stop condition. This is depicted in Figure 7.

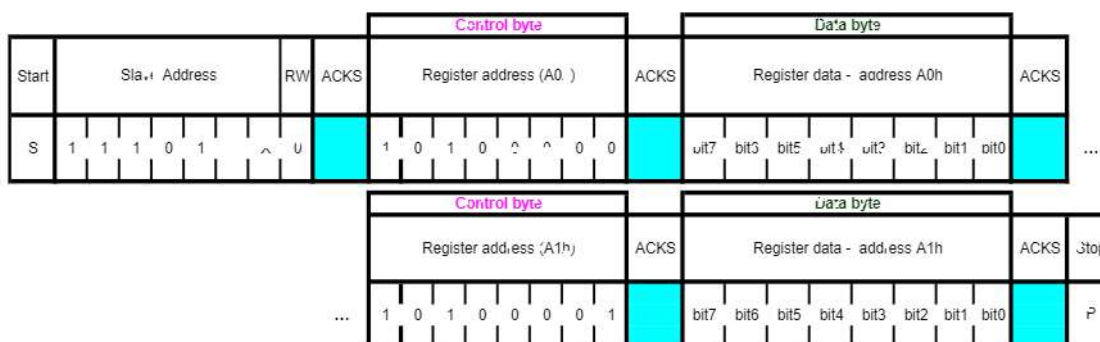


Figure 7: I²C multiple byte write (not auto-incremented)

5.2.2 I²C read

To be able to read registers, first the register address must be sent in write mode (slave address 111011X0). Then either a stop or a repeated start condition must be generated. After this the slave is addressed in read mode (RW = '1') at address 111011X1, after which the slave sends out data from auto-incremented register addresses until a NACKM and stop condition occurs. This is depicted in Figure 8, where two bytes are read from register 0xF6 and 0xF7.



Figure 8: I²C multiple byte read

5.3 SPI interface

The SPI interface is compatible with SPI mode '00' (CPOL = CPHA = '0') and mode '11' (CPOL = CPHA = '1'). The automatic selection between mode '00' and '11' is determined by the value of SCK after the CSB falling edge.

The SPI interface has two modes: 4-wire and 3-wire. The protocol is the same for both. The 3-wire mode is selected by setting '1' to the register spi3w_en. The pad SDI is used as a data pad in 3-wire mode.

The SPI interface uses the following pins:

- CSB: chip select, active low
- SCK: serial clock
- SDI: serial data input; data input/output in 3-wire mode
- SDO: serial data output; hi-Z in 3-wire mode

Refer to chapter 6 for connection instructions.

CSB is active low and has an integrated pull-up resistor. Data on SDI is latched by the device at SCK rising edge and SDO is changed at SCK falling edge. Communication starts when CSB goes to low and stops when CSB goes to high; during these transitions on CSB, SCK must be stable. The SPI protocol is shown in Figure 9. For timing details, please review Table 28.

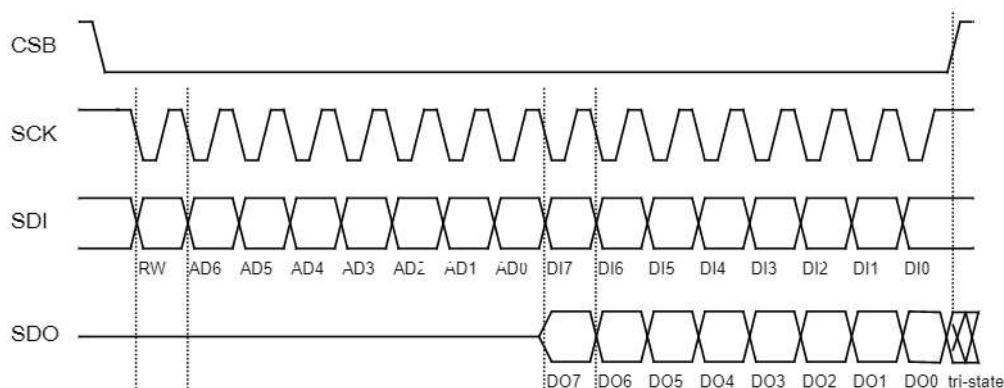


Figure 9: SPI protocol (shown for mode '11' in 4-wire configuration)

In SPI mode, only 7 bits of the register addresses are used; the MSB of register address is not used and replaced by a read/write bit (RW = '0' for write and RW = '1' for read).

Example: address 0xF7 is accessed by using SPI register address 0x77. For write access, the byte 0x77 is transferred, for read access, the byte 0xF7 is transferred.

5.3.1 SPI write

Writing is done by lowering CSB and sending pairs control bytes and register data. The control bytes consist of the SPI register address (= full register address without bit 7) and the write command (bit7 = RW = '0'). Several pairs can be written without raising CSB. The transaction is ended by a raising CSB. The SPI write protocol is depicted in Figure 10.

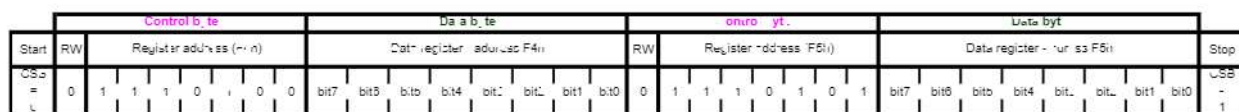


Figure 10: SPI multiple byte write (not auto-incremented)

5.3.2 SPI read

Reading is done by lowering CSB and first sending one control byte. The control bytes consist of the SPI register address (= full register address without bit 7) and the read command (bit 7 = RW = '1'). After writing the control byte, data is sent out of the SDO pin (SDI in 3-wire mode); the register address is automatically incremented. The SPI read protocol is shown in Figure 11.



Figure 11: SPI multiple byte read

5.4 Interface parameter specification

5.4.1 General interface parameters

The general interface parameters are given in Table 26 below.

Table 26: interface parameters

Parameter	Symbol	Condition	Min	Typ	Max	Units
Input – low level	Vil_si	V _{DDIO} =1.2V to 3.6V			0.2 * V _{DDIO}	V
Input – high level	Vih_si	V _{DDIO} =1.2V to 3.6V	0.8 * V _{DDIO}			V
Output – low level for I2C	Vol_SDI	V _{DDIO} =1.62V, iol=3 mA			0.2 * V _{DDIO}	V
Output – low level for I2C	Vol_SDI_1.2	V _{DDIO} =1.20V, iol=3 mA			0.23 * V _{DDIO}	V
Output – low level	Vol_SDO	V _{DDIO} =1.62V, iol=1 mA			0.2 * V _{DDIO}	V
Output – low level	Vol_SDO_1.2	V _{DDIO} =1.20V, iol=1 mA			0.23 * V _{DDIO}	V
Output – high level	Voh	V _{DDIO} =1.62V, ioh=1 mA (SDO, SDI)	0.8 * V _{DDIO}			V
Output – high level	Voh_1.2	V _{DDIO} =1.2V, ioh=1 mA (SDO, SDI)	0.6 * V _{DDIO}			V
Pull-up resistor	Rpull	Internal pull-up resistance to V _{DDIO}	70	120	190	kΩ
I ² C bus load capacitor	Cb	On SDI and SCK			400	pF

5.4.2 I²C timings

For I²C timings, the following abbreviations are used:

- “S&F mode” = standard and fast mode
- “HS mode” = high speed mode
- Cb = bus capacitance on SDA line

All other naming refers to I²C specification 2.1 (January 2000).

The I²C timing diagram is shown in

Figure 12. The corresponding values are given in Table 27.

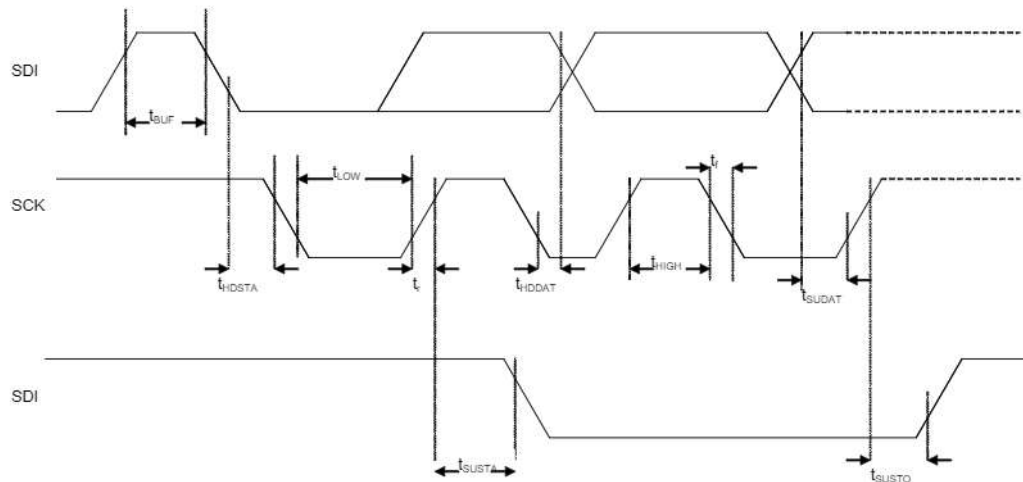

 Figure 12: I²C timing diagram

 Table 27: I²C timings

Parameter	Symbol	Condition	Min	Typ	Max	Units
SDI setup time	$t_{SU;DAT}$	S&F Mode	160			ns
		HS mode	30			ns
SDI hold time	$t_{HD;DAT}$	S&F Mode, $C_b \leq 100$ pF	80			ns
		S&F Mode, $C_b \leq 400$ pF	90			ns
		HS mode, $C_b \leq 100$ pF	18		115	ns
		HS mode, $C_b \leq 400$ pF	24		150	ns
SCK low pulse	t_{LOW}	HS mode, $C_b \leq 100$ pF $V_{DDIO} = 1.62$ V	160			ns
SCK low pulse	t_{LOW}	HS mode, $C_b \leq 100$ pF $V_{DDIO} = 1.2$ V	210			ns

The above-mentioned I²C specific timings correspond to the following internal added delays:

- Input delay between SDI and SCK inputs: SDI is more delayed than SCK by typically 100 ns in Standard and Fast Modes and by typically 20 ns in High Speed Mode.
- Output delay from SCK falling edge to SDI output propagation is typically 140 ns in Standard and Fast Modes and typically 70 ns in High Speed Mode.

5.4.3 SPI timings

The SPI timing diagram is in Figure 13, while the corresponding values are given in Table 28. All timings apply both to 4- and 3-wire SPI.

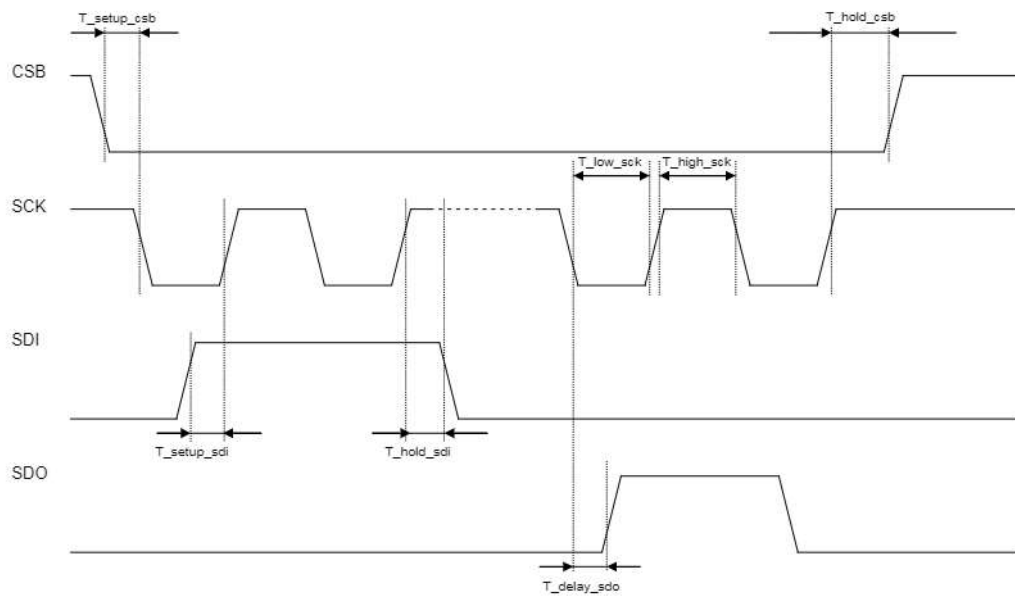


Figure 13: SPI timing diagram

Table 28: SPI timings

Parameter	Symbol	Condition	Min	Typ	Max	Units
SPI clock input frequency	F_spi		0		10	MHz
SCK low pulse	T_low_sck		20			ns
SCK high pulse	T_high_sck		20			ns
SDI setup time	T_setup_sdi		20			ns
SDI hold time	T_hold_sdi		20			ns
SDO output delay	T_delay_sdo	25pF load, V _{DDIO} =1.6V min			30	ns
SDO output delay	T_delay_sdo	25pF load, V _{DDIO} =1.2V min			40	ns
CSB setup time	T_setup_csb		20			ns
CSB hold time	T_hold_csb		20			ns

6. Pin-out and connection diagram

6.1 Pin-out

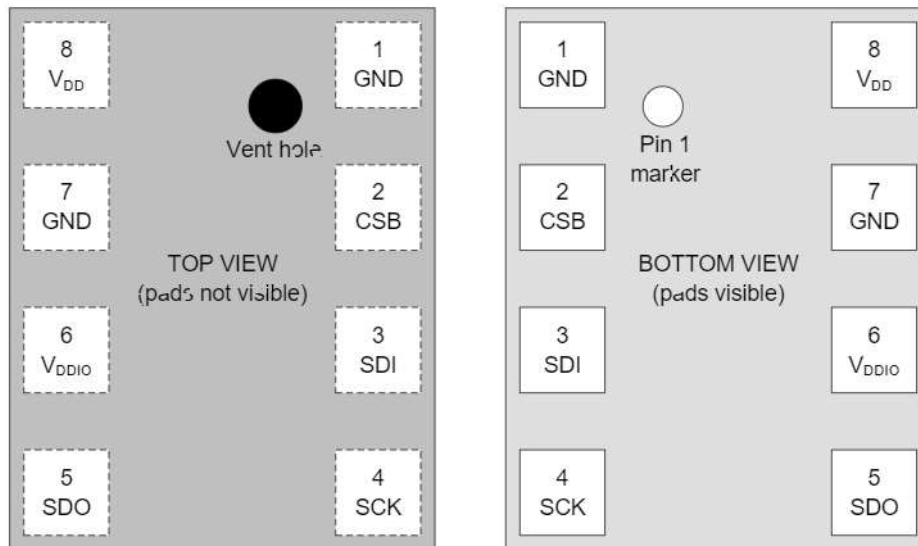


Figure 14: Pin-out top and bottom view

Table 29: Pin description

Pin	Name	I/O Type	Description	Connect to		
				SPI 4W	SPI 3W	I ² C
1	GND	Supply	Ground	GND		
2	CSB	In	Chip select	CSB	CSB	V _{DDIO}
3	SDI	In/Out	Serial data input	SDI	SDI/SDO	SDA
4	SCK	In	Serial clock input	SCK	SCK	SCL
5	SDO	In/Out	Serial data output	SDO	DNC	GND for default address
6	V _{DDIO}	Supply	Digital interface supply	V _{DDIO}		
7	GND	Supply	Ground	GND		
8	V _{DD}	Supply	Analog supply	V _{DD}		

6.2 Connection diagram 4-wire SPI

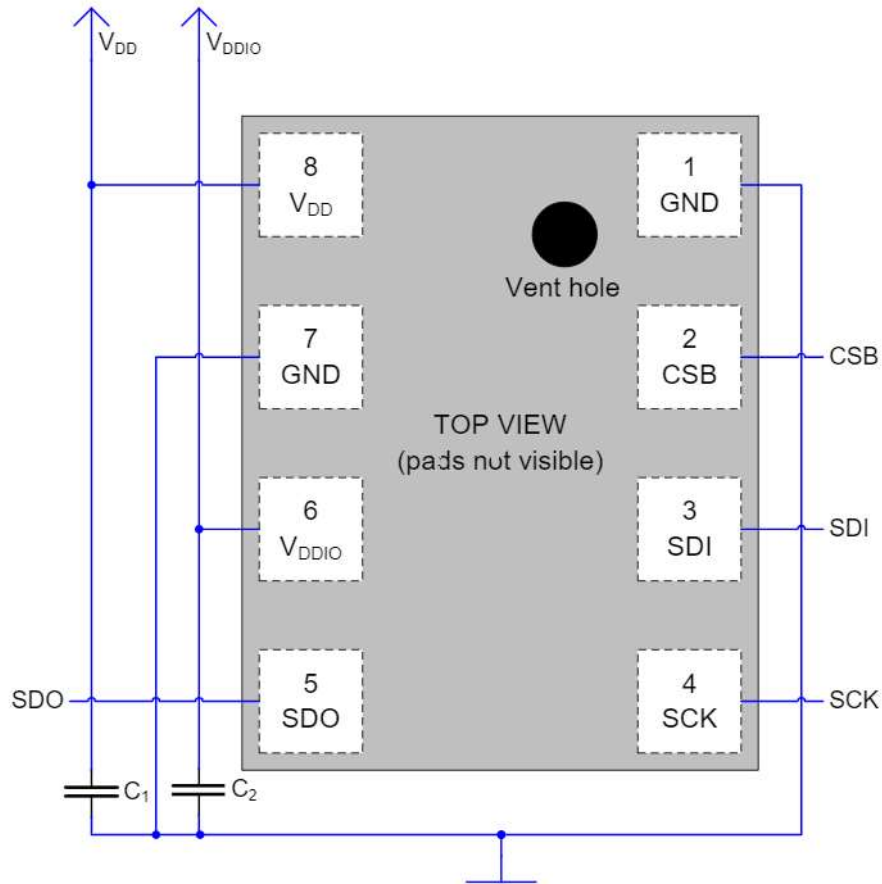


Figure 15: 4-wire SPI connection diagram (Pin1 marking indicated)

Note: the recommended value for C_1 , C_2 is 100 nF.

6.3 Connection diagram 3-wire SPI

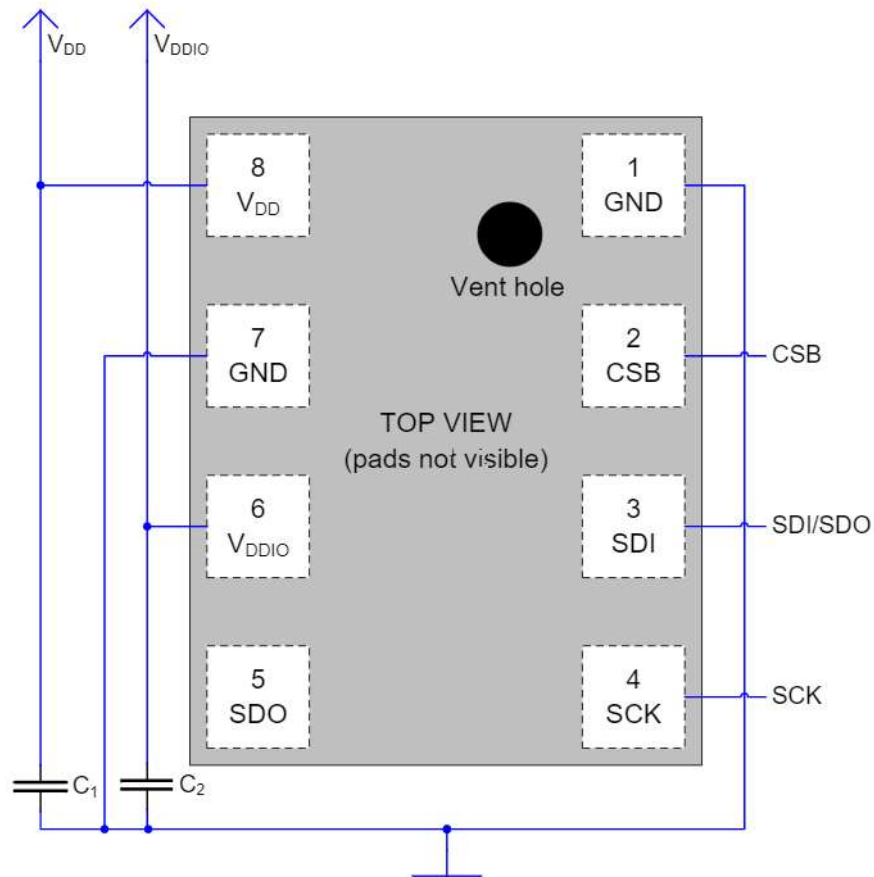


Figure 16: 3-wire SPI connection diagram (Pin1 marking indicated)

Note: the recommended value for C₁, C₂ is 100 nF.

6.4 Connection diagram I²C

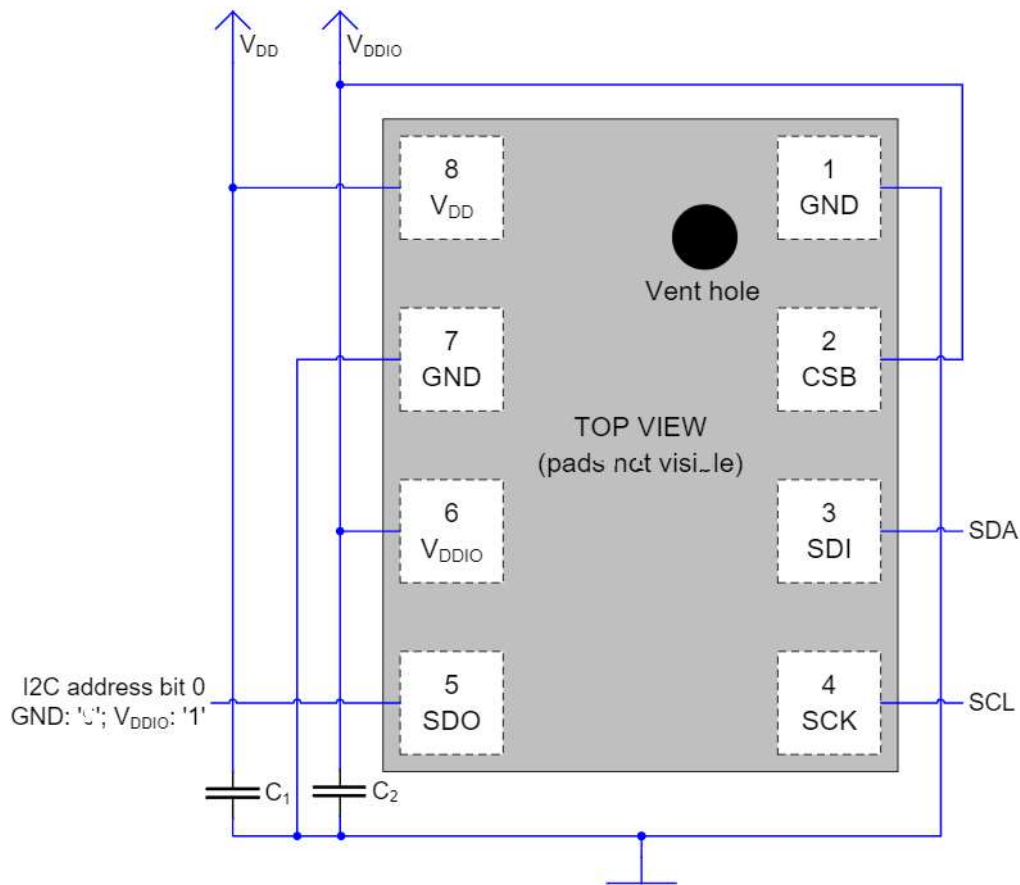


Figure 17: I²C connection diagram (Pin1 marking indicated)

Notes:

- The recommended value for C₁, C₂ is 100 nF.
- A direct connection between CSB and V_{DDIO} is recommended. If CSB is detected as low during startup, the interface will be locked into SPI mode. See chapter 5.1.

7. Package, reel and environment

7.1 Outline dimensions

The sensor housing is an 8-pin metal-lid LGA 2.0 × 2.5 × 0.95 mm³ package. Its dimensions are depicted in Figure 18.

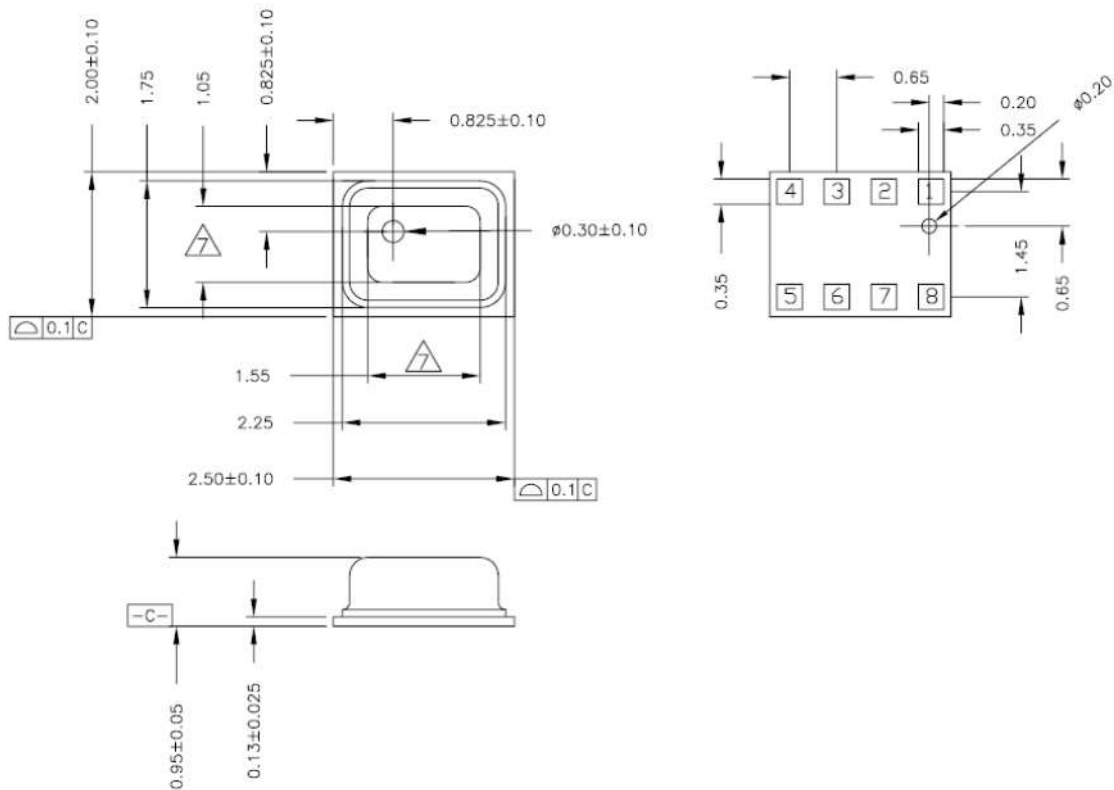


Figure 18: Package outline dimensions for top, bottom and side view

Note: General tolerances are $\pm 50 \mu\text{m}$ (linear) and $\pm 1^\circ$ (angular)

7.2 Landing pattern recommendation

For the design of the landing pattern, the following dimensioning is recommended:

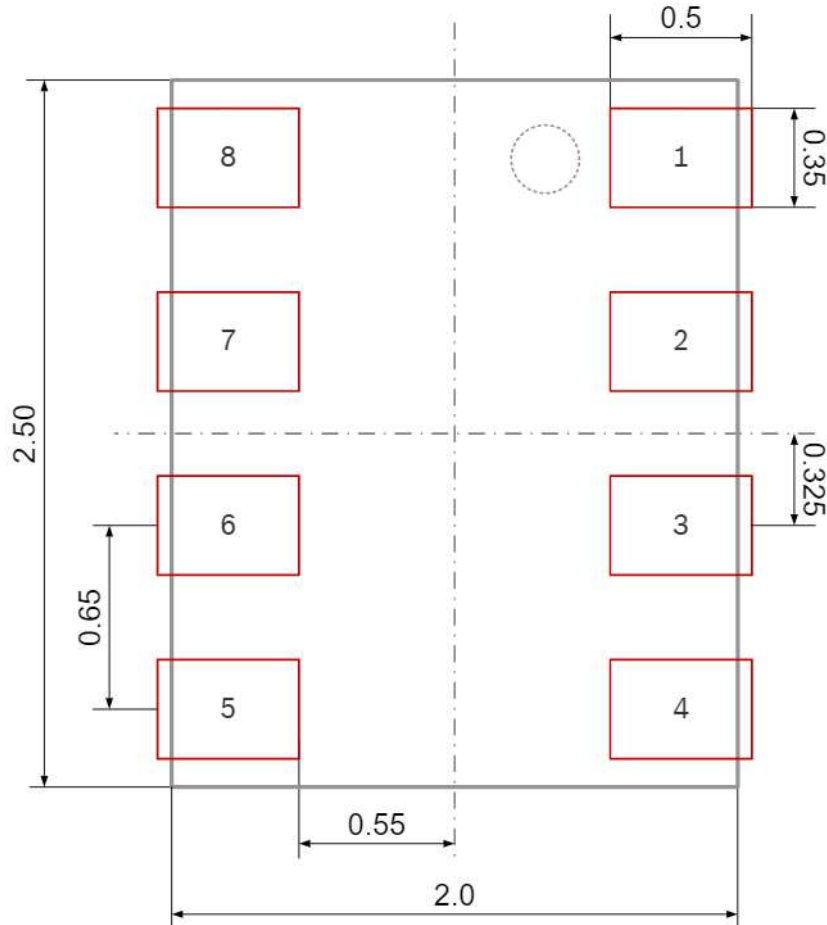


Figure 19: Recommended landing pattern (top view); dimensions are in mm


Note: red areas demark exposed PCB metal pads.

- In case of a solder mask defined (SMD) PCB process, the land dimensions should be defined by solder mask openings. The underlying metal pads are larger than these openings.
- In case of a non solder mask defined (NSMD) PCB process, the land dimensions should be defined in the metal layer. The mask openings are larger than the these metal pads.

7.3 Marking

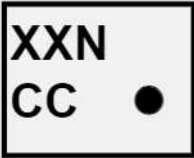
7.3.1 Mass production devices

Table 30: Marking of mass production samples

Labeling	Name	Symbol	Remark
	Lot counter	CCC	3 alphanumeric digits, variable to generate mass production trace-code
	Product number	T	1 alphanumeric digit, fixed to identify product type, T = "K" "K" is associated with the product BMP280
	Sub-con ID	L	1 alphanumeric digit, variable to identify subcon (L = "S" for underfill 0273300436)
	Orientation marker	●	Vent hole

7.3.2 Engineering samples

Table 31: Marking of engineering samples

Labeling	Name	Symbol	Remark
	Eng. Sample ID	N	1 alphanumeric digit, fixed to identify engineering sample, N = "*" or "e" or "E"
	Sample ID	XX	2 alphanumeric digits, variable to generate trace-code
	Counter ID	CC	2 alphanumeric digits, variable to generate trace-code
	Orientation marker	●	Vent hole

7.4 Soldering guidelines

The moisture sensitivity level of the BMP280 sensors corresponds to JEDEC Level 1, see also:

- IPC/JEDEC J-STD-020C “Joint Industry Standard: Moisture/Reflow Sensitivity Classification for non-hermetic Solid State Surface Mount Devices”
- IPC/JEDEC J-STD-033A “Joint Industry Standard: Handling, Packing, Shipping and Use of Moisture/Reflow Sensitive Surface Mount Devices”.

The sensor fulfils the lead-free soldering requirements of the above-mentioned IPC/JEDEC standard, i.e. reflow soldering with a peak temperature up to 260°C. The minimum height of the solder after reflow shall be at least 50µm. This is required for good mechanical decoupling between the sensor device and the printed circuit board (PCB).

Profile Feature	Pb-Free Assembly
Average Ramp-Up Rate ($T_{S_{max}}$ to T_p)	3° C/second max.
Preheat – Temperature Min ($T_{S_{min}}$) – Temperature Max ($T_{S_{max}}$) – Time ($t_{S_{min}}$ to $t_{S_{max}}$)	150 °C 200 °C 60-180 seconds
Time maintained above: – Temperature (T_L) – Time (t_L)	217 °C 60-150 seconds
Peak/Classification Temperature (T_p)	260 °C
Time within 5 °C of actual Peak Temperature (t_p)	20-40 seconds
Ramp-Down Rate	6 °C/second max.
Time 25 °C to Peak Temperature	8 minutes max.

Note 1: All temperatures refer to topside of the package, measured on the package body surface.

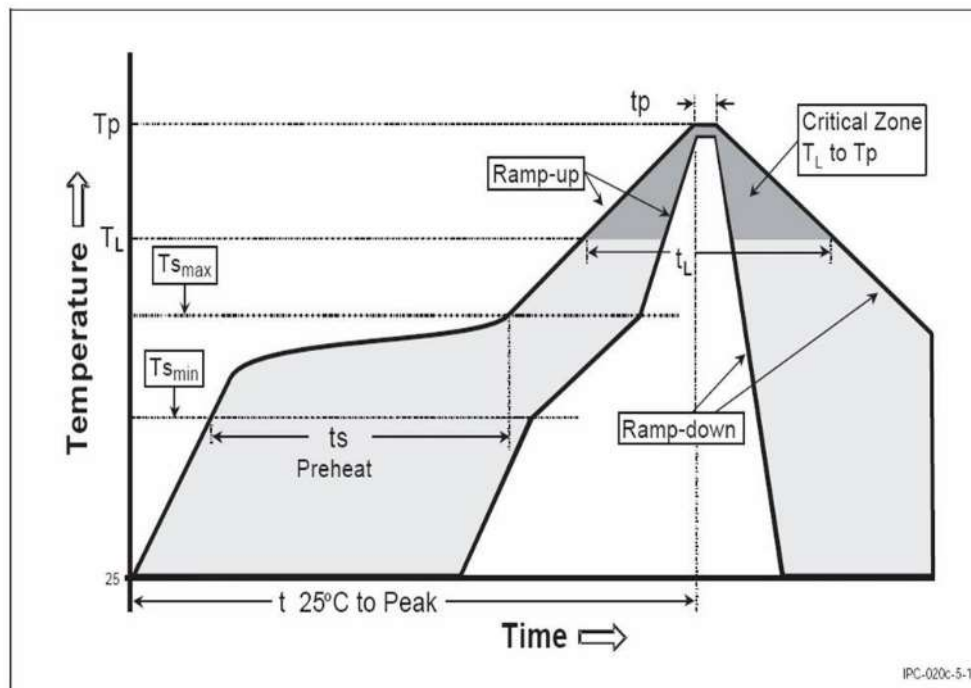


Figure 20: Soldering profile

7.5 Tape and reel specification

7.5.1 Dimensions

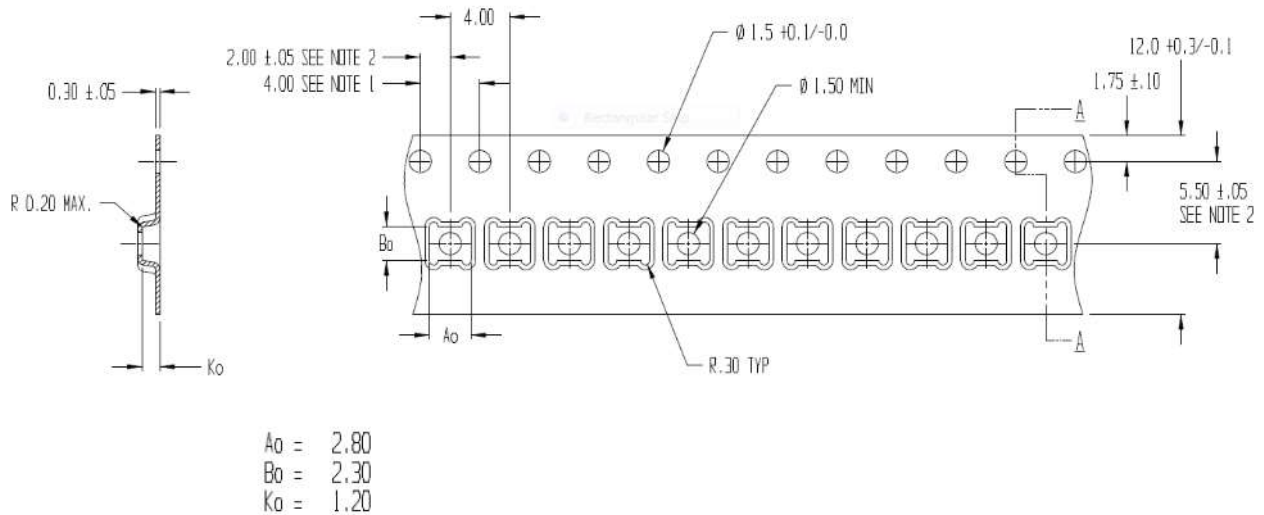


Figure 21: Tape and Reel dimensions

Quantity per reel: 10 kpcs.

7.5.2 Orientation within the reel

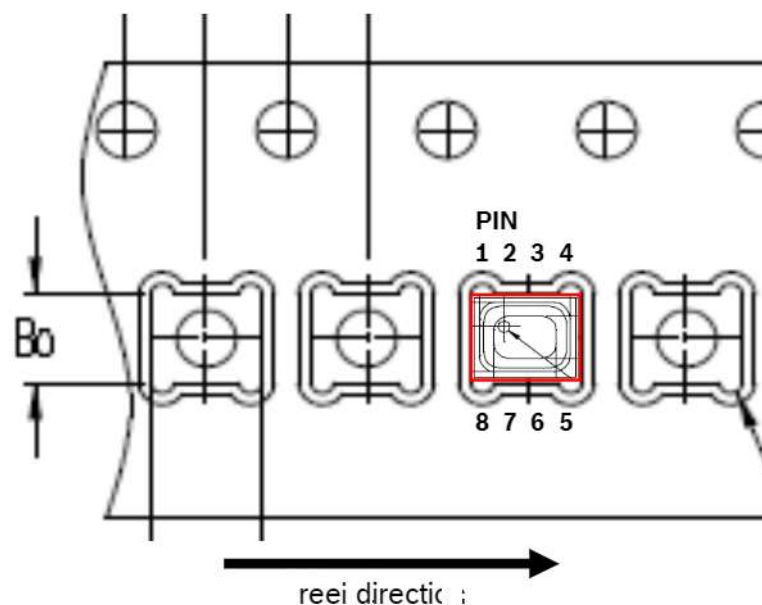


Figure 22: Orientation within tape

7.6 Mounting and assembly recommendations

In addition to “Handling, soldering & mounting instructions BMP280”, the following recommendations should be taken into consideration when mounting a pressure sensor on a printed-circuit board (PCB):

- The clearance above the metal lid shall be 0.1mm at minimum.
- For the device housing appropriate venting needs to be provided in case the ambient pressure shall be measured.
- Liquids shall not come into direct contact with the device.
- During operation the sensor chip is sensitive to light which can influence the accuracy of the measurement (photo-current of silicon). The position of the vent hole minimizes the light exposure of the sensor chip. Nevertheless, BST recommends to avoid the exposure of BMP280 to strong light sources.
- Soldering may not be done using vapor phase processes since the sensor might be damaged.

7.7 Environmental safety

7.7.1 RoHS

The BMP280 sensor meets the requirements of the EC restriction of hazardous substances (RoHS) directive, see also:

Directive 2011/65/EU of the European Parliament and of the Council of 8 June 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment

7.7.2 Halogen content

The BMP280 is halogen-free. For more details on the analysis results please contact your Bosch Sensortec representative.

7.7.3 Internal package structure

Within the scope of Bosch Sensortec’s ambition to improve its products and secure the mass product supply, Bosch Sensortec qualifies additional sources (e.g. 2nd source) for the LGA package of the BMP280.

While Bosch Sensortec took care that all of the technical packages parameters are described above are 100% identical for all sources, there can be differences in the chemical content and the internal structural between the different package sources.

However, as secured by the extensive product qualification process of Bosch Sensortec, this has no impact to the usage or to the quality of the BMP280 product.

8. Appendix 1: Computation formulae for 32 bit systems

8.1 Compensation formula in floating point

Please note that it is strongly advised to use the API available from Bosch Sensortec to perform readout and compensation. If this is not wanted, the code below can be applied at the user’s risk. Both pressure and temperature values are expected to be received in 20 bit format, positive, stored in a 32 bit signed integer.

The variable `t_fine` (signed 32 bit) carries a fine resolution temperature value over to the pressure compensation formula and could be implemented as a global variable.

The data type “BMP280_S32_t” should define a 32 bit signed integer variable type and could usually be defined as “long signed int”. The revision of the code is rev.1.1.

```
// Returns temperature in DegC, double precision. Output value of "51.23" equals 51.23 DegC.
// t_fine carries fine temperature as global value
BMP280_S32_t t_fine;
double bmp280_compensate_T_double(BMP280_S32_t adc_T)
{
    double var1, var2, T;
    var1 = (((double)adc_T)/256.0 - ((double)dig_T1)/16.0) * ((double)dig_T2;
    var2 = (((double)adc_T)/131072.0 - ((double)dig_T1)/8192.0) *
        (((double)adc_T)/131072.0 - ((double)dig_T1)/8192.0) * ((double)dig_T3);
    t_fine = (BMP280_S32_t)(var1 + var2);
    T = (var1 + var2) / 5120.0;
    return T;
}

// Returns pressure in Pa as double. Output value of "9386.2" equals 96386.2 Pa = 93.862 hPa
double bmp280_compensate_p_double(BMP280_S32_t adc_P)
{
    double var1, var2, p;
    var1 = ((double)t_fine/2.0) - 64000.0;
    var2 = var1 * var1 * ((double)dig_P6) / 32768.0;
    var2 = var2 + var1 * ((double)dig_P5) * 2.0;
    var2 = (var2/4.0)+(((double)dig_P4) * 65536.0);
    var1 = (((double)dig_P3) * var1 * var1 / 524288.0 + ((double)dig_P2) * var1) / 524288.0;
    var1 = (1.0 + var1 / 32768.0)*((double)dig_P1);
    if (var1 == 0.0)
    {
        return 0; // avoid exception caused by division by zero
    }
    p = 101325.0 - (double)adc_P;
    p = (p - (var2 / 4096.0)) * 6250.0 / var1;
    var1 = ((double)dig_P9) * p * p / 2147483648.0;
    var2 = p * ((double)dig_P8) / 32768.0;
    p = p + (var1 + var2 + ((double)dig_P7)) / 16.0;
    return p;
}
```

8.2 Compensation formula in 32 bit fixed point

Please note that it is strongly advised to use the API available from Bosch Sensortec to perform readout and compensation. If this is not wanted, the code below can be applied at the user’s risk. Both pressure and temperature values are expected to be received in 20 bit format, positive, stored in a 32 bit signed integer.

The variable `t_fine` (signed 32 bit) carries a fine resolution temperature value over to the pressure compensation formula and could be implemented as a global variable.

The data type “BMP280_S32_t” should define a 32 bit signed integer variable type and can usually be defined as “long signed int”.

The data type “BMP280_U32_t” should define a 32 bit unsigned integer variable type and can usually be defined as “long unsigned int”.

Compensating the pressure value with 32 bit integer has an accuracy of typically 1 Pa (1-sigma). At very high filter levels this adds a noticeable amount of noise to the output values and reduces their resolution.

```
// Returns temperature in DegC, resolution is 0.01 DegC. Output value of "5123" equals 51.23 DegC.
// t_fine carries fine temperature as global value
BMP280_S32_t t_fine;
BMP280_S32_t bmp280_compensate_T_int32(BMP280_S32_t adc_T)
{

```



```
BMP280_S32_t var1, var2, T;
var1 = (((adc_T>>3) - ((BMP280_S32_t)dig_T1<<1))) * ((BMP280_S32_t)dig_T2) >> 11;
var2 = (((((adc_T>>4) - ((BMP280_S32_t)dig_T1)) * ((adc_T>>4) - ((BMP280_S32_t)dig_T1))) >> 12) *
((BMP280_S32_t)dig_T3)) >> 14;
t_fine = var1 + var2;
T = (t_fine * 5 + 128) >> 8;
return T;
}

// Returns pressure in Pa as unsigned 32 bit integer. Output value of "96386" equals 96386 Pa = 963.86 hPa
BMP280_U32_t bmp280_compensate_P_int32(BMP280_S32_t adc_P)
{
    BMP280_S32_t var1, var2;
    BMP280_U32_t p;
    var1 = (((BMP280_S32_t)t_fine)>>1) - (BMP280_S32_t)64000;
    var2 = (((var1>>2) * (var1>>2)) >> 11) * ((BMP280_S32_t)dig_P6);
    var2 = var2 + ((var1*((BMP280_S32_t)dig_P5)<<1);
    var2 = (var2>>2)+(((BMP280_S32_t)dig_P4)<<16);
    var1 = (((dig_P3 * ((var1>>2) * (var1>>2)) >> 13) >> 3) + (((BMP280_S32_t)dig_P2) * var1)>>1)>>16;
    var1 = (((32768+var1))*((BMP280_S32_t)dig_P1)>>15);
    if (var1 == 0)
    {
        return 0; // avoid exception caused by division by zero
    }
    p = (((BMP280_U32_t)(((BMP280_S32_t)1048576)-adc_P) (var2>>12)))*3125;
    if (p < 0x80000000)
    {
        p - (p << 1) / ((BMP280_U32_t)var1);
    }
    else
    {
        p - (p / (BMP280_U32_t)var1) * 2;
    }
    var1 = (((BMP280_S32_t)dig_P9) * ((BMP280_S32_t)((p>>3) * (p>>3))>>13))>>12;
    var2 = (((BMP280_S32_t)(p>>2)) * ((BMP280_S32_t)dig_P8))>>13;
    p = (BMP280_U32_t)((BMP280_S32_t)p + ((var1 + var2 + dig_P7) >> 4));
    return p;
}
```

9. Legal disclaimer

9.1 Engineering samples

Engineering Samples are marked with an asterisk (*) or € (E). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

9.2 Product use

Bosch Sensortec products are developed for the consumer goods industry. They are not designed or approved for use in military applications, life-support appliances, safety-critical automotive applications and devices or systems where malfunctions of these products can reasonably be expected to result in personal injury. They may only be used within the parameters of this product data sheet.

The resale and/or use of products are at the Purchaser's own risk and the Purchaser's own responsibility.

The Purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The Purchaser accepts the responsibility to monitor the market for the purchased products, particularly with regard to product safety, and inform Bosch Sensortec without delay of any security relevant incidents.

9.3 Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

10. Document history and modification

Rev. No	Chapter	Description of modification/changes	Date
0.1		Document creation	2012-08-06
1.0	9.2	Change of product use	2013-11-26
	Table 2	Update of min/max data (only for restricted version)	
		Added comment on the sampling rate	
1.1	1, 3.3.1	Changed value for resolution, values for <i>osrs_p</i> settings changed	2014-02-10
	5.2	Changed sentence and added drawing	2014-02-18
	3.7	Added max values for current consumption	2014-05-08
1.11	4.5.3	Modified write in normal mode	2014-06-25
	5.2	Modified SDI/SCK ESD drawing	
1.12	1	Changed min/max values for standby current, only valid for 25 °C	2014-07-12
	Table 1	Pressure resolution 0.16Pa	2014-07-12
1.13	Page 2	New technical reference codes added	2014-11-12
	7.3	New details about laser marking added	
1.14	Table 6	Changed contents of table	2015-05-04
	Page 1	Removed TRC 0 273 300 354 & 0273 300 391	
	Page 44	Updated RoHS directive to 2011/65/EU effective 8 June 2011	2015-05-07
1.15	Page 2, 3	Modified target devices, applications	2015-10-15
1.16		Special customer-only	2016-01-28
1.17	7.3	Added all available TRC	2016-06-07
1.18	7.3	New marking	2016-11-02
1.19	Page 1	Update of technical reference code	2018-01-08

Bosch Sensortec GmbH
Gerhard-Kinzel-Strasse 3
72770 Leutlingen, Germany

contact@bosch-sensortec.com
www.bosch-sensortec.com

Modifications reserved | Printed in Germany
Specifications subject to change without notice
Document number: BST-BMP280-DS001-19
Revision_1.19_012018

LMx93-N, LM2903-N Low-Power, Low-Offset Voltage, Dual Comparators

1 Features

- Wide Supply
 - Voltage Range: 2.0 V to 36 V
 - Single or Dual Supplies: ± 1.0 V to ± 18 V
- Very Low Supply Current Drain (0.4 mA) — Independent of Supply Voltage
- Low Input Biasing Current: 25 nA
- Low Input Offset Current: ± 5 nA
- Maximum Offset voltage: ± 3 mV
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Low Output Saturation Voltage: 250 mV at 4 mA
- Output Voltage Compatible with TTL, DTL, ECL, MOS and CMOS logic systems
- Available in the 8-Bump (12 mil) DSBGA Package
- See AN-1112 ([SNVA009](#)) for DSBGA Considerations
- Advantages
 - High Precision Comparators
 - Reduced V_{OS} Drift Over Temperature
 - Eliminates Need for Dual Supplies
 - Allows Sensing Near Ground
 - Compatible with All Forms of Logic
 - Power Drain Suitable for Battery Operation

2 Applications

- Battery Powered Applications
- Industrial Applications

3 Description

The LM193-N series consists of two independent precision voltage comparators with an offset voltage specification as low as 2.0 mV max for two comparators which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. These comparators also have a unique characteristic in that the input common-mode voltage range includes ground, even though operated from a single power supply voltage.

Application areas include limit comparators, simple analog to digital converters; pulse, squarewave and time delay generators; wide range VCO; MOS clock timers; multivibrators and high voltage digital logic gates. The LM193-N series was designed to directly interface with TTL and CMOS. When operated from both plus and minus power supplies, the LM19-N series will directly interface with MOS logic where their low power drain is a distinct advantage over standard comparators.

The LM393 and LM2903 parts are available in TI's innovative thin DSBGA package with 8 (12 mil) large bumps.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM193-N	TO-99 (8)	9.08 mm x 9.08 mm
LM293-N		
LM393-N	SOIC (8)	4.90 mm x 3.91 mm
	DSBGA (8)	1.54 mm x 1.54 mm
LM2903-N	SOIC (8)	4.90 mm x 3.91 mm
	DSBGA (8)	1.54 mm x 1.54 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Simplified Schematic

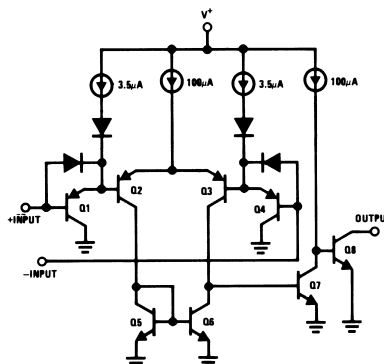


Table of Contents

1 Features	1	7.1 Overview	10
2 Applications	1	7.2 Functional Block Diagram	10
3 Description	1	7.3 Feature Description	10
4 Revision History	2	7.4 Device Functional Modes	10
5 Pin Configuration and Functions	3	8 Application and Implementation	11
6 Specifications	4	8.1 Application Information	11
6.1 Absolute Maximum Ratings	4	8.2 Typical Applications	11
6.2 ESD Ratings	4	9 Power Supply Recommendations	18
6.3 Recommended Operating Conditions	4	10 Layout	18
6.4 Thermal Information	5	10.1 Layout Guidelines	18
6.5 Electrical Characteristics: LM193A $V^+ = 5\text{ V}$, $T_A = 25^\circ\text{C}$	5	10.2 Layout Example	18
6.6 Electrical Characteristics: LM193A ($V^+ = 5\text{ V}$)	5	11 Device and Documentation Support	19
6.7 Electrical Characteristics: LMx93 and LM2903 $V^+ = 5\text{ V}$, $T_A = 25^\circ\text{C}$	6	11.1 Related Links	19
6.8 Electrical Characteristics: LMx93 and LM2903 ($V^+ = 5\text{ V}$) ⁽¹⁾	7	11.2 Receiving Notification of Documentation Updates	19
6.9 Typical Characteristics: LMx93 and LM193A	8	11.3 Trademarks	19
6.10 Typical Characteristics: LM2903	9	11.4 Electrostatic Discharge Caution	19
7 Detailed Description	10	11.5 Glossary	19
		12 Mechanical, Packaging, and Orderable Information	19

4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

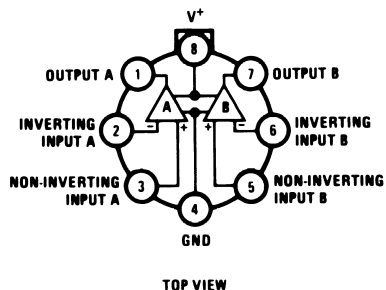
Changes from Revision F (December 2014) to Revision G	Page
• Added DSBGA packages inadvertently omitted from Device Info table during format conversion	1

Changes from Revision E (March 2013) to Revision F	Page
• Added <i>Pin Configuration and Functions</i> section, <i>ESD Ratings</i> table, <i>Feature Description</i> section, <i>Device Functional Modes</i> , <i>Application and Implementation</i> section, <i>Power Supply Recommendations</i> section, <i>Layout</i> section, <i>Device and Documentation Support</i> section, and <i>Mechanical, Packaging, and Orderable Information</i> section	1

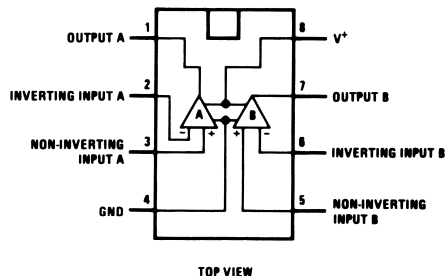
Changes from Revision D (March 2013) to Revision E	Page
• Changed layout of National Data Sheet to TI format	1

5 Pin Configuration and Functions

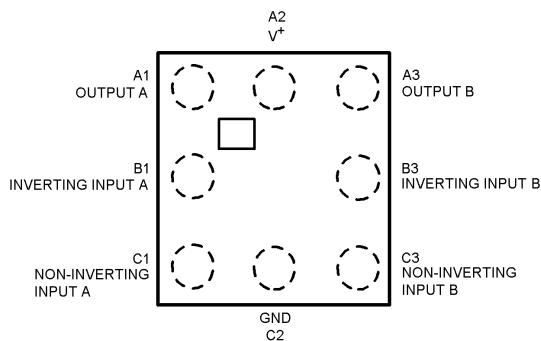
**LMC Package
8-Pin TO-99
Top View**



**P and D Package
8-Pin CDIP, PDIP, SOIC
Top View**



**YZR Package
8-Pin DSBGA
Top View**



Pin Functions

NAME	PIN NO.		I/O	DESCRIPTION
	PDIP/SOIC/ TO-99	DSBGA		
OUTA	1	A1	O	Output, Channel A
-INA	2	B1	I	Inverting Input, Channel A
+INA	3	C1	I	Noninverting Input, Channel A
GND	4	C2	P	Ground
+INB	5	C3	I	Noninverting Input, Channel B
-INB	6	B3	I	Inverting Input, Channel B
OUTB	7	A3	O	Output, Channel B
V+	8	A2	P	Positive power supply

6 Specifications

6.1 Absolute Maximum Ratings

 over operating free-air temperature range (unless otherwise noted)⁽¹⁾⁽²⁾⁽³⁾

		MIN	MAX	UNIT
Differential Input Voltage ⁽⁴⁾			36	V
Input Voltage		-0.3	36	V
Input Current ($V_{IN} < -0.3$ V) ⁽⁵⁾			50	mA
Power Dissipation ⁽⁶⁾	PDIP		780	mW
	TO-99		660	mW
	SOIC		510	mW
	DSBGA		568	mW
Output Short-Circuit to Ground ⁽⁷⁾			Continu ous	
Lead Temperature (Soldering, 10 seconds)			260	°C
Soldering Information	PDIP Package Soldering (10 seconds)		260	°C
	SOIC Package	Vapor Phase (60 seconds)	215	°C
		Infrared (15 seconds)	220	°C
Storage temperature, T_{stg}		-65	150	°C

- (1) *Absolute Maximum Ratings* indicate limits beyond which damage may occur. *Recommended Operating Conditions* indicate conditions for which the device is intended to be functional, but specific performance is not guaranteed. For guaranteed specifications and test conditions, see the Electrical Characteristics.
- (2) Refer to RETS193AX for LM193AH military specifications and to RETS193X for LM193H military specifications.
- (3) If Military/Aerospace specified devices are required, please contact the TI Sales Office/Distributors for availability and specifications.
- (4) Positive excursions of input voltage may exceed the power supply level. As long as the other voltage remains within the common-mode range, the comparator will provide a proper output state. The low input voltage state must not be less than -0.3 V (or 0.3 V below the magnitude of the negative power supply, if used).
- (5) This input current will only exist when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistors becoming forward biased and thereby acting as input diode clamps. In addition to this diode action, there is also lateral NPN parasitic transistor action on the IC chip. This transistor action can cause the output voltages of the comparators to go to the V^+ voltage level (or to ground for a large overdrive) for the time duration that an input is driven negative. This is not destructive and normal output states will re-establish when the input voltage, which was negative, again returns to a value greater than -0.3 V.
- (6) For operating at high temperatures, the LM393 and LM2903 must be derated based on a 125°C maximum junction temperature and a thermal resistance of $170^{\circ}\text{C}/\text{W}$ which applies for the device soldered in a printed circuit board, operating in a still air ambient. The LM193/LM193A/LM293 must be derated based on a 150°C maximum junction temperature. The low bias dissipation and the "ON-OFF" characteristic of the outputs keeps the chip dissipation very small ($P_D \leq 100$ mW), provided the output transistors are allowed to saturate.
- (7) Short circuits from the output to V^+ can cause excessive heating and eventual destruction. When considering short circuits to ground, the maximum output current is approximately 20 mA independent of the magnitude of V^+ .

6.2 ESD Ratings

		VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾	± 1300 V

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

	MIN	NOM	MAX	UNIT
Supply Voltage (V^+) - Single Supply	2.0		36	V
Supply Voltage (V^+) - Dual Supply	± 1.0		± 18	V
Operating Input Voltage on (V_{IN} pin)	0		(V^+) -1.5V	V
Operating junction temperature, T_J : LM193/LM193A	-55		125	°C
Operating junction temperature, T_J : LM2903	-40		85	°C
Operating junction temperature, T_J : LM293	-25		85	°C
Operating junction temperature, T_J : LM393	0		70	°C

6.4 Thermal Information

THERMAL METRIC ⁽¹⁾	LMx93	UNIT
	TO-99	
	8 PINS	
R _{θJA} Junction-to-ambient thermal resistance	170	°C/W

(1) For more information about traditional and new thermal metrics, see the *IC Package Thermal Metrics* application report, [SPRA953](#).

6.5 Electrical Characteristics: LM193A V⁺ = 5 V, T_A = 25°C

Unless otherwise stated.

PARAMETER	TEST CONDITIONS	LM193A			UNIT		
		MIN	TYP	MAX			
Input Offset Voltage	See ⁽¹⁾ .		1.0	2.0	mV		
Input Bias Current	I _{IN(+)} or I _{IN(-)} with Output In Linear Range, V _{CM} = 0 V ⁽²⁾		25	100	nA		
Input Offset Current	I _{IN(+)} - I _{IN(-)} V _{CM} = 0 V		3.0	25	nA		
Input Common Mode Voltage Range	V ₊ = 30 V ⁽³⁾	0		V ⁺ - 1.5	V		
Supply Current	R _L = ∞		V ⁺ = 5 V		0.4	1	mA
			V ⁺ = 36 V		1	2.5	mA
Voltage Gain	R _L ≥ 15 kΩ, V ⁺ = 15 V V _O = 1 V to 11 V	50	200			V/mV	
Large Signal Response Time	V _{IN} = TTL Logic Swing, V _{REF} = 1.4 V V _{RL} = 5V, R _L = 5.1 kΩ		300			ns	
Response Time	V _{RL} = 5V, R _L = 5.1 kΩ ⁽⁴⁾		1.3			μs	
Output Sink Current	V _{IN(-)} = 1V, V _{IN(+)} = 0, V _O ≈ 1.5 V	6.0	16			mA	
Saturation Voltage	V _{IN(-)} = 1V, V _{IN(+)} = 0, I _{SINK} ≤ 4 mA		250	400		mV	
Output Leakage Current	V _{IN(-)} = 0, V _{IN(+)} = 1V, V _O = 5 V		0.1			nA	

- (1) At output switch point, V_O ≈ 1.4V, R_S = 0 Ω with V⁺ from 5V to 30V; and over the full input common-mode range (0V to V⁺ - 1.5V), at 25°C.
- (2) The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the reference or input lines.
- (3) The input common-mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3V. The upper end of the common-mode voltage range is V⁺ - 1.5 V at 25°C, but either or both inputs can go to 36 V without damage, independent of the magnitude of V⁺.
- (4) The response time specified is for a 100 mV input step with 5 mV overdrive. For larger overdrive signals 300 ns can be obtained, see [LMx93 and LM193A Typical Characteristics](#).

6.6 Electrical Characteristics: LM193A (V₊ = 5 V)⁽¹⁾

PARAMETER	TEST CONDITIONS	LM193A			UNIT
		MIN	TYP	MAX	
Input Offset Voltage	See ⁽²⁾			4.0	mV
Input Offset Current	I _{IN(+)} - I _{IN(-)} ; V _{CM} = 0 V			100	nA
Input Bias Current	I _{IN(+)} or I _{IN(-)} with Output in Linear Range, V _{CM} = 0 V ⁽³⁾			300	nA
Input Common Mode Voltage Range	V ⁺ = 30 V ⁽⁴⁾	0		V ⁺ - 2.0	V
Saturation Voltage	V _{IN(-)} = 1V, V _{IN(+)} = 0, I _{SINK} ≤ 4 mA			700	mV
Output Leakage Current	V _{IN(-)} = 0, V _{IN(+)} = 1V, V _O = 30 V			1.0	μA
Differential Input Voltage	Keep All V _{IN} 's ≥ 0 V (or V ⁻ , if Used), ⁽⁵⁾			36	V

- (1) These specifications are limited to -55°C ≤ T_A ≤ 125°C, for the LM193/LM193A. With the LM293 all temperature specifications are limited to -25°C ≤ T_A ≤ 85°C and the LM393 temperature specifications are limited to 0°C ≤ T_A ≤ 70°C. The LM2903 is limited to -40°C ≤ T_A ≤ 85°C.
- (2) At output switch point, V_O ≈ 1.4V, R_S = 0 Ω with V⁺ from 5V to 30V; and over the full input common-mode range (0V to V⁺ - 1.5V), at 25°C.
- (3) The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the reference or input lines.
- (4) The input common-mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3V. The upper end of the common-mode voltage range is V⁺ - 1.5 V at 25°C, but either or both inputs can go to 36 V without damage, independent of the magnitude of V⁺.
- (5) Positive excursions of input voltage may exceed the power supply level. As long as the other voltage remains within the common-mode range, the comparator will provide a proper output state. The low input voltage state must not be less than -0.3V (or 0.3V below the magnitude of the negative power supply, if used).

6.7 Electrical Characteristics: LMx93 and LM2903 $V^+ = 5\text{ V}$, $T_A = 25^\circ\text{C}$

Unless otherwise stated.

PARAMETER	TEST CONDITIONS	LM193-N			LM293-N, LM393-N			LM2903-N			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MI N	TYP	MAX	
Input Offset Voltage	See ⁽¹⁾		1.0	5.0		1.0	5.0		2.0	7.0	mV
Input Bias Current	$I_{IN(+)}$ or $I_{IN(-)}$ with Output In Linear Range, $V_{CM} = 0\text{ V}$ ⁽²⁾		25	100		25	250		25	250	nA
Input Offset Current	$I_{IN(+)} - I_{IN(-)}$ $V_{CM} = 0\text{ V}$		3.0	25		5.0	50		5.0	50	nA
Input Common Mode Voltage Range	$V^+ = 30\text{ V}$ ⁽³⁾	0		$V^+ - 1.5$	0		$V^+ - 1.5$	0		$V^+ - 1.5$	V
Supply Current	$R_L = \infty$		$V^+ = 5\text{ V}$			$V^+ = 5\text{ V}$			$V^+ = 5\text{ V}$		
			$V^+ = 36\text{ V}$			$V^+ = 36\text{ V}$			$V^+ = 36\text{ V}$		
Voltage Gain	$R_L \geq 15\text{ k}\Omega$, $V^+ = 15\text{ V}$ $V_O = 1\text{ V}$ to 11 V	50	200		50	200		25	100		V/mV
Large Signal Response Time	$V_{IN} = \text{TTL Logic Swing}$, $V_{REF} = 1.4\text{ V}$ $V_{RL} = 5\text{ V}$, $R_L = 5.1\text{ k}\Omega$		300			300			300		ns
Response Time	$V_{RL} = 5\text{ V}$, $R_L = 5.1\text{ k}\Omega$ ⁽⁴⁾		1.3			1.3			1.5		μs
Output Sink Current	$V_{IN(-)} = 1\text{ V}$, $V_{IN(+)} = 0$, $V_O \leq 1.5\text{ V}$	6.0	16		6.0	16		6.0	16		mA
Saturation Voltage	$V_{IN(-)} = 1\text{ V}$, $V_{IN(+)} = 0$, $I_{SINK} \leq 4\text{ mA}$		250	400		250	400		250	400	mV
Output Leakage Current	$V_{IN(-)} = 0$, $V_{IN(+)} = 1\text{ V}$, $V_O = 5\text{ V}$		0.1			0.1			0.1		nA

- (1) At output switch point, $V_O = 1.4\text{ V}$, $R_S = 0\ \Omega$ with V^+ from 5 V to 30 V ; and over the full input common-mode range (0 V to $V^+ - 1.5\text{ V}$), at 25°C .
- (2) The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the reference or input lines.
- (3) The input common-mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3 V . The upper end of the common-mode voltage range is $V^+ - 1.5\text{ V}$ at 25°C , but either or both inputs can go to 36 V without damage, independent of the magnitude of V^+ .
- (4) The response time specified is for a 100 mV input step with 5 mV overdrive. For larger overdrive signals 300 ns can be obtained, see [LMx93 and LM193A Typical Characteristics](#).

6.8 Electrical Characteristics: LMx93 and LM2903 ($V_+ = 5\text{ V}$)⁽¹⁾

PARAMETER	TEST CONDITIONS	LM193-N			LM293-N, LM393-N			LM290-N			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
Input Offset Voltage	See ⁽²⁾			9			9		9	15	mV
Input Offset Current	$I_{IN(+)} - I_{IN(-)}$, $V_{CM}=0\text{ V}$			100			150		50	200	nA
Input Bias Current	$I_{IN(+)}$ or $I_{IN(-)}$ with Output in Linear Range, $V_{CM}=0\text{ V}$ ⁽³⁾			300			400		200	500	nA
Input Common Mode Voltage Range	$V^+ = 30\text{ V}$ ⁽⁴⁾	0		$V^+ - 2.0$	0		$V^+ - 2.0$	0		$V^+ - 2.0$	V
Saturation Voltage	$V_{IN(-)} = 1\text{ V}$, $V_{IN(+)} = 0$, $I_{SINK} \leq 4\text{ mA}$			700			700		400	700	mV
Output Leakage Current	$V_{IN(-)} = 0$, $V_{IN(+)} = 1\text{ V}$, $V_O = 30\text{ V}$			1.0			1.0			1.0	μA
Differential Input Voltage	Keep All V_{IN} 's $\geq 0\text{ V}$ (or V^- , if Used), ⁽⁵⁾			36			36			36	V

- (1) These specifications are limited to $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$, for the LM193/LM193A. With the LM293 all temperature specifications are limited to $-25^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$ and the LM393 temperature specifications are limited to $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$. The LM2903 is limited to $-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$.
- (2) At output switch point, $V_O = 1.4\text{ V}$, $R_S = 0\ \Omega$ with V^+ from 5V to 30V; and over the full input common-mode range (0V to $V^+ - 1.5\text{ V}$), at 25°C .
- (3) The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the reference or input lines.
- (4) The input common-mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3V. The upper end of the common-mode voltage range is $V^+ - 1.5\text{ V}$ at 25°C , but either or both inputs can go to 36 V without damage, independent of the magnitude of V^+ .
- (5) Positive excursions of input voltage may exceed the power supply level. As long as the other voltage remains within the common-mode range, the comparator will provide a proper output state. The low input voltage state must not be less than -0.3 V (or 0.3V below the magnitude of the negative power supply, if used).

6.9 Typical Characteristics: LMx93 and LM193A

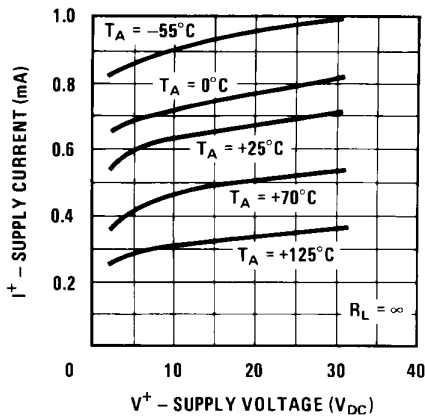


Figure 1. Supply Current

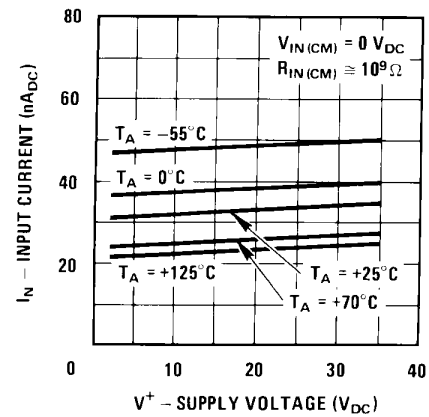


Figure 2. Input Current

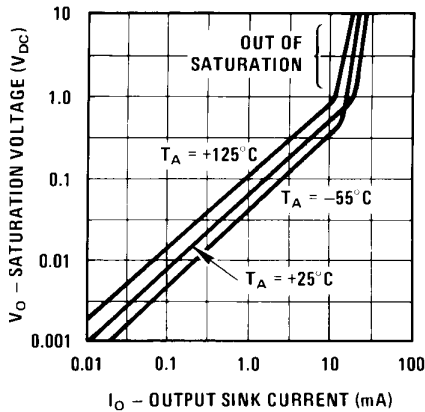


Figure 3. Output Saturation Voltage

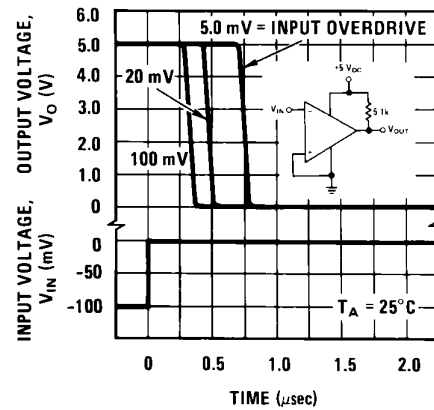


Figure 4. Response Time for Various Input Overdrives—Negative Transition

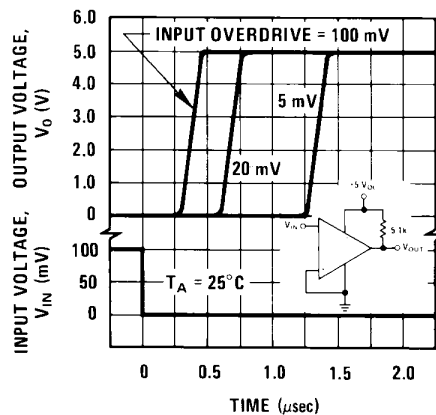


Figure 5. Response Time for Various Input Overdrives—Positive Transition

6.10 Typical Characteristics: LM2903

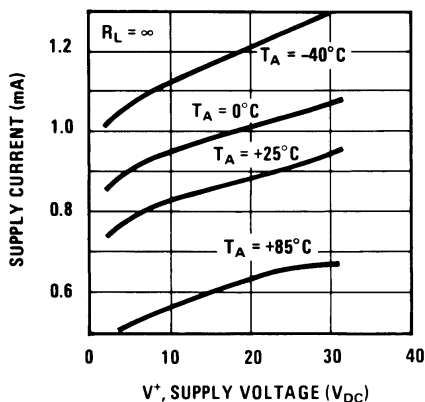


Figure 6. Supply Current

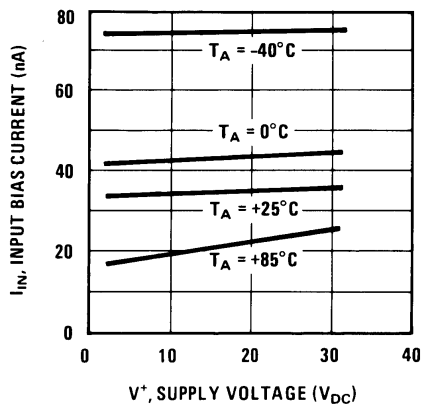


Figure 7. Input Current

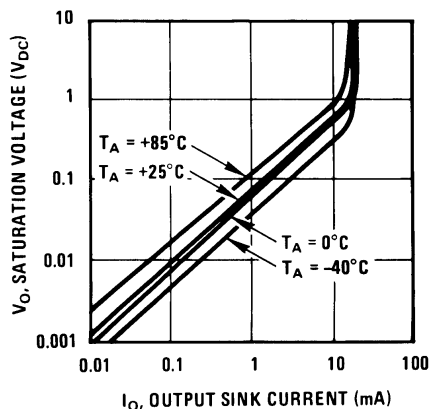


Figure 8. Output Saturation Voltage

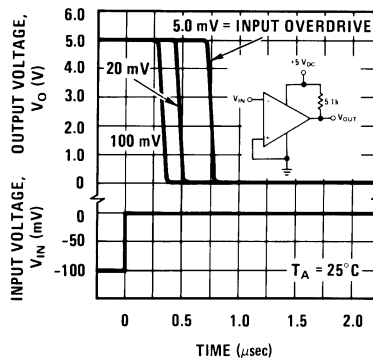


Figure 9. Response Time for Various Input Overdrives—Negative Transition

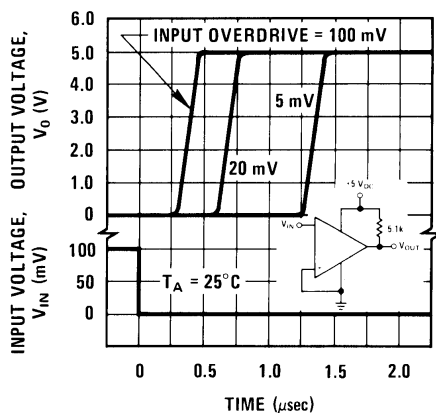


Figure 10. Response Time for Various Input Overdrives—Positive Transition

7 Detailed Description

7.1 Overview

The LM139 provides two independently functioning, high-precision, low V_{OS} drift, low input bias current comparators in a single package. The low power consumption of 0.4 mA at 5 V and the 2.0 V supply operation makes the LM139 suitable for battery powered applications.

7.2 Functional Block Diagram

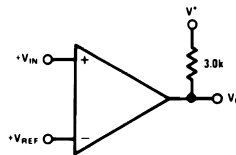


Figure 11. Basic Comparator

7.3 Feature Description

The input bias current of 25 nA enables the LM193 to use even very high impedance nodes as inputs. The differential voltage input range equals the supply voltage range.

The LM193 can be operated with a single supply, where $V+$ can be from 2.0 V to 36 V, or in a dual supply voltage configuration, where GND pin is used as a $V-$ supply. The supply current draws only 0.4 mA for both comparators.

The output of each comparator in the LM193 is the open collector of a grounded-emitter NPN output transistor which can typically draw up to 16 mA.

7.4 Device Functional Modes

A basic comparator circuit is used for converting analog signals to a digital output. The output is HIGH when the voltage on the non-inverting (+IN) input is greater than the inverting (-IN) input. The output is LOW when the voltage on the non-inverting (+IN) input is less than the inverting (-IN) input. The inverting input (-IN) is also commonly referred to as the "reference" or "VREF" input. All pins of any unused comparators should be tied to the negative supply.

8 Application and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

8.1 Application Information

The LM193 series are high gain, wide bandwidth devices which, like most comparators, can easily oscillate if the output lead is inadvertently allowed to capacitively couple to the inputs via stray capacitance. This shows up only during the output voltage transition intervals as the comparator change states. Power supply bypassing is not required to solve this problem. Standard PC board layout is helpful as it reduces stray input-output coupling. Reducing the input resistors to $< 10\text{ k}\Omega$ reduces the feedback signal levels and finally, adding even a small amount (1.0 to 10 mV) of positive feedback (hysteresis) causes such a rapid transition that oscillations due to stray feedback are not possible. Simply socketing the IC and attaching resistors to the pins will cause input-output oscillations during the small transition intervals unless hysteresis is used. If the input signal is a pulse waveform, with relatively fast rise and fall times, hysteresis is not required.

All input pins of any unused comparators should be tied to the negative supply.

The bias network of the LM193 series establishes a drain current which is independent of the magnitude of the power supply voltage over the range of from 2.0 V_{DC} to 30 V_{DC} .

The differential input voltage may be larger than V^+ without damaging the device [Typical Applications](#). Protection should be provided to prevent the input voltages from going negative more than $-0.3\text{ V}_{\text{DC}}$ (at 25°C). An input clamp diode can be used as shown in [Typical Applications](#).

The output of the LM193 series is the uncommitted collector of a grounded-emitter NPN output transistor. Many collectors can be tied together to provide an output OR'ing function. An output pullup resistor can be connected to any available power supply voltage within the permitted supply voltage range and there is no restriction on this voltage due to the magnitude of the voltage which is applied to the V^+ terminal of the LM193 package. The output can also be used as a simple SPST switch to ground (when a pullup resistor is not used). The amount of current which the output device can sink is limited by the drive available (which is independent of V^+) and the β of this device. When the maximum current limit is reached (approximately 16 mA), the output transistor will come out of saturation and the output voltage will rise very rapidly. The output saturation voltage is limited by the approximately $60\ \Omega\ r_{\text{SAT}}$ of the output transistor. The low offset voltage of the output transistor (1.0 mV) allows the output to clamp essentially to ground level for small load currents.

8.2 Typical Applications

8.2.1 Basic Comparator

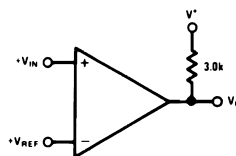


Figure 12. Basic Comparator

8.2.1.1 Design Requirements

The basic usage of a comparator is to indicate when a specific analog signal has exceeded some predefined threshold. In this application, the negative input (IN-) is tied to a reference voltage, and the positive input (IN+) is connected to the input signal. The output is pulled up with a resistor to the logic supply voltage, V^+ with a pullup resistor.

For an example application, the supply voltage is 5V. The input signal varies between 1 V and 3 V, and we want to know when the input exceeds $2.5\text{ V} \pm 1\%$. The supply current draw should not exceed 1 mA.

Typical Applications (continued)

8.2.1.2 Detailed Design Procedure

First, we determine the biasing for the 2.5-V reference. With the 5-V supply voltage, we would use a voltage divider consisting of one resistor from the supply to IN⁻ and an second resistor from IN⁻. The 25 nA of input current bias should be < 1% of the bias current for V_{ref}. With a 100-kΩ resistor from IN⁻ to V₊ and an additional 100-kΩ resistor from IN⁻ to ground, there would be 25 μA of current through the two resistors. The 3-kΩ pullup shown will need $5\text{ V}/3\text{ k}\Omega \rightarrow 1.67\text{ mA}$, which exceeds our current budget.

With the 400-μA supply current and 25 μA of V_{REF} bias current, there is 575 μA remaining for output pullup resistor; with 5-V supply, we need a pullup larger than 8.7 kΩ. A 10-kΩ pullup is a value that is commonly available and can be used here.

8.2.1.3 Application Curve

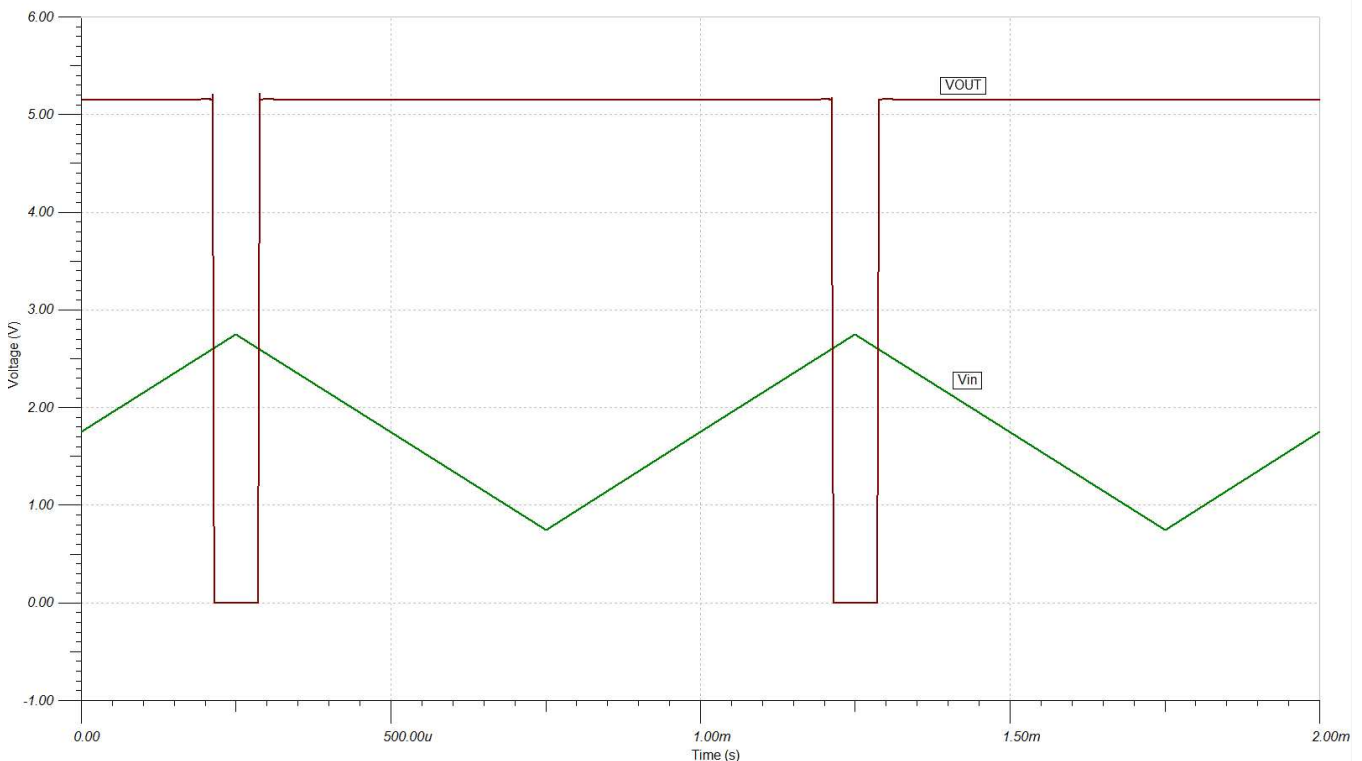
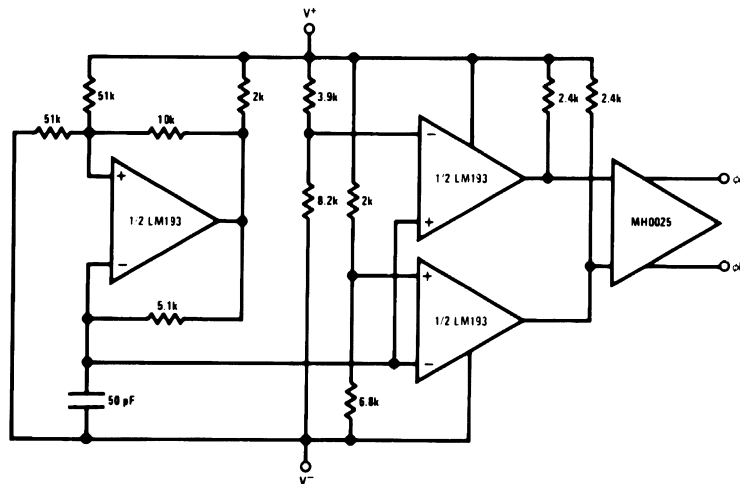


Figure 13. Basic Comparator Response

Typical Applications (continued)

8.2.2 System Examples

8.2.2.1 Split-Supply Application



($V_+ = -15 V_{DC}$ and $V_- = 15 V_{DC}$)

Figure 14. MOS Clock Driver

8.2.2.2 $V_+ = 5.0 V_{DC}$ Application Circuits

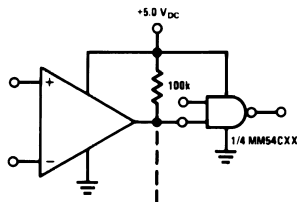


Figure 15. Driving CMOS

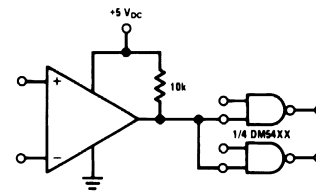


Figure 16. Driving TTL

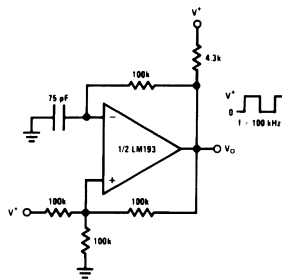


Figure 17. Squarewave Oscillator

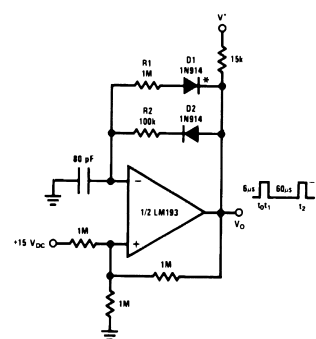


Figure 18. Pulse Generator

* For large ratios of R_1/R_2 ,
D1 can be omitted.

Typical Applications (continued)

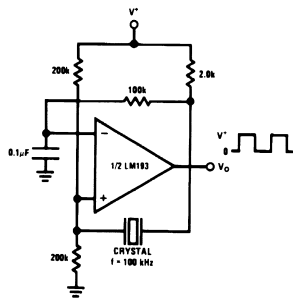
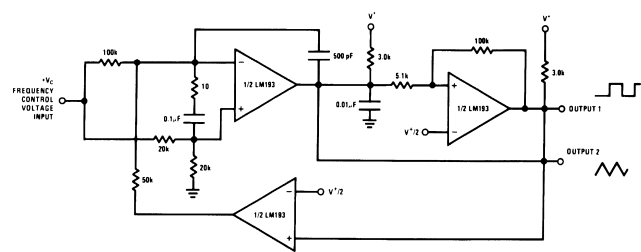


Figure 19. Crystal Controlled Oscillator



$V^* = +30 V_{DC}$
 $+250 mV_{DC} \leq V_C \leq +50 V_{DC}$
 $700Hz \leq f_o \leq 100kHz$

Figure 20. Two-Decade High Frequency VCO

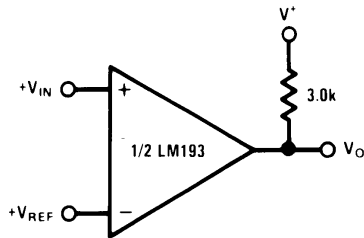


Figure 21. Basic Comparator

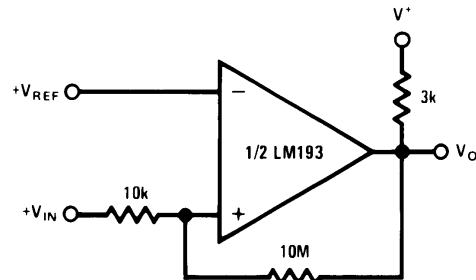


Figure 22. Non-Inverting Comparator With Hysteresis

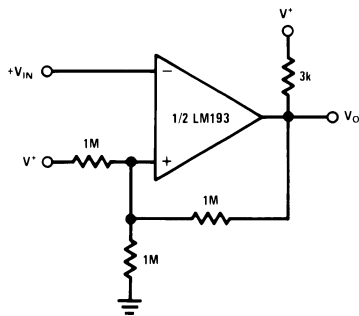
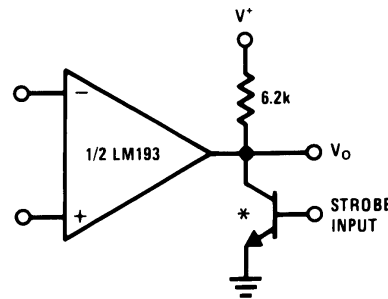


Figure 23. Inverting Comparator With Hysteresis



* OR LOGIC GATE WITHOUT PULL-UP RESISTOR

Figure 24. Output Strobing

Typical Applications (continued)

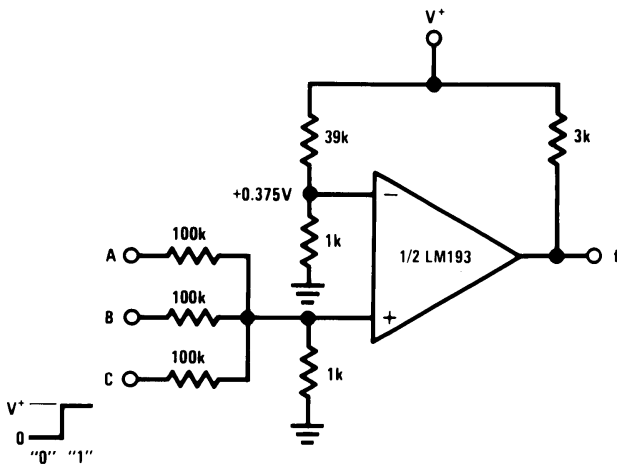


Figure 25. And Gate

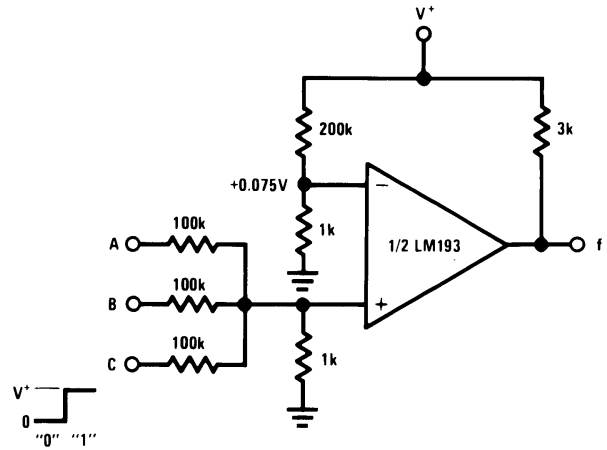


Figure 26. Or Gate

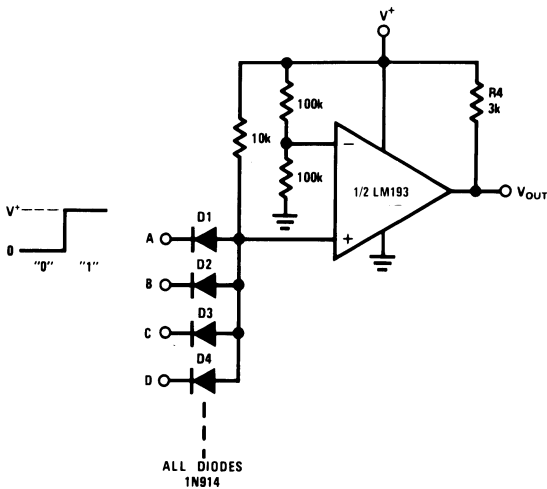


Figure 27. Large Fan-In and Gate

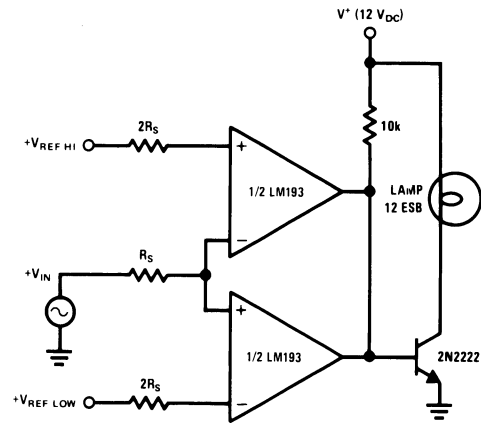


Figure 28. Limit Comparator

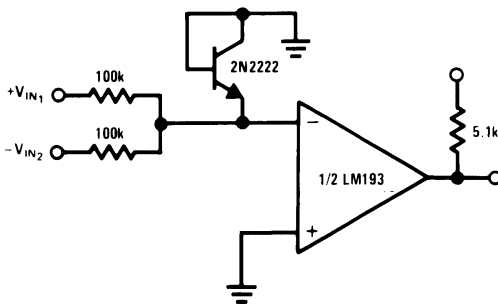


Figure 29. Comparing Input Voltages of Opposite Polarity

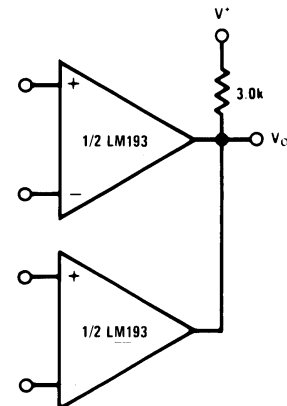


Figure 30. Oring the Outputs

Typical Applications (continued)

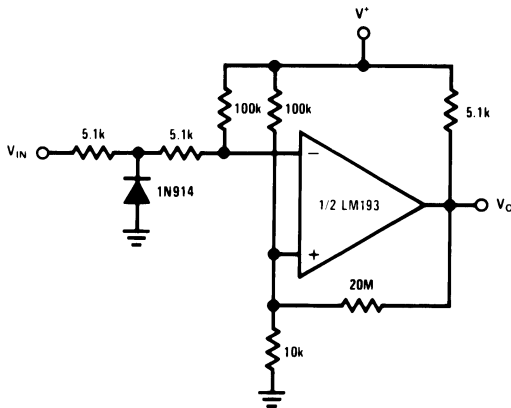


Figure 31. Zero Crossing Detector (Single Power Supply)

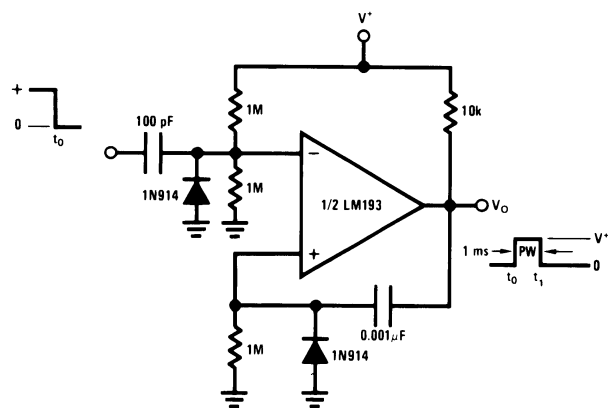


Figure 32. One-Shot Multivibrator

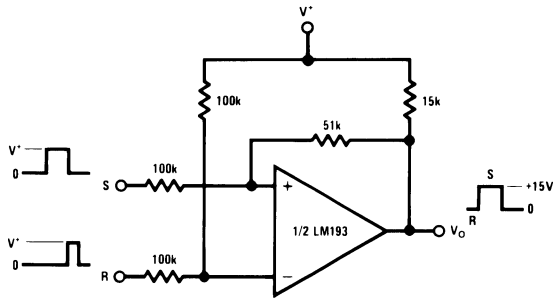


Figure 33. Bi-Stable Multivibrator

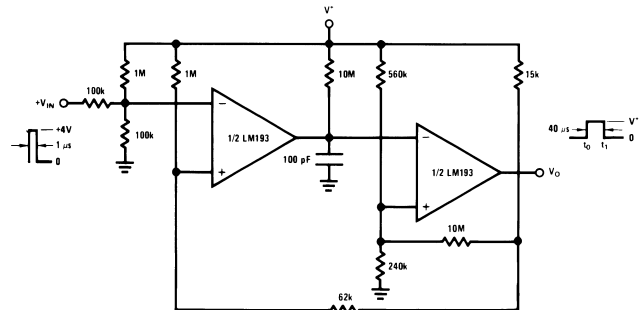


Figure 34. One-Shot Multivibrator With Input Lock Out

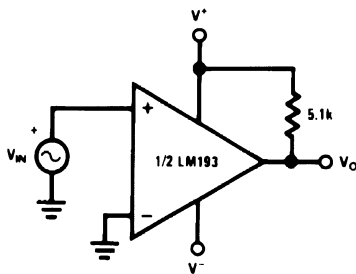


Figure 35. Zero Crossing Detector

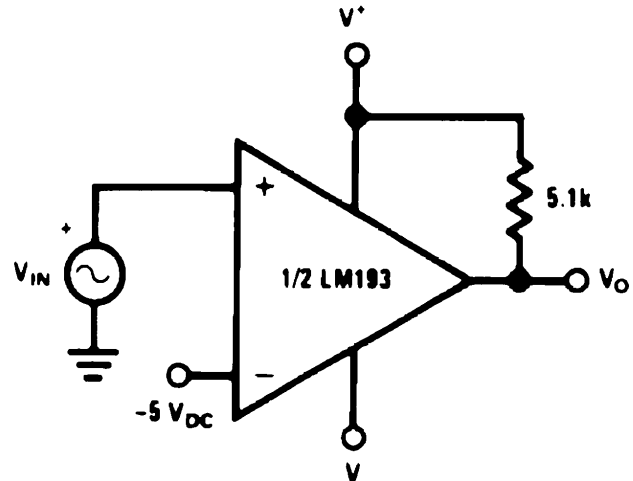


Figure 36. Comparator With a Negative Reference

Typical Applications (continued)

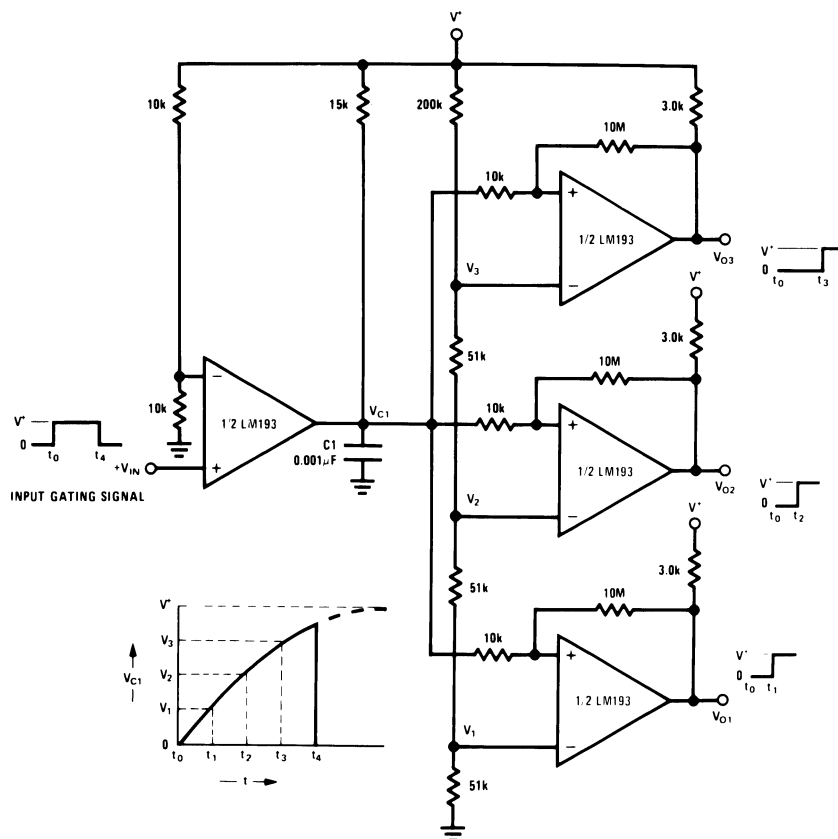


Figure 37. Time Delay Generator

9 Power Supply Recommendations

Even in low frequency applications, the LM139-N can have internal transients which are extremely quick. For this reason, bypassing the power supply with 1.0 μ F to ground will provide improved performance; the supply bypass capacitor should be placed as close as possible to the supply pin and have a solid connection to ground. The bypass capacitor should have a low ESR and also a SRF greater than 50MHz.

10 Layout

10.1 Layout Guidelines

Try to minimize parasitic impedances on the inputs to avoid oscillation. Any positive feedback used as hysteresis should place the feedback components as close as possible to the input pins. Care should be taken to ensure that the output pins do not couple to the inputs. This can occur through capacitive coupling if the traces are too close and lead to oscillations on the output. The optimum placement for the bypass capacitor is closest to the V+ and ground pins. Take care to minimize the loop area formed by the bypass capacitor connection between V+ and ground. The ground pin should be connected to the PCB ground plane at the pin of the device. The feedback components should be placed as close to the device as possible minimizing strays.

10.2 Layout Example

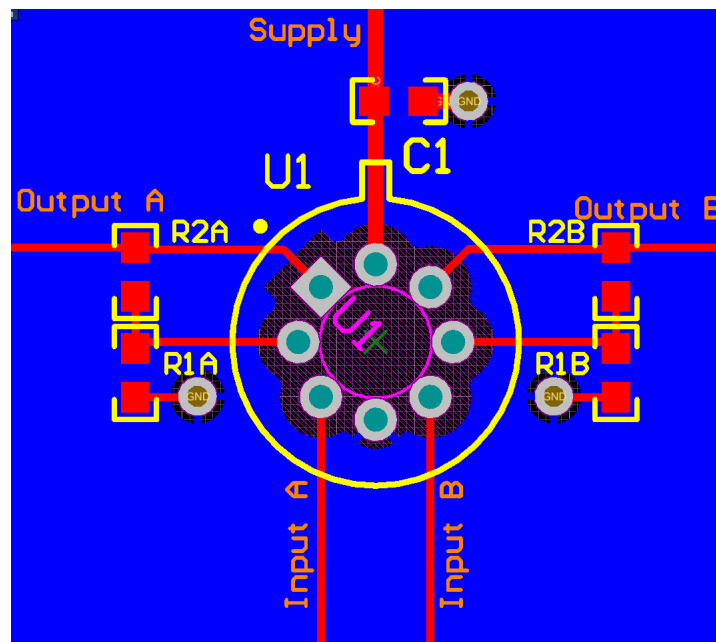


Figure 38. Layout Example

11 Device and Documentation Support

11.1 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to order now.

Table 1. Related Links

PARTS	PRODUCT FOLDER	ORDER NOW	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
LM193-N	Click here	Click here	Click here	Click here	Click here
LM2903-N	Click here	Click here	Click here	Click here	Click here
LM293-N	Click here	Click here	Click here	Click here	Click here
LM393-N	Click here	Click here	Click here	Click here	Click here

11.2 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. In the upper right corner, click on *Alert me* to register and receive a weekly digest of any product information that has changed. For change details, review the revision history included in any revised document.

11.3 Trademarks

All trademarks are the property of their respective owners.

11.4 Electrostatic Discharge Caution



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

11.5 Glossary

[SLYZ022](#) — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

12 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
LM193AH	ACTIVE	TO-99	LMC	8	500	Non-RoHS & Non-Green	Call TI	Call TI	-55 to 125	(LM193AH, LM193AH)	Samples
LM193AH/NOPB	ACTIVE	TO-99	LMC	8	500	RoHS & Green	Call TI	Level-1-NA-UNLIM	-55 to 125	(LM193AH, LM193AH)	Samples
LM193H	ACTIVE	TO-99	LMC	8	500	Non-RoHS & Non-Green	Call TI	Call TI	-55 to 125	(LM193H, LM193H)	Samples
LM193H/NOPB	ACTIVE	TO-99	LMC	8	500	RoHS & Green	Call TI	Level-1-NA-UNLIM	-55 to 125	(LM193H, LM193H)	Samples
LM2903ITL/NOPB	ACTIVE	DSBGA	YZR	8	250	RoHS & Green	SNAGCU	Level-1-260C-UNLIM	-40 to 85	C 03	Samples
LM2903ITLX/NOPB	ACTIVE	DSBGA	YZR	8	3000	RoHS & Green	SNAGCU	Level-1-260C-UNLIM	-40 to 85	C 03	Samples
LM2903M	ACTIVE	SOIC	D	8	95	Non-RoHS & Green	Call TI	Level-1-235C-UNLIM	-40 to 85	LM 2903M	Samples
LM2903M/NOPB	ACTIVE	SOIC	D	8	95	RoHS & Green	SN	Level-1-260C-UNLIM	-40 to 85	LM 2903M	Samples
LM2903MX/NOPB	ACTIVE	SOIC	D	8	2500	RoHS & Green	SN	Level-1-260C-UNLIM	-40 to 85	LM 2903M	Samples
LM2903N/NOPB	ACTIVE	PDIP	P	8	40	RoHS & Green	NIPDAU	Level-1-NA-UNLIM	-40 to 85	LM 2903N	Samples
LM293H	ACTIVE	TO-99	LMC	8	500	Non-RoHS & Non-Green	Call TI	Call TI	-25 to 85	(LM293H, LM293H)	Samples
LM293H/NOPB	ACTIVE	TO-99	LMC	8	500	RoHS & Green	Call TI	Level-1-NA-UNLIM	-25 to 85	(LM293H, LM293H)	Samples
LM393M	ACTIVE	SOIC	D	8	95	Non-RoHS & Green	Call TI	Level-1-235C-UNLIM	0 to 70	LM 393M	Samples
LM393M/NOPB	ACTIVE	SOIC	D	8	95	RoHS & Green	SN	Level-1-260C-UNLIM	0 to 70	LM 393M	Samples
LM393MX	ACTIVE	SOIC	D	8	2500	Non-RoHS & Green	Call TI	Level-1-235C-UNLIM	0 to 70	LM 393M	Samples
LM393MX/NOPB	ACTIVE	SOIC	D	8	2500	RoHS & Green	SN	Level-1-260C-UNLIM	0 to 70	LM 393M	Samples
LM393N/NOPB	ACTIVE	PDIP	P	8	40	RoHS & Green	NIPDAU	Level-1-NA-UNLIM	0 to 70	LM 393N	Samples

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
LM393TL/NOPB	ACTIVE	DSBGA	YZR	8	250	RoHS & Green	SNAGCU	Level-1-260C-UNLIM	0 to 70	C 02	Samples
LM393TLX/NOPB	ACTIVE	DSBGA	YZR	8	3000	RoHS & Green	SNAGCU	Level-1-260C-UNLIM	0 to 70	C 02	Samples

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSELETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".

RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.

Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer:The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

OTHER QUALIFIED VERSIONS OF LM2903-N, LM293-N :

- Automotive : [LM2903-Q1](#)
- Enhanced Product : [LM293-EP](#)

NOTE: Qualified Version Definitions:

- Automotive - Q100 devices qualified for high-reliability automotive applications targeting zero defects
- Enhanced Product - Supports Defense, Aerospace and Medical Applications

TAPE AND REEL INFORMATION

QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE


*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
LM2903ITL/NOPB	DSBGA	YZR	8	250	178.0	8.4	1.7	1.7	0.76	4.0	8.0	Q1
LM2903ITLX/NOPB	DSBGA	YZR	8	3000	178.0	8.4	1.7	1.7	0.76	4.0	8.0	Q1
LM2903MX/NOPB	SOIC	D	8	2500	330.0	12.4	6.5	5.4	2.0	8.0	12.0	Q1
LM393MX	SOIC	D	8	2500	330.0	12.4	6.5	5.4	2.0	8.0	12.0	Q1
LM393MX/NOPB	SOIC	D	8	2500	330.0	12.4	6.5	5.4	2.0	8.0	12.0	Q1
LM393TL/NOPB	DSBGA	YZR	8	250	178.0	8.4	1.7	1.7	0.76	4.0	8.0	Q1
LM393TLX/NOPB	DSBGA	YZR	8	3000	178.0	8.4	1.7	1.7	0.76	4.0	8.0	Q1

TAPE AND REEL BOX DIMENSIONS


*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
LM2903ITL/NOPB	DSBGA	YZR	8	250	208.0	191.0	35.0
LM2903ITLX/NOPB	DSBGA	YZR	8	3000	208.0	191.0	35.0
LM2903MX/NOPB	SOIC	D	8	2500	367.0	367.0	35.0
LM393MX	SOIC	D	8	2500	367.0	367.0	35.0
LM393MX/NOPB	SOIC	D	8	2500	367.0	367.0	35.0
LM393TL/NOPB	DSBGA	YZR	8	250	208.0	191.0	35.0
LM393TLX/NOPB	DSBGA	YZR	8	3000	208.0	191.0	35.0

TUBE


*All dimensions are nominal

Device	Package Name	Package Type	Pins	SPQ	L (mm)	W (mm)	T (μm)	B (mm)
LM2903M	D	SOIC	8	95	495	8	4064	3.05
LM2903M	D	SOIC	8	95	495	8	4064	3.05
LM2903M/NOPB	D	SOIC	8	95	495	8	4064	3.05
LM2903N/NOPB	P	PDIP	8	40	502	14	11938	4.32
LM393M	D	SOIC	8	95	495	8	4064	3.05
LM393M	D	SOIC	8	95	495	8	4064	3.05
LM393M/NOPB	D	SOIC	8	95	495	8	4064	3.05
LM393N/NOPB	P	PDIP	8	40	502	14	11938	4.32



D0008A

PACKAGE OUTLINE

SOIC - 1.75 mm max height

SMALL OUTLINE INTEGRATED CIRCUIT



4214825/C 02/2019

NOTES:

- Linear dimensions are in inches [millimeters]. Dimensions in parenthesis are for reference only. Controlling dimensions are in inches. Dimensioning and tolerancing per ASME Y14.5M.
- This drawing is subject to change without notice.
- This dimension does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed $.006$ [0.15] per side.
- This dimension does not include interlead flash.
- Reference JEDEC registration MS-012, variation AA.

EXAMPLE BOARD LAYOUT

D0008A

SOIC - 1.75 mm max height

SMALL OUTLINE INTEGRATED CIRCUIT



LAND PATTERN EXAMPLE
EXPOSED METAL SHOWN
SCALE:8X



SOLDER MASK DETAILS

4214825/C 02/2019

NOTES: (continued)

6. Publication IPC-7351 may have alternate designs.

7. Solder mask tolerances between and around signal pads can vary based on board fabrication site.

EXAMPLE STENCIL DESIGN

D0008A

SOIC - 1.75 mm max height

SMALL OUTLINE INTEGRATED CIRCUIT



SOLDER PASTE EXAMPLE
BASED ON .005 INCH [0.125 MM] THICK STENCIL
SCALE:8X

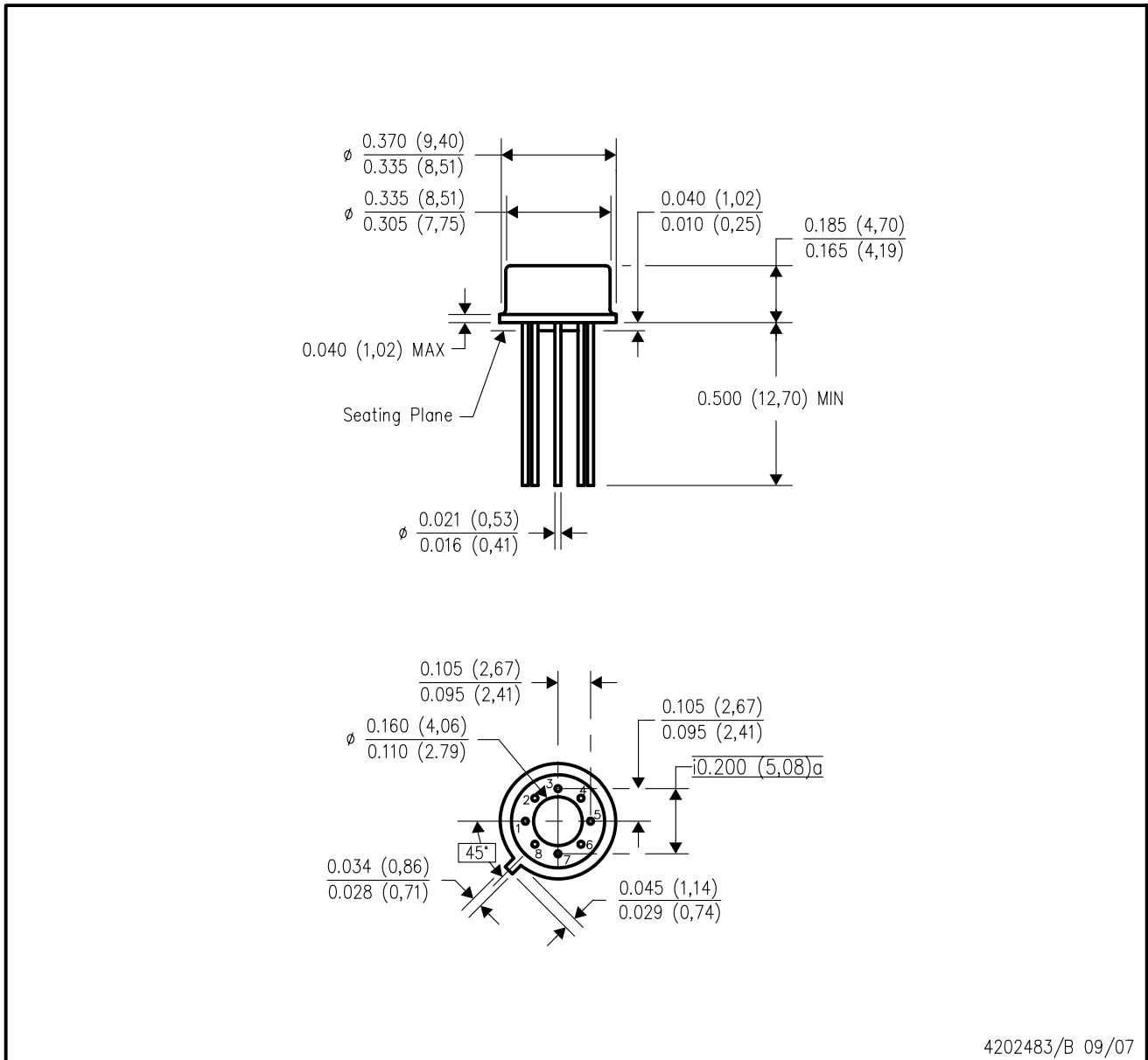
4214825/C 02/2019

NOTES: (continued)

8. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release. IPC-7525 may have alternate design recommendations.
9. Board assembly site may have different recommendations for stencil design.

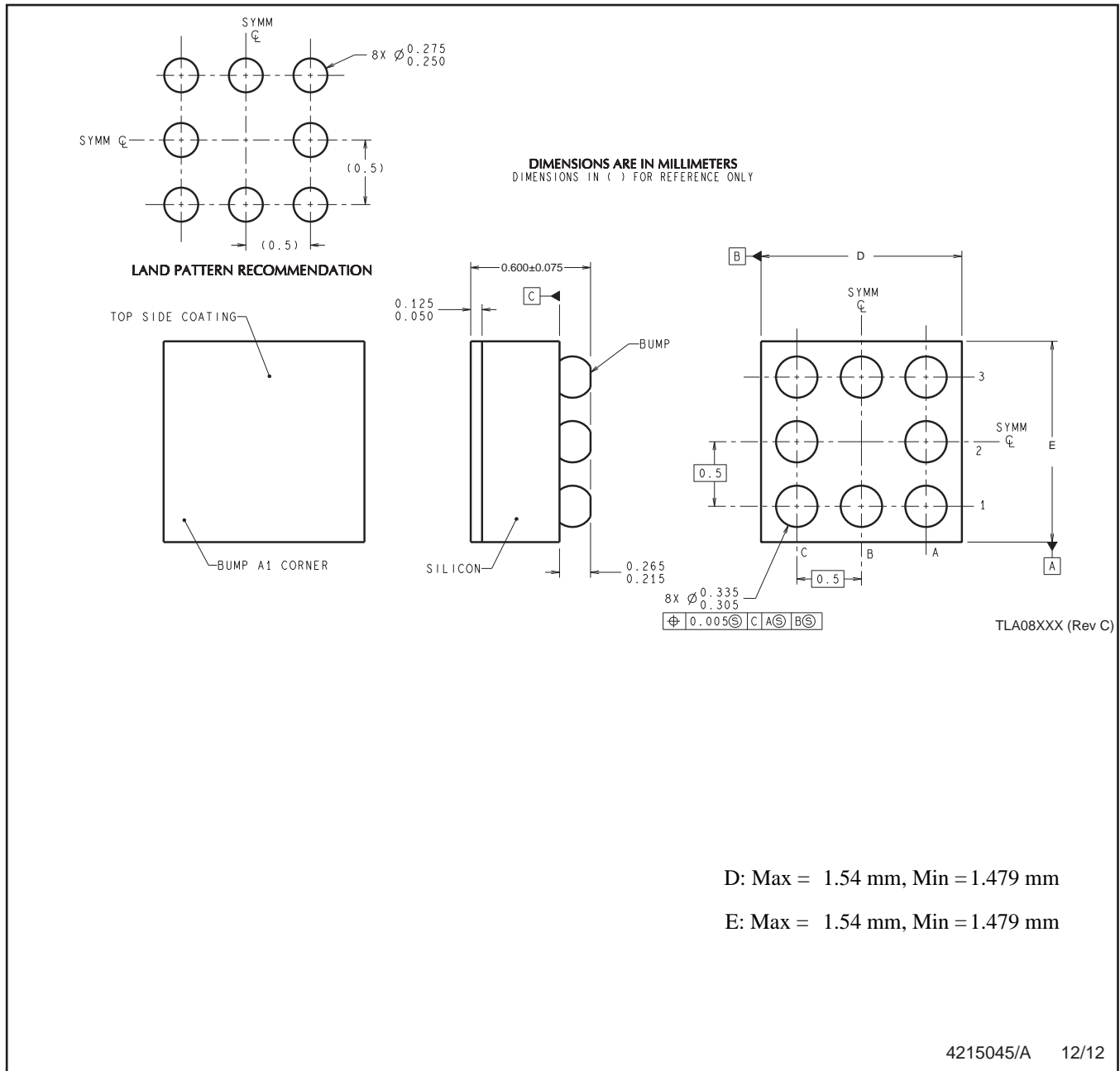
LMC (O-MBCY-W8)

METAL CYLINDRICAL PACKAGE



- NOTES:
- A. All linear dimensions are in inches (millimeters).
 - B. This drawing is subject to change without notice.
 - C. Leads in true position within 0.010 (0,25) R @ MMC at seating plane.
 - D. Pin numbers shown for reference only. Numbers may not be marked on package.
 - E. Falls within JEDEC MO-002/TO-99.

YZR0008



D: Max = 1.54 mm, Min = 1.479 mm

E: Max = 1.54 mm, Min = 1.479 mm

4215045/A 12/12

NOTES: A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M-1994.
 B. This drawing is subject to change without notice.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

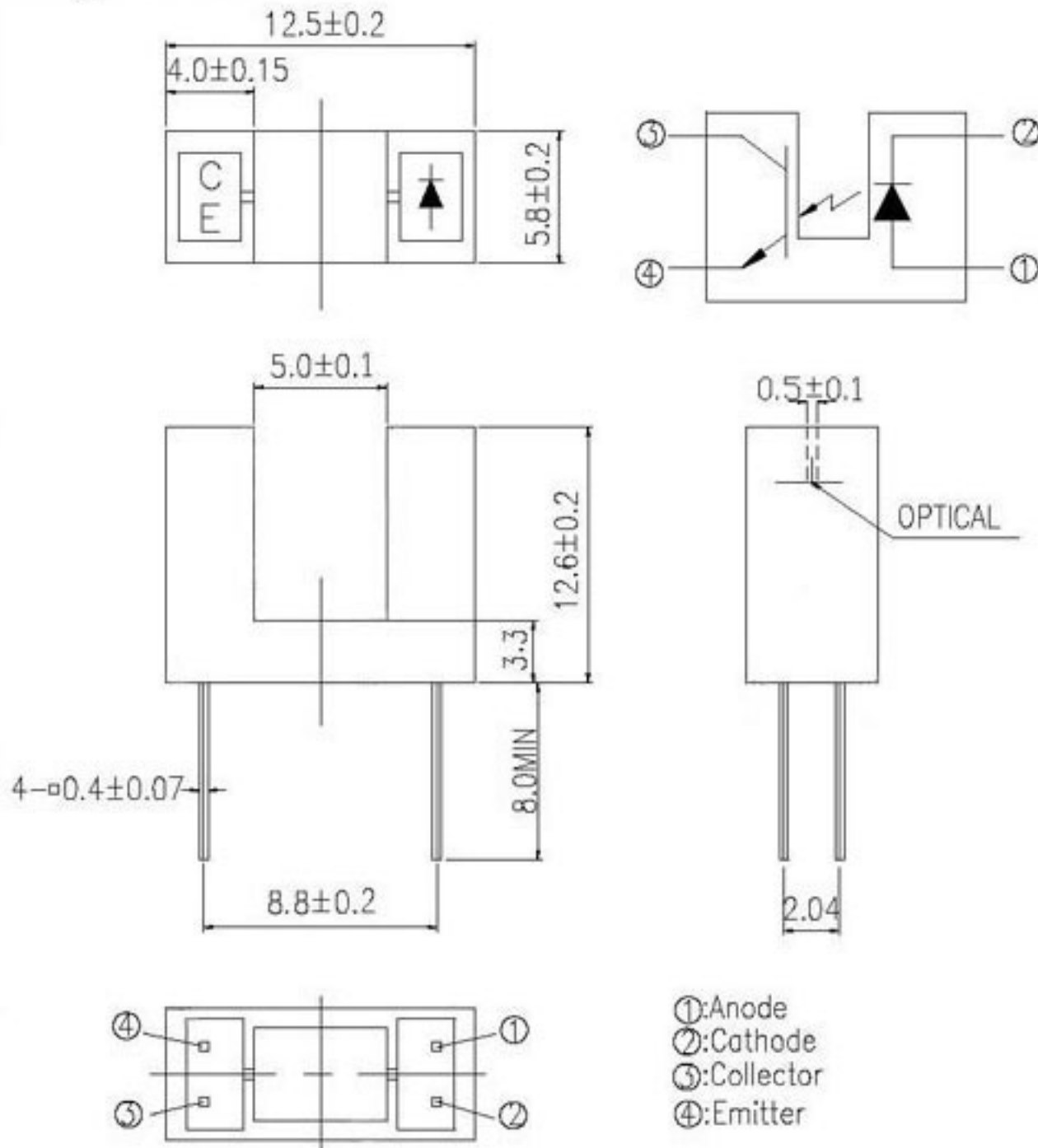
These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated

■ Package Dimensions



Absolute Maximum Ratings (Ta=25°C)

Parameter		Symbol	Ratings	Unit
Input	Power Dissipation at(or below) 25°C Free Air Temperature	Pd	75	mW
	Reverse Voltage	V _R	5	V
	Forward Current	I _F	50	mA
	Peak Forward Current (*1) Pulse width ≤ 100 μs, Duty cycle=1%	I _{FP}	1	A
Output	Collector Power Dissipation	P _C	75	mW
	Collector Current	I _C	20	mA
	Collector-Emitter Voltage	B V _{CEO}	30	V
	Emitter-Collector Voltage	B V _{ECO}	5	V
Operating Temperature		Topr	-25~+85	°C
Storage Temperature		Tstg	-40~+85	°C
Lead Soldering Temperature (*2) (1/16 inch form body for 5 seconds)		Tsol	260	°C

 (*1) $t_w=100 \mu sec.$, $T=10 msec.$ (*2) $t=5 Sec$
Electro-Optical Characteristics (Ta=25°C)

Parameter		Symbol	Min.	Typ.	Max.	Unit	Conditions
Input	Forward Voltage	V _{F1}	---	1.2	1.5	V	I _F =20mA
		V _{F2}	---	1.4	1.85		I _F =100mA, t _p =100 μs, t _p /T=0.01
		V _{F3}	---	2.6	4.0		I _F =1A, t _p =100 μs, t _p /T=0.01
	Reverse Current	I _R	---	---	10	μA	V _R =5V
	Peak Wavelength	λ _p	---	940	---	nm	I _F =20mA
	View Angle	2θ _{1/2}	---	60	---	Deg	I _F =20mA
Output	Dark Current	I _{CEO}	---	---	100	nA	V _{CE} =20V, E _e =0mW/cm ²
	C-E Saturation Voltage	V _{CE(sat)}	---	---	0.4	V	I _C =2mA , E _e =1mW/cm ²
Transfer Characteristics	Collect Current	I _{C(ON)}	0.5	---	10	mA	V _{CE} =5V I _F =20mA
	Rise time	t _r	---	15	---	μsec	V _{CE} =5V
	Fall time	t _f	---	15	---	μsec	I _C =1mA R _L =1KΩ

Typical Electrical/Optical/Characteristics Curves for IR

Fig.1 Forward Current vs.

Ambient Temperature

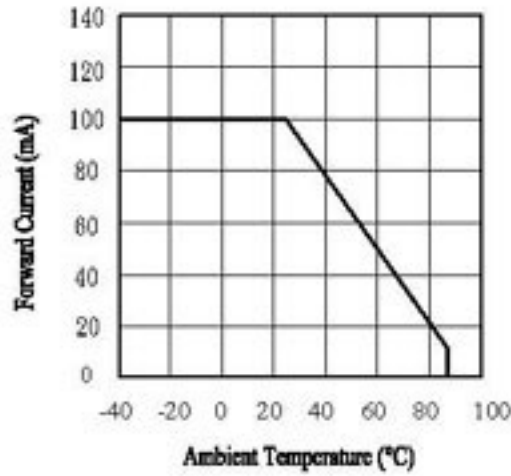


Fig.2 Spectral Distribution

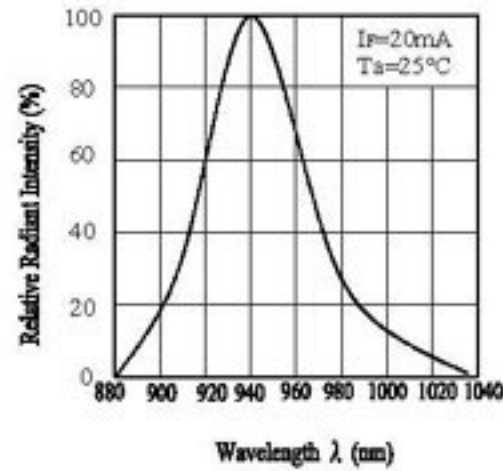


Fig.5 Relative Intensity vs.

Forward Current

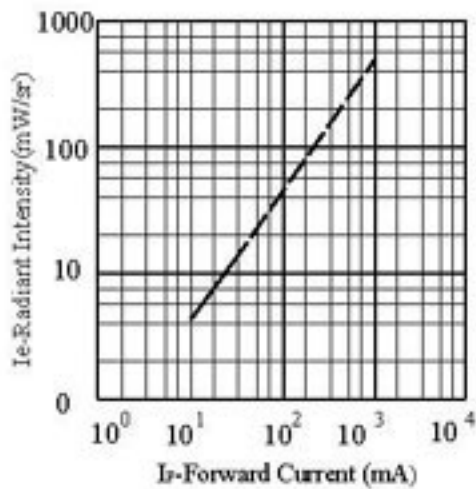


Fig.6 Relative Radiant Intensity vs.

Angular Displacement

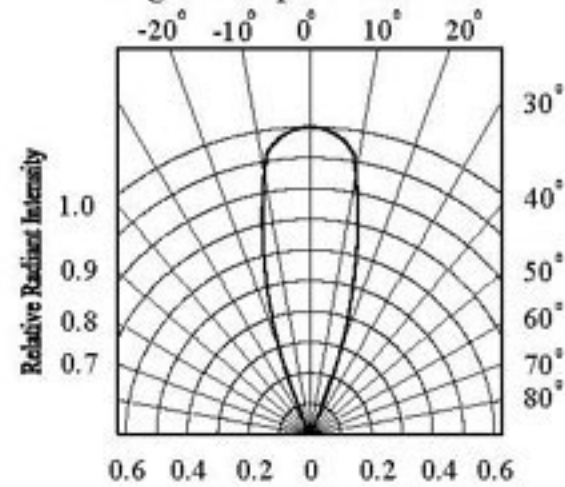


Fig.7 Relative Intensity vs.

Ambient Temperature(°C)

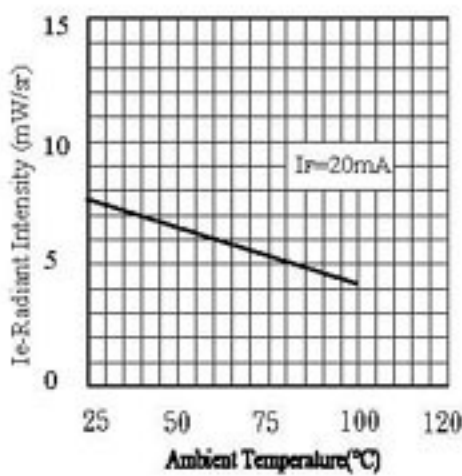
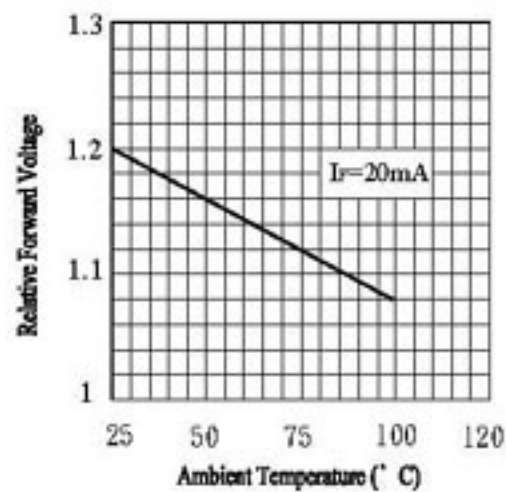


Fig.8 Forward Current vs.

Ambient Temperature(°C)



■ Typical Electrical/Optical/Characteristics Curves for PT

Fig.1 Collector Power Dissipation vs. Ambient Temperature

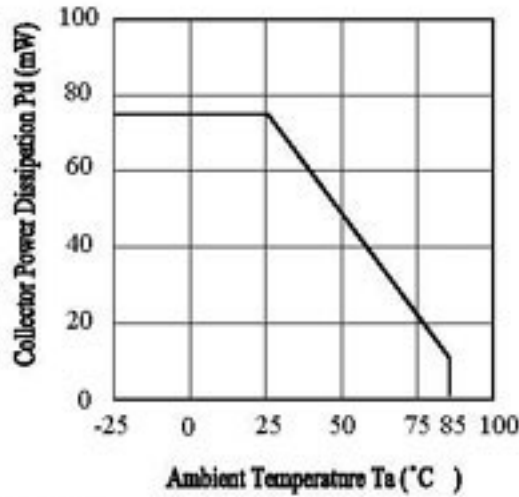


Fig.2 Spectral Sensitivity

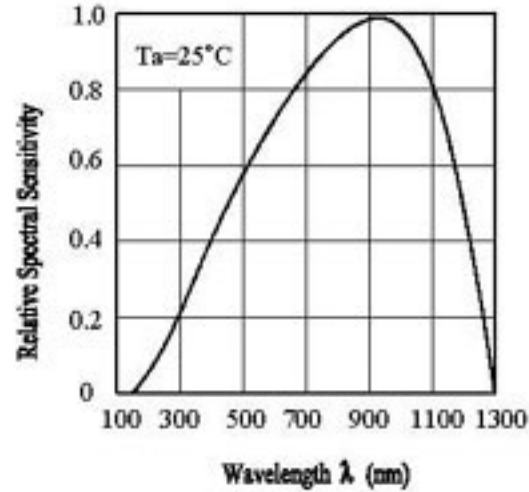


Fig.3 Relative Collector Current vs. Ambient Temperature

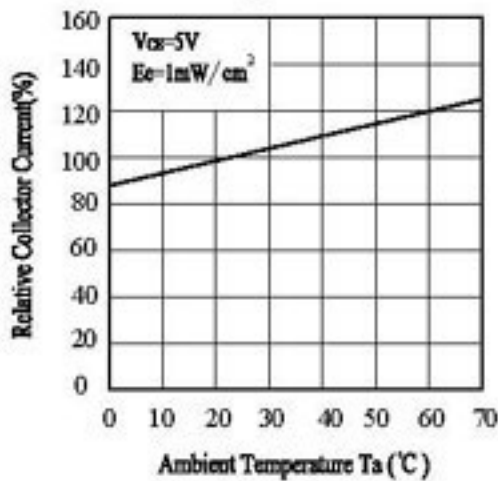


Fig.4 Collector Current vs. Irradiance

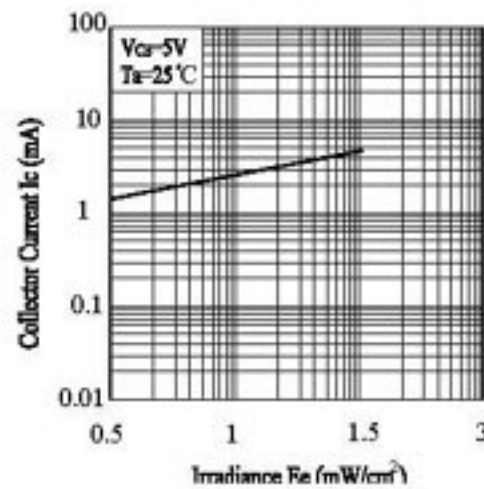


Fig.5 Collector Dark Current vs. Ambient Temperature

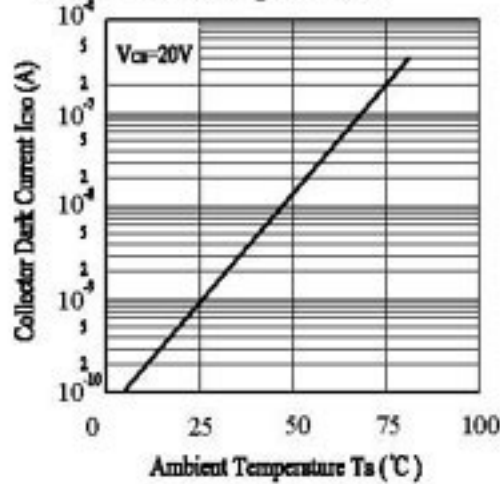
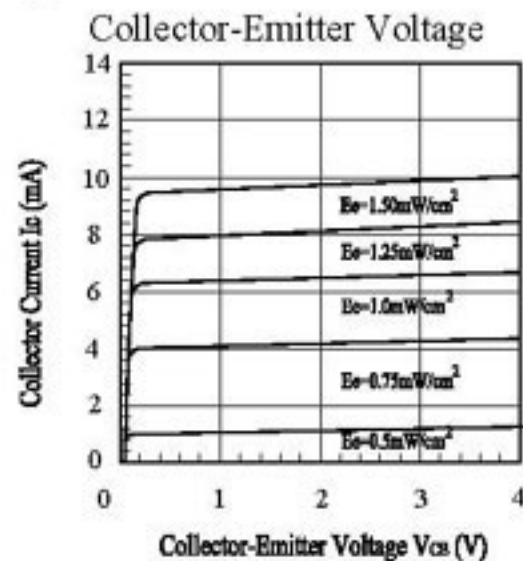


Fig.6 Collector Current vs. Collector-Emitter Voltage



HD44780U (LCD-II)

(Dot Matrix Liquid Crystal Display Controller/Driver)

HITACHI

ADE-207-272(Z)

'99.9

Rev. 0.0

Description

The HD44780U dot-matrix liquid crystal display controller and driver LSI displays alphanumerics, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

A single HD44780U can display up to one 8-character line or two 8-character lines.

The HD44780U has pin function compatibility with the HD44780S which allows the user to easily replace an LCD-II with an HD44780U. The HD44780U character generator ROM is extended to generate 208 5×8 dot character fonts and 32 5×10 dot character fonts for a total of 240 different character fonts.

The low power supply (2.7V to 5.5V) of the HD44780U is suitable for any portable battery-driven product requiring low power dissipation.

Features

- 5×8 and 5×10 dot matrix possible
- Low power operation support:
 - 2.7 to 5.5V
- Wide range of liquid crystal display driver power
 - 3.0 to 11V
- Liquid crystal drive waveform
 - A (One line frequency AC waveform)
- Correspond to high speed MPU bus interface
 - 2 MHz (when $V_{CC} = 5V$)
- 4-bit or 8-bit MPU interface enabled
- 80 \times 8-bit display RAM (80 characters max.)
- 9,920-bit character generator ROM for a total of 240 character fonts
 - 208 character fonts (5×8 dot)
 - 32 character fonts (5×10 dot)

HD44780U

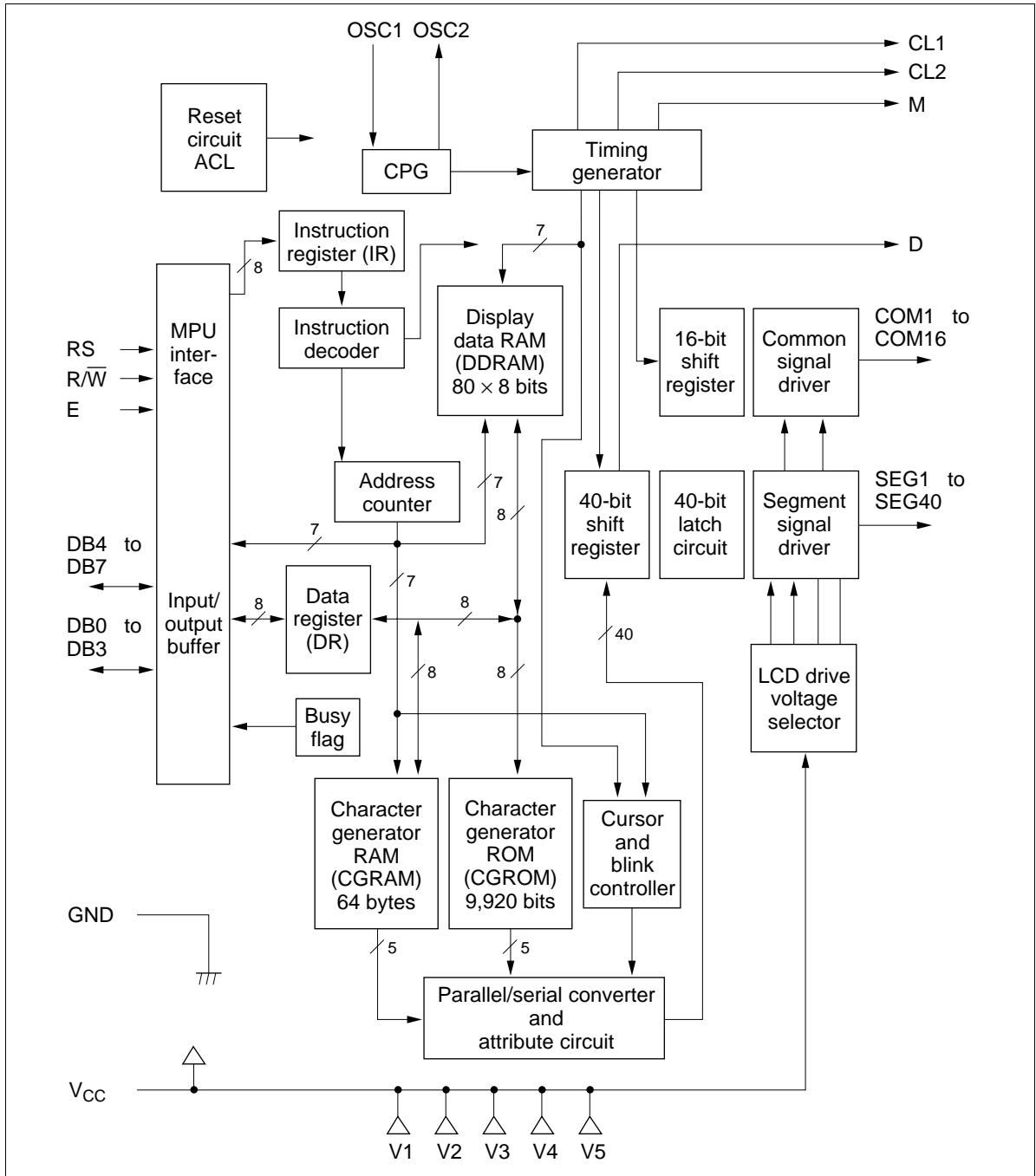
- 64 × 8-bit character generator RAM
 - 8 character fonts (5 × 8 dot)
 - 4 character fonts (5 × 10 dot)
- 16-common × 40-segment liquid crystal display driver
- Programmable duty cycles
 - 1/8 for one line of 5 × 8 dots with cursor
 - 1/11 for one line of 5 × 10 dots with cursor
 - 1/16 for two lines of 5 × 8 dots with cursor
- Wide range of instruction functions:
 - Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
- Pin function compatibility with HD44780S
- Automatic reset circuit that initializes the controller/driver after power on
- Internal oscillator with external resistors
- Low power consumption

Ordering Information

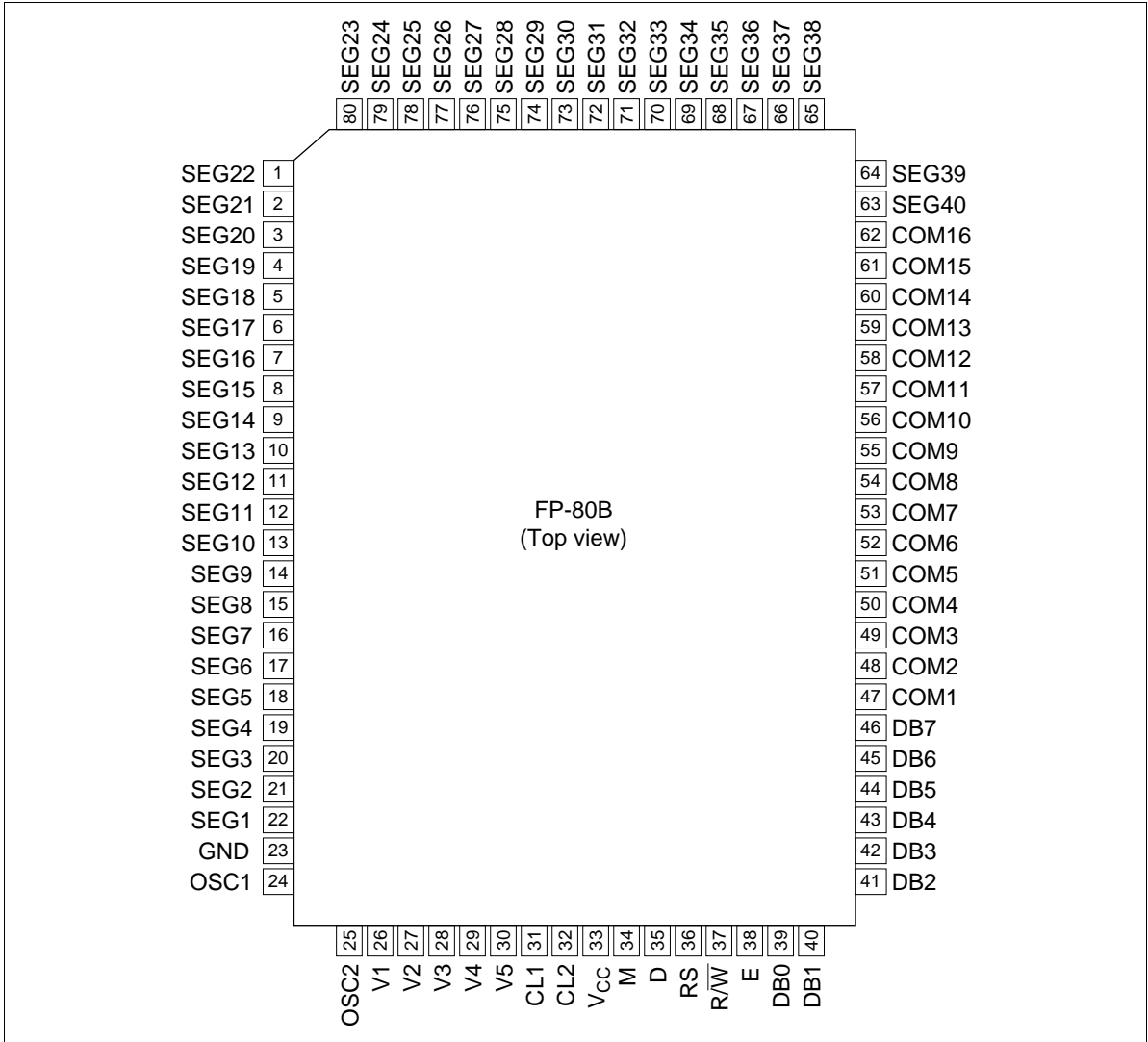
Type No.	Package	CGROM
HD44780UA00FS HCD44780UA00 HD44780UA00TF	FP-80B Chip TFP-80F	Japanese standard font
HD44780UA02FS HCD44780UA02 HD44780UA02TF	FP-80B Chip TFP-80F	European standard font
HD44780UBxxFS HCD44780UBxx HD44780UBxxTF	FP-80B Chip TFP-80F	Custom font

Note: xx: ROM code No.

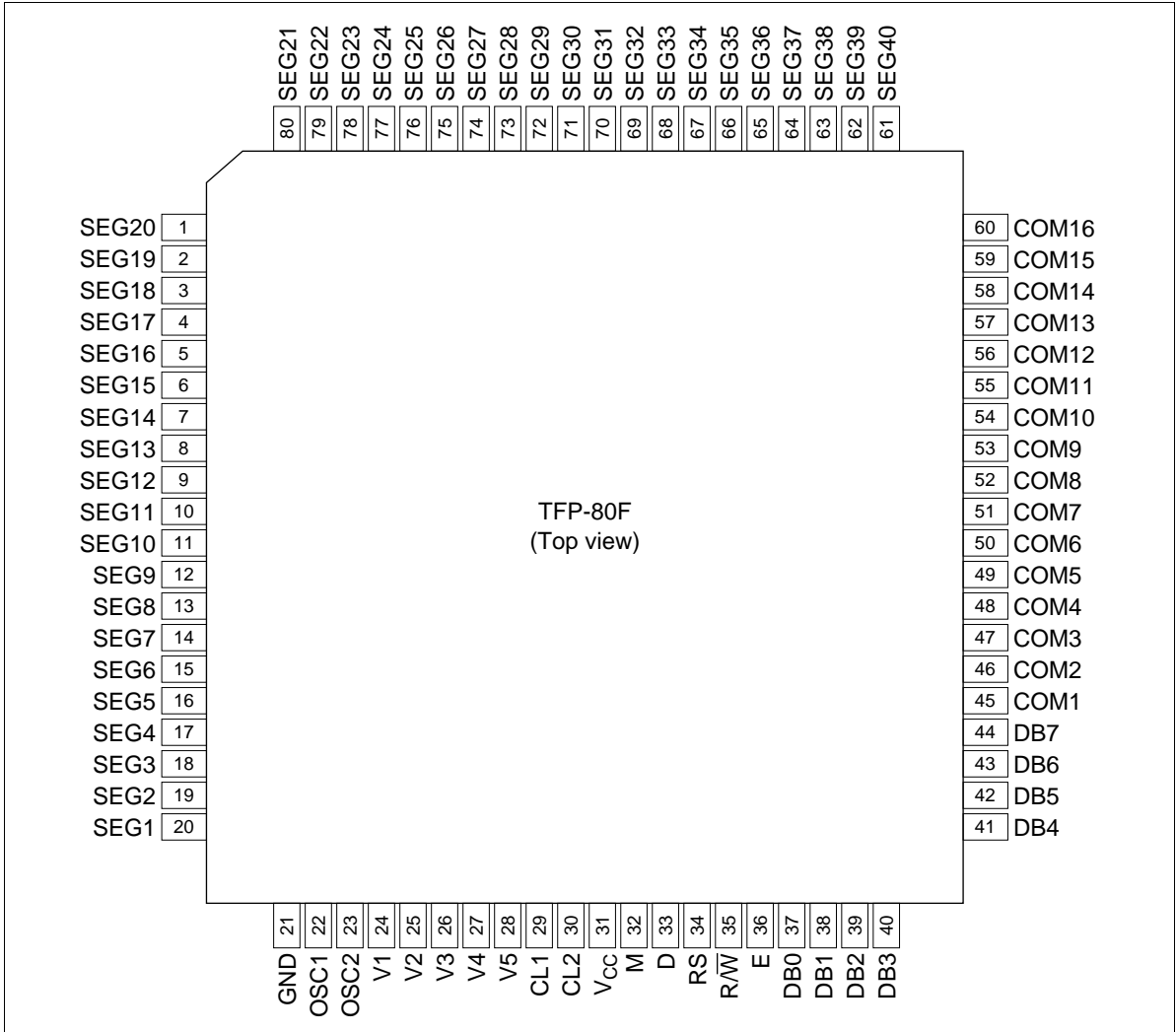
HD44780U Block Diagram



HD44780U Pin Arrangement (FP-80B)



HD44780U Pin Arrangement (TFP-80F)



HD44780U

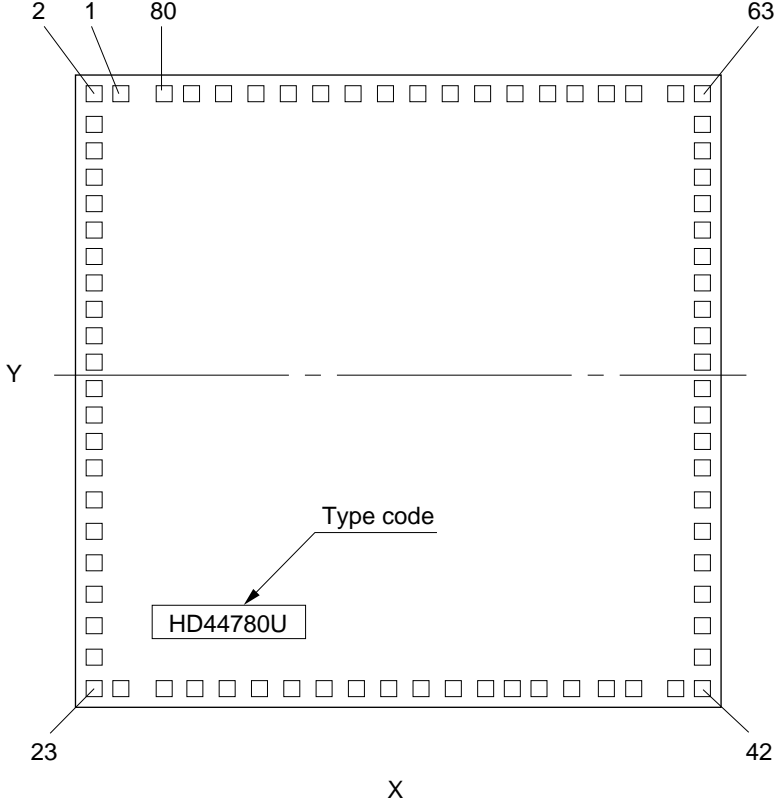
HD44780U Pad Arrangement

Chip size: $4.90 \times 4.90 \text{ mm}^2$

Coordinate: Pad center (μm)

Origin: Chip center

Pad size: $114 \times 114 \mu\text{m}^2$



HCD44780U Pad Location Coordinates

Pad No.	Function	Coordinate		Pad No.	Function	Coordinate	
		X (um)	Y (um)			X (um)	Y (um)
1	SEG22	-2100	2313	41	DB2	2070	-2290
2	SEG21	-2280	2313	42	DB3	2260	-2290
3	SEG20	-2313	2089	43	DB4	2290	-2099
4	SEG19	-2313	1833	44	DB5	2290	-1883
5	SEG18	-2313	1617	45	DB6	2290	-1667
6	SEG17	-2313	1401	46	DB7	2290	-1452
7	SEG16	-2313	1186	47	COM1	2313	-1186
8	SEG15	-2313	970	48	COM2	2313	-970
9	SEG14	-2313	755	49	COM3	2313	-755
10	SEG13	-2313	539	50	COM4	2313	-539
11	SEG12	-2313	323	51	COM5	2313	-323
12	SEG11	-2313	108	52	COM6	2313	-108
13	SEG10	-2313	-108	53	COM7	2313	108
14	SEG9	-2313	-323	54	COM8	2313	323
15	SEG8	-2313	-539	55	COM9	2313	539
16	SEG7	-2313	-755	56	COM10	2313	755
17	SEG6	-2313	-970	57	COM11	2313	970
18	SEG5	-2313	-1186	58	COM12	2313	1186
19	SEG4	-2313	-1401	59	COM13	2313	1401
20	SEG3	-2313	-1617	60	COM14	2313	1617
21	SEG2	-2313	-1833	61	COM15	2313	1833
22	SEG1	-2313	-2073	62	COM16	2313	2095
23	GND	-2280	-2290	63	SEG40	2296	2313
24	OSC1	-2080	-2290	64	SEG39	2100	2313
25	OSC2	-1749	-2290	65	SEG38	1617	2313
26	V1	-1550	-2290	66	SEG37	1401	2313
27	V2	-1268	-2290	67	SEG36	1186	2313
28	V3	-941	-2290	68	SEG35	970	2313
29	V4	-623	-2290	69	SEG34	755	2313
30	V5	-304	-2290	70	SEG33	539	2313
31	CL1	-48	-2290	71	SEG32	323	2313
32	CL2	142	-2290	72	SEG31	108	2313
33	V _{CC}	309	-2290	73	SEG30	-108	2313
34	M	475	-2290	74	SEG29	-323	2313
35	D	665	-2290	75	SEG28	-539	2313
36	RS	832	-2290	76	SEG27	-755	2313
37	R/W	1022	-2290	77	SEG26	-970	2313
38	E	1204	-2290	78	SEG25	-1186	2313
39	DB0	1454	-2290	79	SEG24	-1401	2313
40	DB1	1684	-2290	80	SEG23	-1617	2313

Pin Functions

Signal	No. of Lines	I/O	Device Interfaced with	Function
RS	1	I	MPU	Selects registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/W	1	I	MPU	Selects read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. DB7 can be used as a busy flag.
DB0 to DB3	4	I/O	MPU	Four low order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the HD44780U. These pins are not used during 4-bit operation.
CL1	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
CL2	1	O	Extension driver	Clock to shift serial data D
M	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
D	1	O	Extension driver	Character pattern data corresponding to each segment signal
COM1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveforms. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.
SEG1 to SEG40	40	O	LCD	Segment signals
V1 to V5	5	—	Power supply	Power supply for LCD drive $V_{CC} - V5 = 11\text{ V (max)}$
V_{CC} , GND	2	—	Power supply	V_{CC} : 2.7V to 5.5V, GND: 0V
OSC1, OSC2	2	—	Oscillation resistor clock	When crystal oscillation is performed, a resistor must be connected externally. When the pin input is an external clock, it must be input to OSC1.

Function Description

Registers

The HD44780U has two 8-bit registers, an instruction register (IR) and a data register (DR).

The IR stores instruction codes, such as display clear and cursor shift, and address information for display data RAM (DDRAM) and character generator RAM (CGRAM). The IR can only be written from the MPU.

The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR from the MPU is automatically written into DDRAM or CGRAM by an internal operation. The DR is also used for data storage when reading data from DDRAM or CGRAM. When address information is written into the IR, data is read and then stored into the DR from DDRAM or CGRAM by an internal operation. Data transfer between the MPU is then completed when the MPU reads the DR. After the read, data in DDRAM or CGRAM at the next address is sent to the DR for the next read from the MPU. By the register selector (RS) signal, these two registers can be selected (Table 1).

Busy Flag (BF)

When the busy flag is 1, the HD44780U is in the internal operation mode, and the next instruction will not be accepted. When $RS = 0$ and $R/\overline{W} = 1$ (Table 1), the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

Address Counter (AC)

The address counter (AC) assigns addresses to both DDRAM and CGRAM. When an address of an instruction is written into the IR, the address information is sent from the IR to the AC. Selection of either DDRAM or CGRAM is also determined concurrently by the instruction.

After writing into (reading from) DDRAM or CGRAM, the AC is automatically incremented by 1 (decremented by 1). The AC contents are then output to DB0 to DB6 when $RS = 0$ and $R/\overline{W} = 1$ (Table 1).

Table 1 **Register Selection**

RS	R/\overline{W}	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Display Data RAM (DDRAM)

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80×8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses and positions on the liquid crystal display.

The DDRAM address (A_{DD}) is set in the address counter (AC) as hexadecimal.

- 1-line display ($N = 0$) (Figure 2)
 - When there are fewer than 80 display characters, the display begins at the head position. For example, if using only the HD44780, 8 characters are displayed. See Figure 3.
 - When the display shift operation is performed, the DDRAM address shifts. See Figure 3.

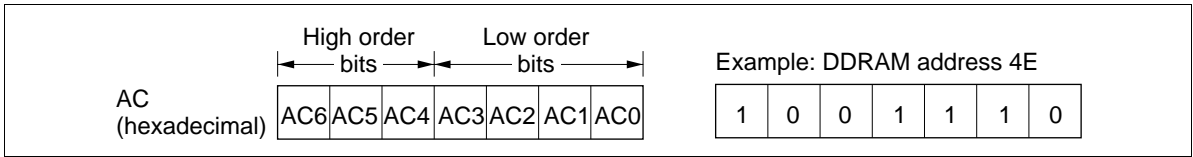


Figure 1 DDRAM Address

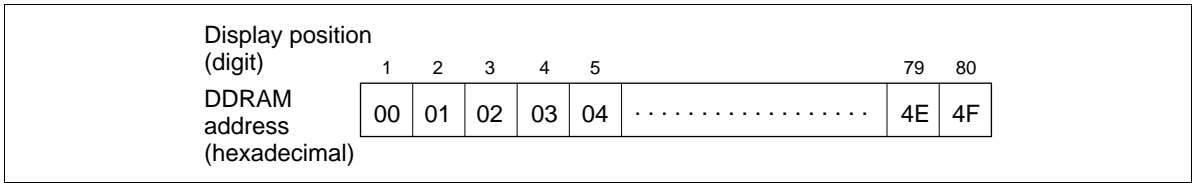


Figure 2 1-Line Display

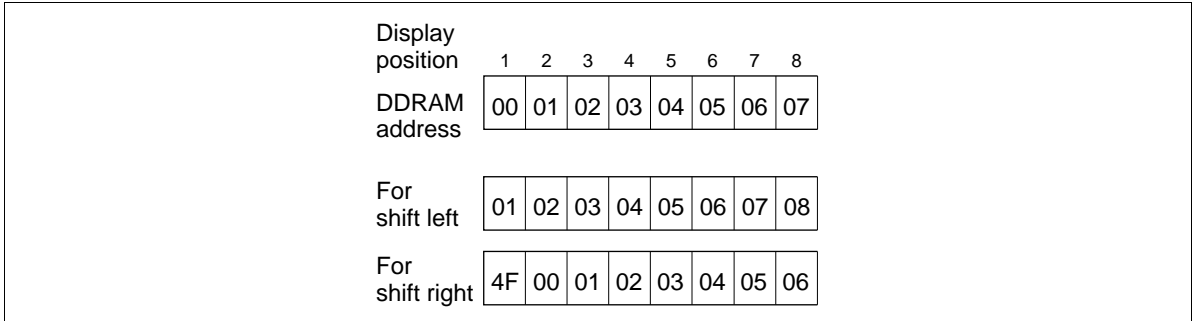


Figure 3 1-Line by 8-Character Display Example

- 2-line display (N = 1) (Figure 4)
 - Case 1: When the number of display characters is less than 40×2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters \times 2 lines are displayed. See Figure 5.

When display shift operation is performed, the DDRAM address shifts. See Figure 5.

Display position	1	2	3	4	5	39	40
DDRAM address (hexadecimal)	00	01	02	03	04	26	27
	40	41	42	43	44	66	67

Figure 4 2-Line Display

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07
	40	41	42	43	44	45	46	47
For shift left	01	02	03	04	05	06	07	08
	41	42	43	44	45	46	47	48
For shift right	27	00	01	02	03	04	05	06
	67	40	41	42	43	44	45	46

Figure 5 2-Line by 8-Character Display Example

— Case 2: For a 16-character × 2-line display, the HD44780 can be extended using one 40-output extension driver. See Figure 6.

When display shift operation is performed, the DDRAM address shifts. See Figure 6.

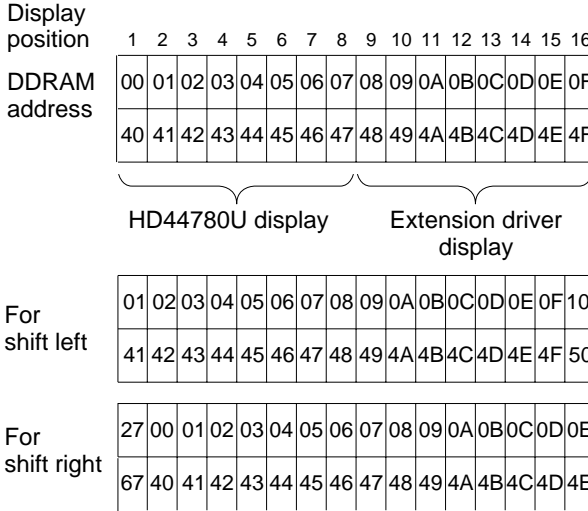


Figure 6 2-Line by 16-Character Display Example

Character Generator ROM (CGROM)

The character generator ROM generates 5×8 dot or 5×10 dot character patterns from 8-bit character codes (Table 4). It can generate 208 5×8 dot character patterns and 32 5×10 dot character patterns. User-defined character patterns are also available by mask-programmed ROM.

Character Generator RAM (CGRAM)

In the character generator RAM, the user can rewrite character patterns by program. For 5×8 dots, eight character patterns can be written, and for 5×10 dots, four character patterns can be written.

Write into DDRAM the character codes at the addresses shown as the left column of Table 4 to show the character patterns stored in CGRAM.

See Table 5 for the relationship between CGRAM addresses and data and display patterns.

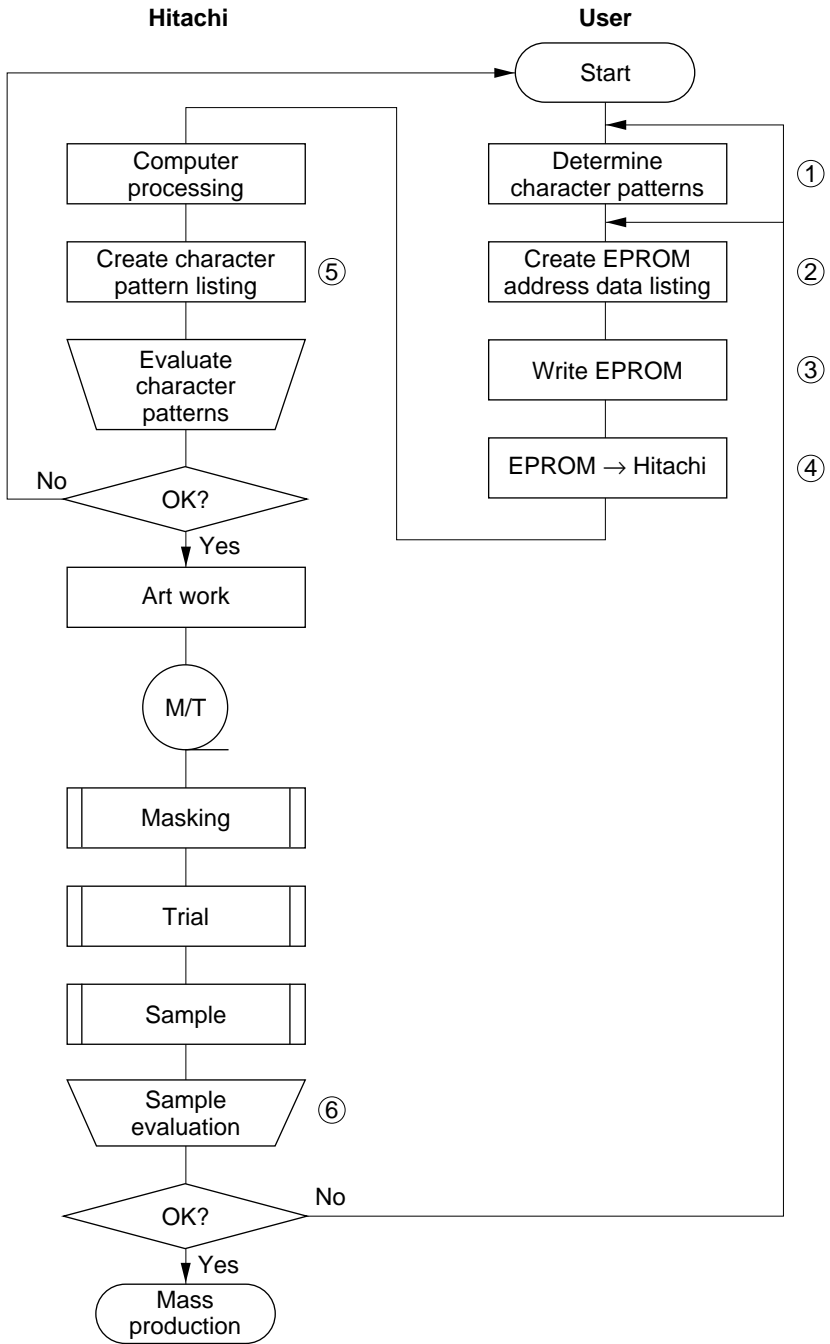
Areas that are not used for display can be used as general data RAM.

Modifying Character Patterns

- Character pattern development procedure

The following operations correspond to the numbers listed in Figure 7:

1. Determine the correspondence between character codes and character patterns.
2. Create a listing indicating the correspondence between EPROM addresses and data.
3. Program the character patterns into the EPROM.
4. Send the EPROM to Hitachi.
5. Computer processing on the EPROM is performed at Hitachi to create a character pattern listing, which is sent to the user.
6. If there are no problems within the character pattern listing, a trial LSI is created at Hitachi and samples are sent to the user for evaluation. When it is confirmed by the user that the character patterns are correctly written, mass production of the LSI proceeds at Hitachi.



Note: For a description of the numbers used in this figure, refer to the preceding page.

Figure 7 Character Pattern Development Procedure

• Programming character patterns

This section explains the correspondence between addresses and data used to program character patterns in EPROM. The HD44780U character generator ROM can generate 208 5 × 8 dot character patterns and 32 5 × 10 dot character patterns for a total of 240 different character patterns.

— Character patterns

EPROM address data and character pattern data correspond with each other to form a 5 × 8 or 5 × 10 dot character pattern (Tables 2 and 3).

Table 2 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 8 Dots)

EPROM Address										Data						
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	O0
								0	0	0	0	1	0	0	0	0
								0	0	0	1	1	0	0	0	0
								0	0	1	0	1	0	1	1	0
								0	0	1	1	1	1	0	0	1
								0	1	0	0	1	0	0	0	1
								0	1	0	1	1	0	0	0	1
								0	1	1	0	1	1	1	1	0
								0	1	1	1	0	0	0	0	0
								1	0	0	0	0	0	0	0	0
								1	0	0	1	0	0	0	0	0
								1	0	1	0	0	0	0	0	0
								1	0	1	1	0	0	0	0	0
								1	1	0	0	0	0	0	0	0
								1	1	0	1	0	0	0	0	0
								1	1	1	0	0	0	0	0	0
								1	1	1	1	0	0	0	0	0

← Cursor position

Character code

Line position

- Notes:
1. EPROM addresses A11 to A4 correspond to a character code.
 2. EPROM addresses A3 to A0 specify a line position of the character pattern.
 3. EPROM data O4 to O0 correspond to character pattern data.
 4. EPROM data O5 to O7 must be specified as 0.
 5. A lit display position (black) corresponds to a 1.
 6. Line 9 and the following lines must be blanked with 0s for a 5 × 8 dot character fonts.

— Handling unused character patterns

1. EPROM data outside the character pattern area: Always input 0s.
2. EPROM data in CGRAM area: Always input 0s. (Input 0s to EPROM addresses 00H to FFH.)
3. EPROM data used when the user does not use any HD44780U character pattern: According to the user application, handled in one of the two ways listed as follows.
 - a. When unused character patterns are not programmed: If an unused character code is written into DDRAM, all its dots are lit. By not programming a character pattern, all of its bits become lit. (This is due to the EPROM being filled with 1s after it is erased.)
 - b. When unused character patterns are programmed as 0s: Nothing is displayed even if unused character codes are written into DDRAM. (This is equivalent to a space.)

Table 3 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 10 Dots)

EPROM Address										Data						
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	O4	O3	O2	O1	O0
								0	0	0	0	0	0	0	0	0
								0	0	0	1	0	0	0	0	0
								0	0	1	0	0	1	1	0	1
								0	0	1	1	1	0	0	1	1
								0	1	0	0	1	0	0	0	1
								0	1	0	1	1	0	0	0	1
								0	1	1	0	0	1	1	1	1
0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1
								1	0	0	0	0	0	0	0	1
								1	0	0	1	0	0	0	0	1
								1	0	1	0	0	0	0	0	0
								1	0	1	1	0	0	0	0	0
								1	1	0	0	0	0	0	0	0
								1	1	0	1	0	0	0	0	0
								1	1	1	0	0	0	0	0	0
								1	1	1	1	0	0	0	0	0

← Cursor position

Character code Line position

- Notes:
1. EPROM addresses A11 to A3 correspond to a character code.
 2. EPROM addresses A3 to A0 specify a line position of the character pattern.
 3. EPROM data O4 to O0 correspond to character pattern data.
 4. EPROM data O5 to O7 must be specified as 0.
 5. A lit display position (black) corresponds to a 1.
 6. Line 11 and the following lines must be blanked with 0s for a 5 × 10 dot character fonts.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	a	P	`	P				-	9	E	o	p	
xxxx0001	(2)		!	1	A	Q	a	9				.	7	4	ä	q	
xxxx0010	(3)		"	2	B	R	b	r				Γ	イ	ツ	β	θ	
xxxx0011	(4)		#	3	C	S	c	s				┘	ウ	テ	ε	ω	
xxxx0100	(5)		\$	4	D	T	d	t				、	エ	ト	μ	Ω	
xxxx0101	(6)		%	5	E	U	e	u				.	オ	ナ	ι	ü	
xxxx0110	(7)		&	6	F	V	f	v				ヲ	カ	ニ	ρ	Σ	
xxxx0111	(8)		'	7	G	W	g	w				フ	チ	ヌ	g	π	
xxxx1000	(1)		(8	H	X	h	x				イ	ウ	ネ	Γ	×	
xxxx1001	(2))	9	I	Y	i	y				ウ	ク	ル	ι	υ	
xxxx1010	(3)		*	:	J	Z	j	z				エ	コ	ソ	┘	┘	
xxxx1011	(4)		+	;	K	C	k	c				オ	サ	ヒ	□	×	π
xxxx1100	(5)		,	<	L	*	l	l				ホ	シ	フ	フ	φ	π
xxxx1101	(6)		-	=	M	J	m	j				ユ	ズ	ソ	┘	┘	÷
xxxx1110	(7)		.	>	N	^	n	→				ヨ	セ	ホ	ソ	π	
xxxx1111	(8)		/	?	O	_	o	←				ウ	ソ	マ	°	ö	

Note: The user can specify any pattern for character-generator RAM.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A02)

Lower 4 Bits \ Upper 4 Bits		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000		CG RAM (1)	⬆	⊖	Ⓜ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	ⓐ
	(2)	⬇	!	1	A	Q	a	9	A	♪	i	±	Á	Ñ	á	ñ	
xxxx0010	(3)	“	"	2	B	R	b	r	W	Γ	¢	²	À	Ó	à	ó	
	(4)	”	#	3	C	S	c	s	3	π	€	³	Ã	Ö	ã	ö	
xxxx0100	(5)	⬆	\$	4	D	T	d	t	M	Σ	×	℔	Ä	Ö	ä	ö	
	(6)	⬇	%	5	E	U	e	u	Ń	σ	¥	℥	Å	Ö	å	ö	
xxxx0110	(7)	⬆	&	6	F	V	f	v	J	¼	¦	¶	Ë	Ö	ë	ö	
	(8)	⬇	'	7	G	W	g	w	Π	∞	•	Ç	×	ç	÷		
xxxx1000	(1)	↑	(8	H	X	h	x	Y	♯	♠	ω	É	È	é	è	
	(2)	↓)	9	I	Y	i	y	U	⊖	⊕	¹	É	Ù	é	ù	
xxxx1010	(3)	÷	*	:	J	Z	j	z	4	⊖	⊕	⊖	É	Ù	é	ù	
	(4)	←	+	;	K	[k	[w	δ	⊗	⊗	È	Ö	è	ö	
xxxx1100	(5)	Σ	,	<	L	\	l	l	W	⊗	W	⊗	ì	ü	í	ü	
	(6)	Σ	-	=	M]	m]	b	⬆	Ⓢ	Ⓢ	ì	ý	í	ÿ	
xxxx1110	(7)	⬆	.	>	N	^	n	~	W	ε	Ⓢ	Ⓢ	ì	þ	í	þ	
	(8)	⬇	/	?	O	_	o	ó	Ⓢ	Ⓢ	°	¿	ì	ß	í	ÿ	

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

For 5 × 8 dot character patterns

Character Codes (DDRAM data)		CGRAM Address		Character Patterns (CGRAM data)						
7	6	5	4	3	2	1	0			
High			Low			High		Low		
0 0 0 0 * 0 0 0		0 0 0		0	0	0	*	*	*	
				0	0	1	1	1	1	0
				0	1	0	0	0	0	1
				0	1	0	0	0	0	1
				0	1	1	1	1	1	0
				1	0	1	0	1	0	0
				1	0	1	0	1	0	0
				1	1	1	1	1	1	0
0 0 0 0 * 0 0 1		0 0 1		0	0	0	*	*	*	
				0	0	1	1	1	1	0
				0	1	0	0	1	0	1
				0	1	0	1	1	1	1
				0	0	1	0	1	0	0
				1	1	1	1	1	1	1
				0	0	1	0	0	1	0
				0	0	1	0	0	0	0
0 0 0 0 * 1 1 1		1 1 1		0	0	0	*	*	*	
				1	0	0	1	1	1	1
				1	0	1	1	1	1	1
				1	1	1	1	1	1	1

- Notes:
- Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
 - CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logical OR with the cursor. Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display. If the 8th line data is 1, 1 bits will light up the 8th line regardless of the cursor presence.
 - Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
 - As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
 - 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- * Indicates no effect.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data) (cont)

For 5 × 10 dot character patterns

Character Codes (DDRAM data)		CGRAM Address		Character Patterns (CGRAM data)			
7 6 5 4 3 2 1 0		5 4 3 2 1 0		7 6 5 4 3 2 1 0			
High Low		High Low		High Low			
0 0 0 0 * 0 0 *		0 0	0 0 0 0	* * *	0 0 0 0 0	} Character pattern	
			0 0 0 1	↑	0 0 0 0 0		
			0 0 1 0		1 0 1 1 0		
			0 0 1 1		1 1 0 0 1		
			0 1 0 0		1 0 0 0 1		
			0 1 0 1		1 0 0 0 1		
			0 1 1 0		1 1 1 1 0		
			0 1 1 1		1 0 0 0 0		
			1 0 0 0		1 0 0 0 0		
			1 0 0 1		1 0 0 0 0		
			1 0 1 0		1 0 0 0 0		} Cursor position
1 0 1 1	*	* * *	* * * * *				
1 1 0 0	↑		↑				
1 1 0 1	↓		↓				
1 1 1 0		*	* * * * *				
1 1 1 1		*	* * * * *				
0 0 0 0		0 0 0 0					
0 0 0 1	↑						
0 0 0 0 * 1 1 *		1 1	1 0 0 1	↓	* * *		
			1 0 1 0		* * *		
			1 0 1 1	↑	* * *	* * * * *	
			1 1 0 0	↓		↑	
			1 1 0 1			↓	
1 1 1 0							
1 1 1 1		* * *	* * * * *				

- Notes:
- Character code bits 1 and 2 correspond to CGRAM address bits 4 and 5 (2 bits: 4 types).
 - CGRAM address bits 0 to 3 designate the character pattern line position. The 11th line is the cursor position and its display is formed by a logical OR with the cursor.
Maintain the 11th line data corresponding to the cursor display position at 0 as the cursor display. If the 11th line data is "1", "1" bits will light up the 11th line regardless of the cursor presence. Since lines 12 to 16 are not used for display, they can be used for general data RAM.
 - Character pattern row positions are the same as 5 × 8 dot character pattern positions.
 - CGRAM character patterns are selected when character code bits 4 to 7 are all 0.
However, since character code bits 0 and 3 have no effect, the P display example above can be selected by character codes 00H, 01H, 08H, and 09H.
 - 1 for CGRAM data corresponds to display selection and 0 to non-selection.
- * Indicates no effect.

Timing Generation Circuit

The timing generation circuit generates timing signals for the operation of internal circuits such as DDRAM, CGROM and CGRAM. RAM read timing for display and internal operation timing by MPU access are generated separately to avoid interfering with each other. Therefore, when writing data to DDRAM, for example, there will be no undesirable interferences, such as flickering, in areas other than the display area.

Liquid Crystal Display Driver Circuit

The liquid crystal display driver circuit consists of 16 common signal drivers and 40 segment signal drivers. When the character font and number of lines are selected by a program, the required common signal drivers automatically output drive waveforms, while the other common signal drivers continue to output non-selection waveforms.

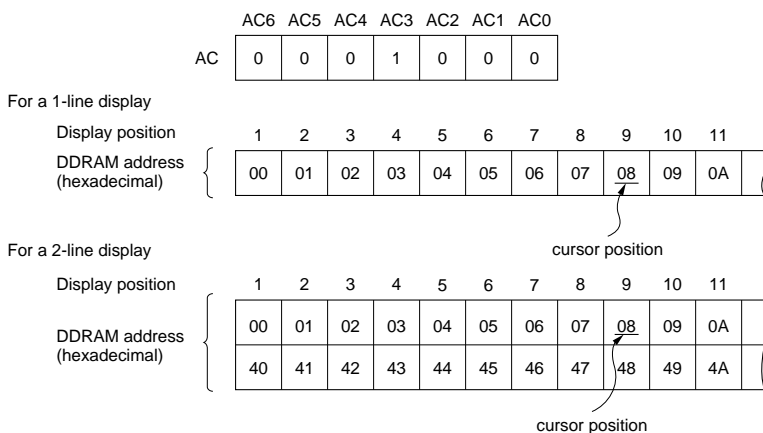
Sending serial data always starts at the display data character pattern corresponding to the last address of the display data RAM (DDRAM).

Since serial data is latched when the display data character pattern corresponding to the starting address enters the internal shift register, the HD44780U drives from the head display.

Cursor/Blink Control Circuit

The cursor/blink control circuit generates the cursor or character blinking. The cursor or the blinking will appear with the digit located at the display data RAM (DDRAM) address set in the address counter (AC).

For example (Figure 8), when the address counter is 08H, the cursor position is displayed at DDRAM address 08H.



Note: The cursor or blinking appears when the address counter (AC) selects the character generator RAM (CGRAM). However, the cursor and blinking become meaningless. The cursor or blinking is displayed in the meaningless position when the AC is a CGRAM address.

Figure 8 Cursor/Blink Display Example

Interfacing to the MPU

The HD44780U can send data in either two 4-bit operations or one 8-bit operation, thus allowing interfacing with 4- or 8-bit MPUs.

- For 4-bit interface data, only four bus lines (DB4 to DB7) are used for transfer. Bus lines DB0 to DB3 are disabled. The data transfer between the HD44780U and the MPU is completed after the 4-bit data has been transferred twice. As for the order of data transfer, the four high order bits (for 8-bit operation, DB4 to DB7) are transferred before the four low order bits (for 8-bit operation, DB0 to DB3).

The busy flag must be checked (one instruction) after the 4-bit data has been transferred twice. Two more 4-bit operations then transfer the busy flag and address counter data.

- For 8-bit interface data, all eight bus lines (DB0 to DB7) are used.

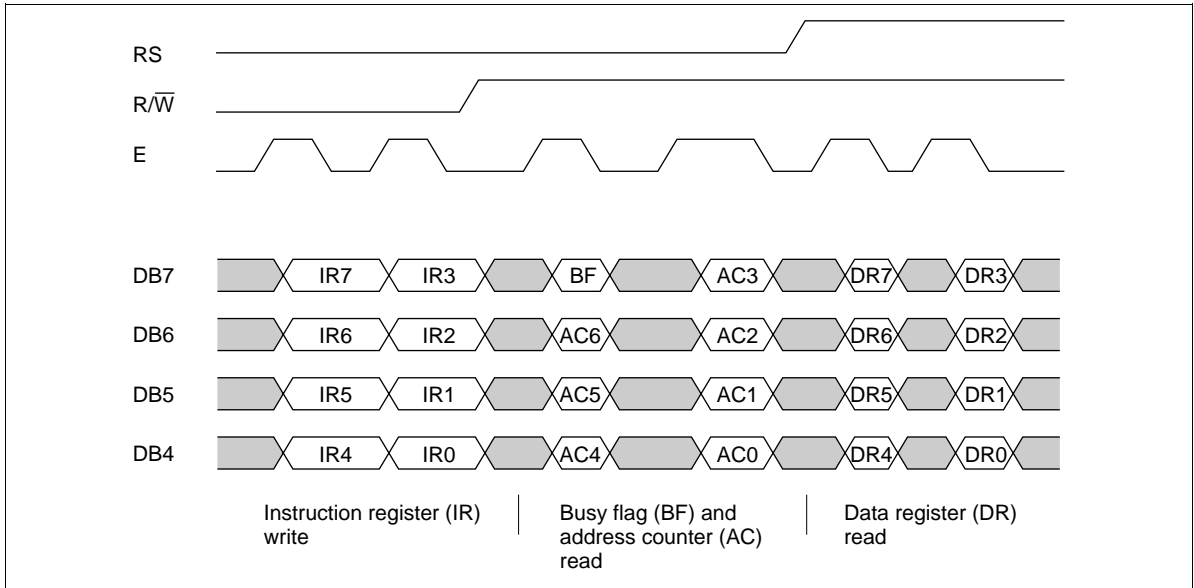


Figure 9 4-Bit Transfer Example

Reset Function

Initializing by Internal Reset Circuit

An internal reset circuit automatically initializes the HD44780U when the power is turned on. The following instructions are executed during the initialization. The busy flag (BF) is kept in the busy state until the initialization ends (BF = 1). The busy state lasts for 10 ms after V_{CC} rises to 4.5 V.

1. Display clear
2. Function set:
 - DL = 1; 8-bit interface data
 - N = 0; 1-line display
 - F = 0; 5 × 8 dot character font
3. Display on/off control:
 - D = 0; Display off
 - C = 0; Cursor off
 - B = 0; Blinking off
4. Entry mode set:
 - I/D = 1; Increment by 1
 - S = 0; No shift

Note: If the electrical characteristics conditions listed under the table Power Supply Conditions Using Internal Reset Circuit are not met, the internal reset circuit will not operate normally and will fail to initialize the HD44780U. For such a case, initialization must be performed by the MPU as explained in the section, Initializing by Instruction.

Instructions

Outline

Only the instruction register (IR) and the data register (DR) of the HD44780U can be controlled by the MPU. Before starting the internal operation of the HD44780U, control information is temporarily stored into these registers to allow interfacing with various MPUs, which operate at different speeds, or various peripheral control devices. The internal operation of the HD44780U is determined by signals sent from the MPU. These signals, which include register selection signal (RS), read/

write signal (R/\overline{W}), and the data bus (DB0 to DB7), make up the HD44780U instructions (Table 6). There are four categories of instructions that:

- Designate HD44780U functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions

Normally, instructions that perform data transfer with internal RAM are used the most. However, auto-incrementation by 1 (or auto-decrementation by 1) of internal HD44780U RAM addresses after each data write can lighten the program load of the MPU. Since the display shift instruction (Table 11) can perform concurrently with display data write, the user can minimize system development time with maximum programming efficiency.

When an instruction is being executed for internal operation, no instruction other than the busy flag/address read instruction can be executed.

Because the busy flag is set to 1 while an instruction is being executed, check it to make sure it is 0 before sending another instruction from the MPU.

Note: Be sure the HD44780U is not in the busy state (BF = 0) before sending an instruction from the MPU to the HD44780U. If an instruction is sent without checking the busy flag, the time between the first instruction and next instruction will take much longer than the instruction time itself. Refer to Table 6 for the list of each instruction execution time.

Table 6 Instructions

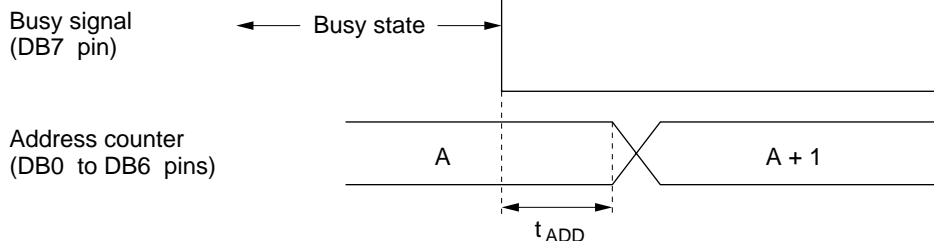
Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 μ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 μ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 μ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 μ s
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 μ s
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μ s

Table 6 Instructions (cont)

Instruction	Code										Description	Execution Time (max) (when f_{cp} or f_{OSC} is 270 kHz)						
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0								
Write data to CG or DDRAM	1	0	Write data										Writes data into DDRAM or CGRAM.	$37 \mu s$ $t_{ADD} = 4 \mu s^*$				
Read data from CG or DDRAM	1	1	Read data										Reads data from DDRAM or CGRAM.	$37 \mu s$ $t_{ADD} = 4 \mu s^*$				
			I/D = 1:	Increment												DDRAM: Display data RAM	Execution time changes when frequency changes Example: When f_{cp} or f_{OSC} is 250 kHz, $37 \mu s \times \frac{270}{250} = 40 \mu s$	
			I/D = 0:	Decrement												CGRAM: Character generator RAM		
			S = 1:	Accompanies display shift												ACG: CGRAM address		
			S/C = 1:	Display shift												ADD: DDRAM address		
			S/C = 0:	Cursor move												(corresponds to cursor address)		
			R/L = 1:	Shift to the right												AC: Address counter used for both DD and CGRAM addresses		
			R/L = 0:	Shift to the left														
			DL = 1:	8 bits, DL = 0: 4 bits														
			N = 1:	2 lines, N = 0: 1 line														
			F = 1:	5×10 dots, F = 0: 5×8 dots														
			BF = 1:	Internally operating														
			BF = 0:	Instructions acceptable														

Note: — indicates no effect.

- * After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, t_{ADD} is the time elapsed after the busy flag turns off until the address counter is updated.



Note: t_{ADD} depends on the operation frequency
 $t_{ADD} = 1.5/(f_{cp} \text{ or } f_{OSC})$ seconds

Figure 10 Address Counter Update

Instruction Description

Clear Display

Clear display writes space code 20H (character pattern for character code 20H must be a blank pattern) into all DDRAM addresses. It then sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. In other words, the display disappears and the cursor or blinking goes to the left edge of the display (in the first line if 2 lines are displayed). It also sets I/D to 1 (increment mode) in entry mode. S of entry mode does not change.

Return Home

Return home sets DDRAM address 0 into the address counter, and returns the display to its original status if it was shifted. The DDRAM contents do not change.

The cursor or blinking go to the left edge of the display (in the first line if 2 lines are displayed).

Entry Mode Set

I/D: Increments ($I/D = 1$) or decrements ($I/D = 0$) the DDRAM address by 1 when a character code is written into or read from DDRAM.

The cursor or blinking moves to the right when incremented by 1 and to the left when decremented by 1. The same applies to writing and reading of CGRAM.

S: Shifts the entire display either to the right ($I/D = 0$) or to the left ($I/D = 1$) when S is 1. The display does not shift if S is 0.

If S is 1, it will seem as if the cursor does not move but the display does. The display does not shift when reading from DDRAM. Also, writing into or reading out from CGRAM does not shift the display.

Display On/Off Control

D: The display is on when D is 1 and off when D is 0. When off, the display data remains in DDRAM, but can be displayed instantly by setting D to 1.

C: The cursor is displayed when C is 1 and not displayed when C is 0. Even if the cursor disappears, the function of I/D or other specifications will not change during display data write. The cursor is displayed using 5 dots in the 8th line for 5×8 dot character font selection and in the 11th line for the 5×10 dot character font selection (Figure 13).

B: The character indicated by the cursor blinks when B is 1 (Figure 13). The blinking is displayed as switching between all blank dots and displayed characters at a speed of 409.6-ms intervals when f_{cp} or f_{osc} is 250 kHz. The cursor and blinking can be set to display simultaneously. (The blinking frequency changes according to f_{osc} or the reciprocal of f_{cp} . For example, when f_{cp} is 270 kHz, $409.6 \times 250/270 = 379.2$ ms.)

Cursor or Display Shift

Cursor or display shift shifts the cursor position or display to the right or left without writing or reading display data (Table 7). This function is used to correct or search the display. In a 2-line display, the cursor moves to the second line when it passes the 40th digit of the first line. Note that the first and second line displays will shift at the same time.

When the displayed data is shifted repeatedly each line moves only horizontally. The second line display does not shift into the first line position.

The address counter (AC) contents will not change if the only action performed is a display shift.

Function Set

DL: Sets the interface data length. Data is sent or received in 8-bit lengths (DB7 to DB0) when DL is 1, and in 4-bit lengths (DB7 to DB4) when DL is 0. When 4-bit length is selected, data must be sent or received twice.

N: Sets the number of display lines.

F: Sets the character font.

Note: Perform the function at the head of the program before executing any instructions (except for the read busy flag and address instruction). From this point, the function set instruction cannot be executed unless the interface data length is changed.

Set CGRAM Address

Set CGRAM address sets the CGRAM address binary AAAAAA into the address counter.

Data is then written to or read from the MPU for CGRAM.

		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Clear display	Code	0	0	0	0	0	0	0	0	0	1	
Return home	Code	0	0	0	0	0	0	0	0	1	*	Note: * Don't care.
Entry mode set	Code	0	0	0	0	0	0	0	1	I/D	S	
Display on/off control	Code	0	0	0	0	0	0	1	D	C	B	
Cursor or display shift	Code	0	0	0	0	0	1	S/C	R/L	*	*	Note: * Don't care.
Function set	Code	0	0	0	0	1	DL	N	F	*	*	
Set CGRAM address	Code	0	0	0	1	A	A	A	A	A	A	

← Higher order bit Lower order bit →

Figure 11 Instruction (1)

Set DDRAM Address

Set DDRAM address sets the DDRAM address binary AAAAAAA into the address counter.

Data is then written to or read from the MPU for DDRAM.

However, when N is 0 (1-line display), AAAAAAA can be 00H to 4FH. When N is 1 (2-line display), AAAAAAA can be 00H to 27H for the first line, and 40H to 67H for the second line.

Read Busy Flag and Address

Read busy flag and address reads the busy flag (BF) indicating that the system is now internally operating on a previously received instruction. If BF is 1, the internal operation is in progress. The next instruction will not be accepted until BF is reset to 0. Check the BF status before the next write operation. At the same time, the value of the address counter in binary AAAAAAA is read out. This address counter is used by both CG and DDRAM addresses, and its value is determined by the previous instruction. The address contents are the same as for instructions set CGRAM address and set DDRAM address.

Table 7 Shift Function

S/C	R/L	
0	0	Shifts the cursor position to the left. (AC is decremented by one.)
0	1	Shifts the cursor position to the right. (AC is incremented by one.)
1	0	Shifts the entire display to the left. The cursor follows the display shift.
1	1	Shifts the entire display to the right. The cursor follows the display shift.

Table 8 Function Set

N	F	No. of Display Lines	Character Font	Duty Factor	Remarks
0	0	1	5 × 8 dots	1/8	
0	1	1	5 × 10 dots	1/11	
1	*	2	5 × 8 dots	1/16	Cannot display two lines for 5 × 10 dot character font

Note: * Indicates don't care.

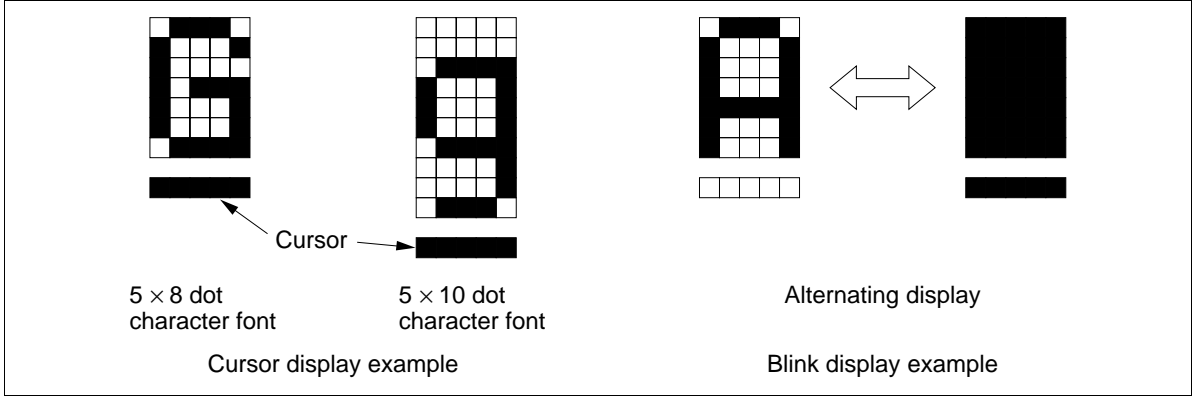


Figure 12 Cursor and Blinking

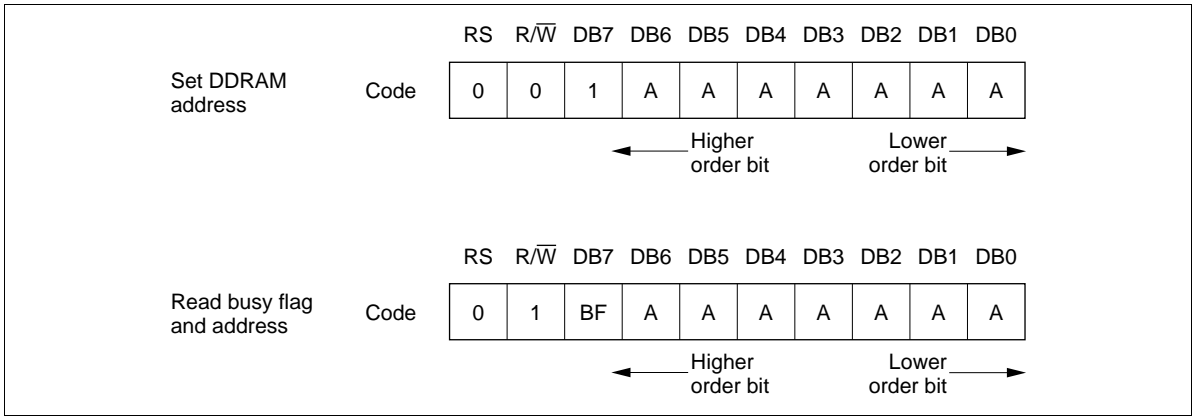


Figure 13 Instruction (2)

Write Data to CG or DDRAM

Write data to CG or DDRAM writes 8-bit binary data DDDDDDDD to CG or DDRAM.

To write into CG or DDRAM is determined by the previous specification of the CGRAM or DDRAM address setting. After a write, the address is automatically incremented or decremented by 1 according to the entry mode. The entry mode also determines the display shift.

Read Data from CG or DDRAM

Read data from CG or DDRAM reads 8-bit binary data DDDDDDDD from CG or DDRAM.

The previous designation determines whether CG or DDRAM is to be read. Before entering this read instruction, either CGRAM or DDRAM address set instruction must be executed. If not executed, the first read data will be invalid. When serially executing read instructions, the next address data is normally read from the second read. The address set instructions need not be executed just before this read instruction when shifting the cursor by the cursor shift instruction (when reading out DDRAM). The operation of the cursor shift instruction is the same as the set DDRAM address instruction.

After a read, the entry mode automatically increases or decreases the address by 1. However, display shift is not executed regardless of the entry mode.

Note: The address counter (AC) is automatically incremented or decremented by 1 after the write instructions to CGRAM or DDRAM are executed. The RAM data selected by the AC cannot be read out at this time even if read instructions are executed. Therefore, to correctly read data, execute either the address set instruction or cursor shift instruction (only with DDRAM), then just before reading the desired data, execute the read instruction from the second time the read instruction is sent.

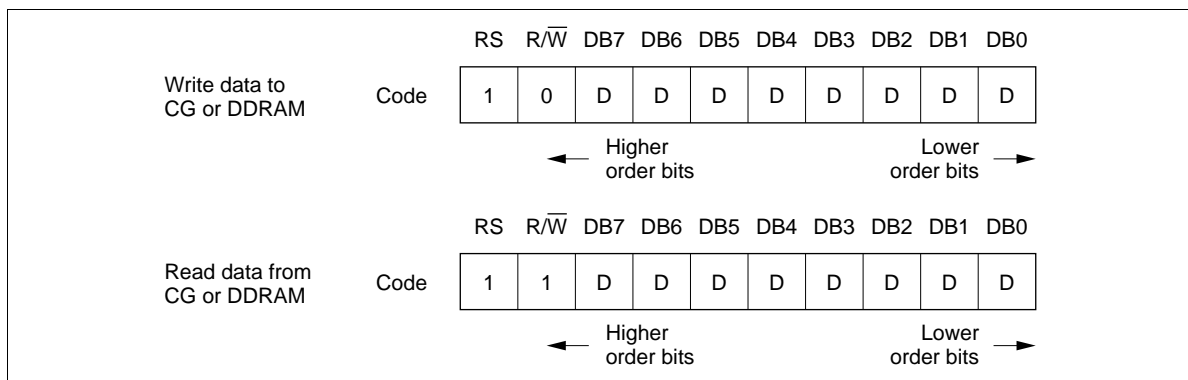


Figure 14 Instruction (3)

Interfacing the HD44780U

Interface to MPUs

- Interfacing to an 8-bit MPU

See Figure 16 for an example of using a I/O port (for a single-chip microcomputer) as an interface device.

In this example, P30 to P37 are connected to the data bus DB0 to DB7, and P75 to P77 are connected to E, R/W, and RS, respectively.

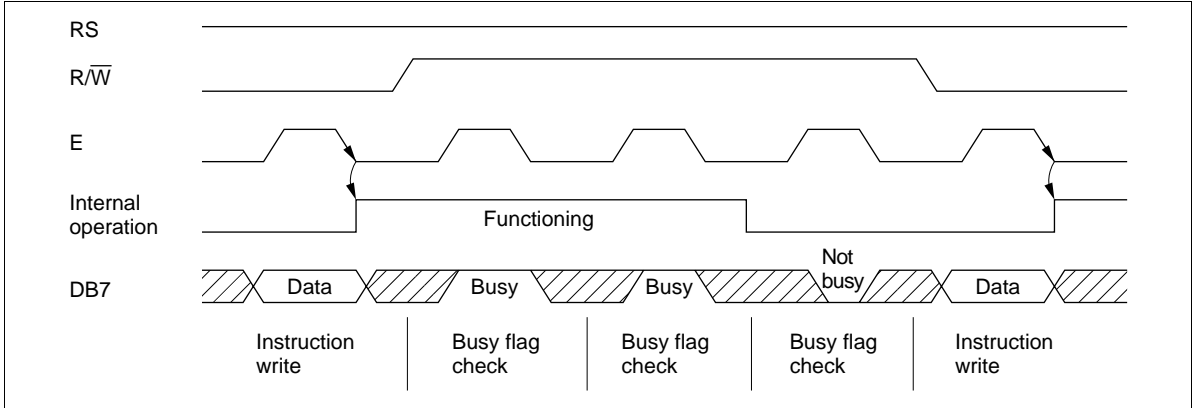


Figure 15 Example of Busy Flag Check Timing Sequence

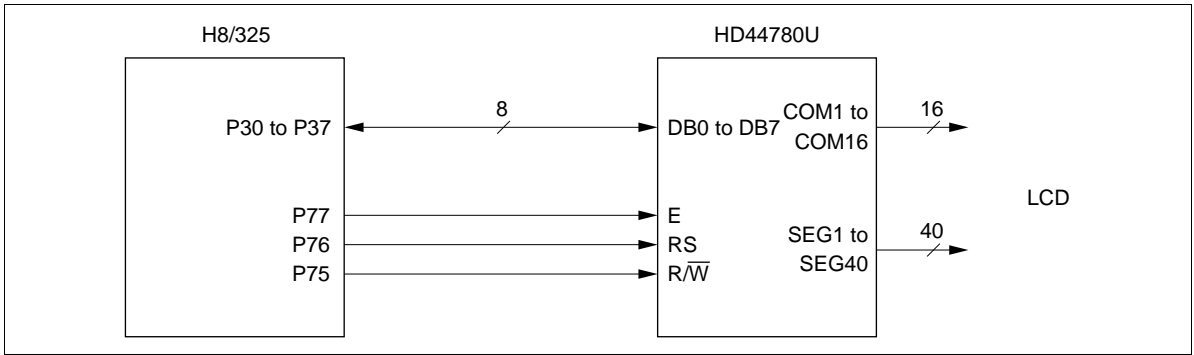


Figure 16 H8/325 Interface (Single-Chip Mode)

- Interfacing to a 4-bit MPU

The HD44780U can be connected to the I/O port of a 4-bit MPU. If the I/O port has enough bits, 8-bit data can be transferred. Otherwise, one data transfer must be made in two operations for 4-bit data. In this case, the timing sequence becomes somewhat complex. (See Figure 17.)

See Figure 18 for an interface example to the HMCS4019R.

Note that two cycles are needed for the busy flag check as well as for the data transfer. The 4-bit operation is selected by the program.

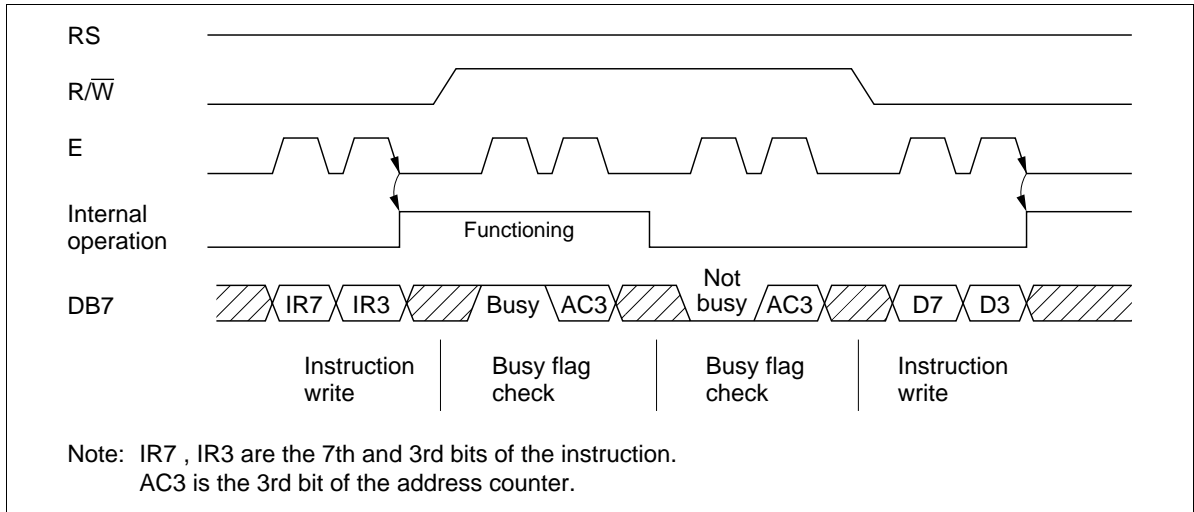


Figure 17 Example of 4-Bit Data Transfer Timing Sequence

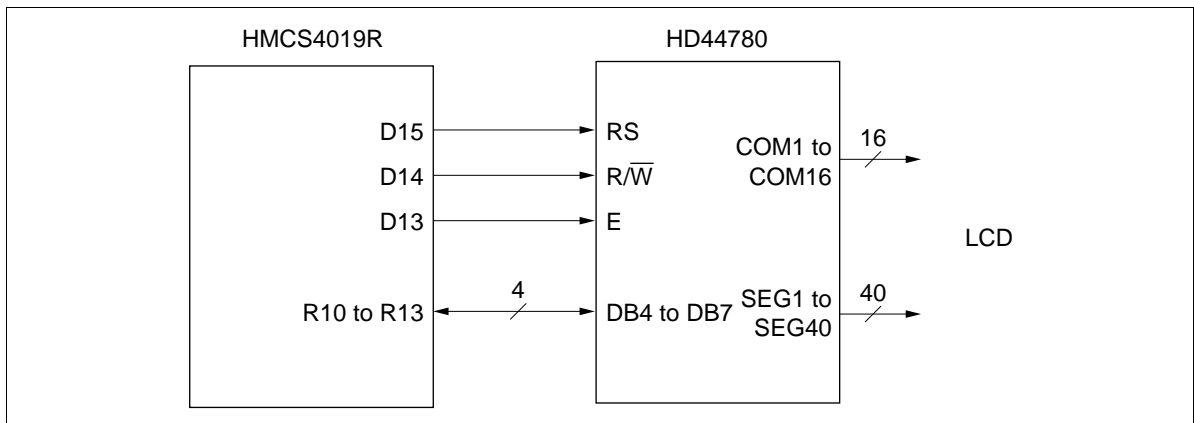


Figure 18 Example of Interface to HMCS4019R

Interface to Liquid Crystal Display

Character Font and Number of Lines: The HD44780U can perform two types of displays, 5×8 dot and 5×10 dot character fonts, each with a cursor.

Up to two lines are displayed for 5×8 dots and one line for 5×10 dots. Therefore, a total of three types of common signals are available (Table 9).

The number of lines and font types can be selected by the program. (See Table 6, Instructions.)

Connection to HD44780 and Liquid Crystal Display: See Figure 19 for the connection examples.

Table 9 Common Signals

Number of Lines	Character Font	Number of Common Signals	Duty Factor
1	5×8 dots + cursor	8	1/8
1	5×10 dots + cursor	11	1/11
2	5×8 dots + cursor	16	1/16

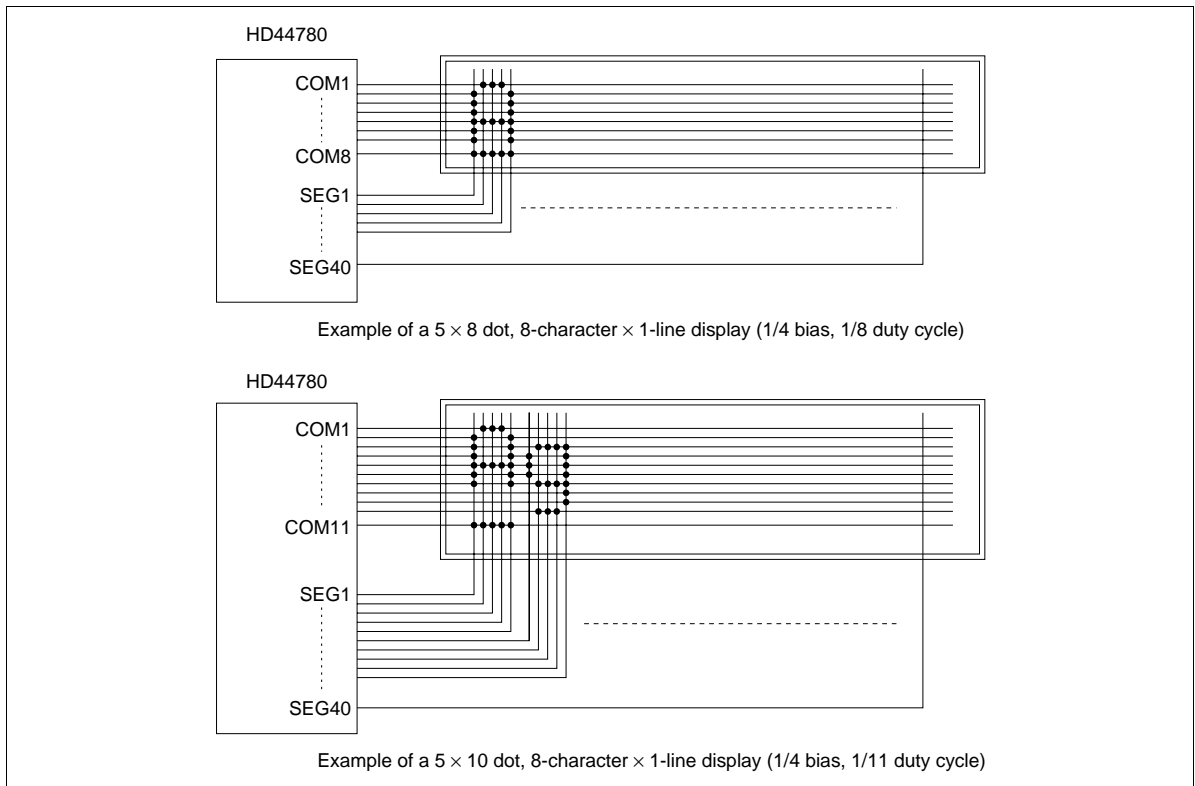


Figure 19 Liquid Crystal Display and HD44780 Connections

Since five segment signal lines can display one digit, one HD44780U can display up to 8 digits for a 1-line display and 16 digits for a 2-line display.

The examples in Figure 19 have unused common signal pins, which always output non-selection waveforms. When the liquid crystal display panel has unused extra scanning lines, connect the extra scanning lines to these common signal pins to avoid any undesirable effects due to crosstalk during the floating state.

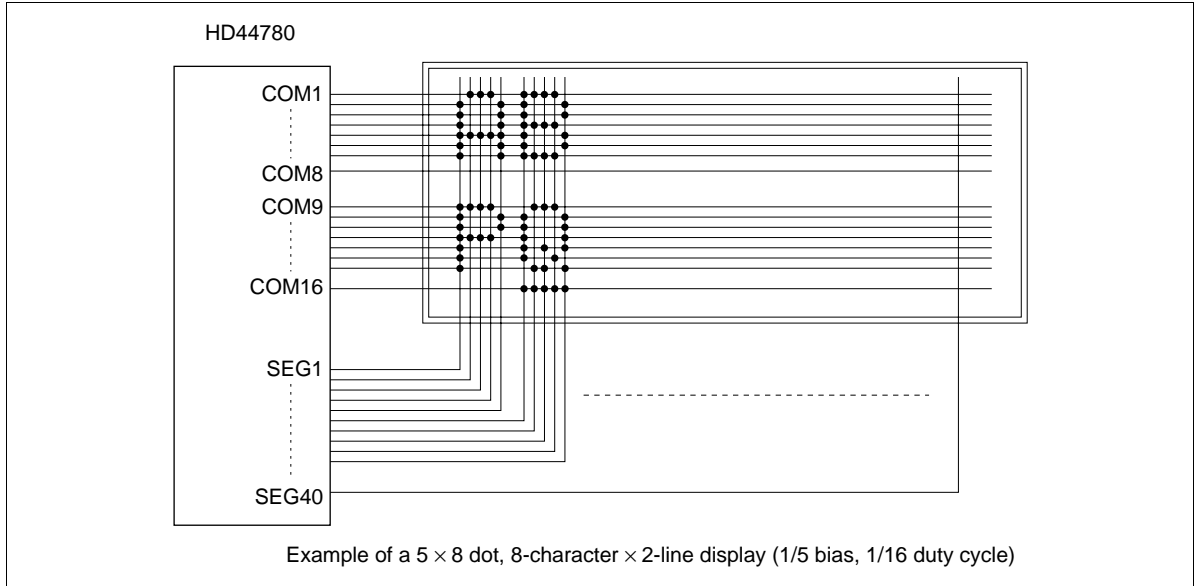


Figure 19 Liquid Crystal Display and HD44780 Connections (cont)

Connection of Changed Matrix Layout: In the preceding examples, the number of lines correspond to the scanning lines. However, the following display examples (Figure 20) are made possible by altering the matrix layout of the liquid crystal display panel. In either case, the only change is the layout. The display characteristics and the number of liquid crystal display characters depend on the number of common signals or on duty factor. Note that the display data RAM (DDRAM) addresses for 4 characters \times 2 lines and for 16 characters \times 1 line are the same as in Figure 19.

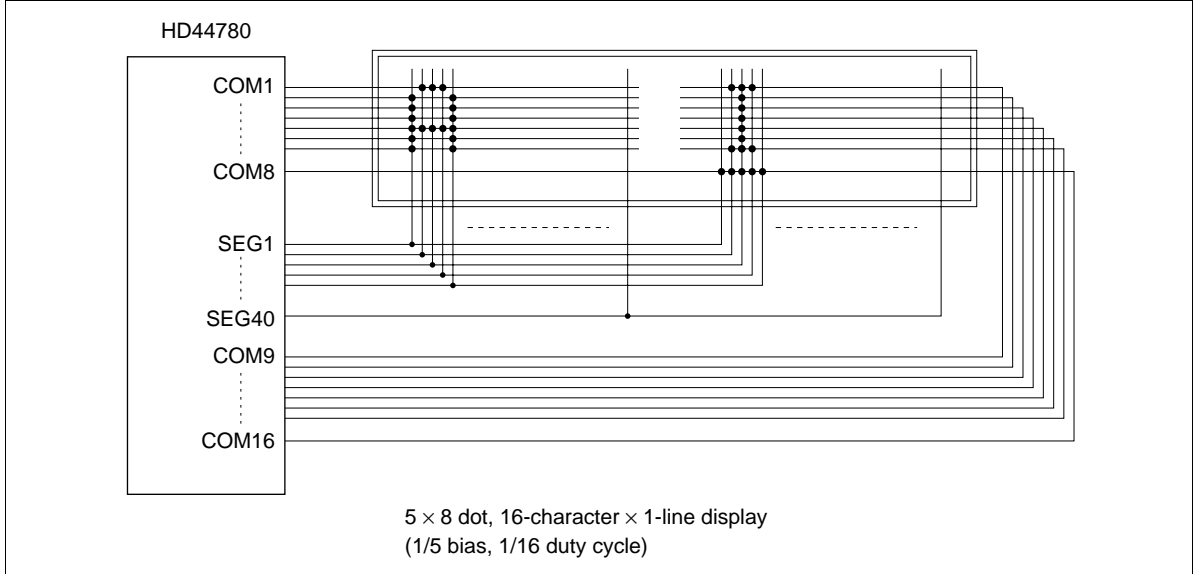


Figure 20 Changed Matrix Layout Displays

Power Supply for Liquid Crystal Display Drive

Various voltage levels must be applied to pins V1 to V5 of the HD44780U to obtain the liquid crystal display drive waveforms. The voltages must be changed according to the duty factor (Table 10).

VLCD is the peak value for the liquid crystal display drive waveforms, and resistance dividing provides voltages V1 to V5 (Figure 21).

Table 10 Duty Factor and Power Supply for Liquid Crystal Display Drive

Power Supply	Duty Factor	
	1/8, 1/11	1/16
	Bias	
	1/4	1/5
V1	$V_{CC}-1/4 \text{ VLCD}$	$V_{CC}-1/5 \text{ VLCD}$
V2	$V_{CC}-1/2 \text{ VLCD}$	$V_{CC}-2/5 \text{ VLCD}$
V3	$V_{CC}-1/2 \text{ VLCD}$	$V_{CC}-3/5 \text{ VLCD}$
V4	$V_{CC}-3/4 \text{ VLCD}$	$V_{CC}-4/5 \text{ VLCD}$
V5	$V_{CC}-\text{VLCD}$	$V_{CC}-\text{VLCD}$

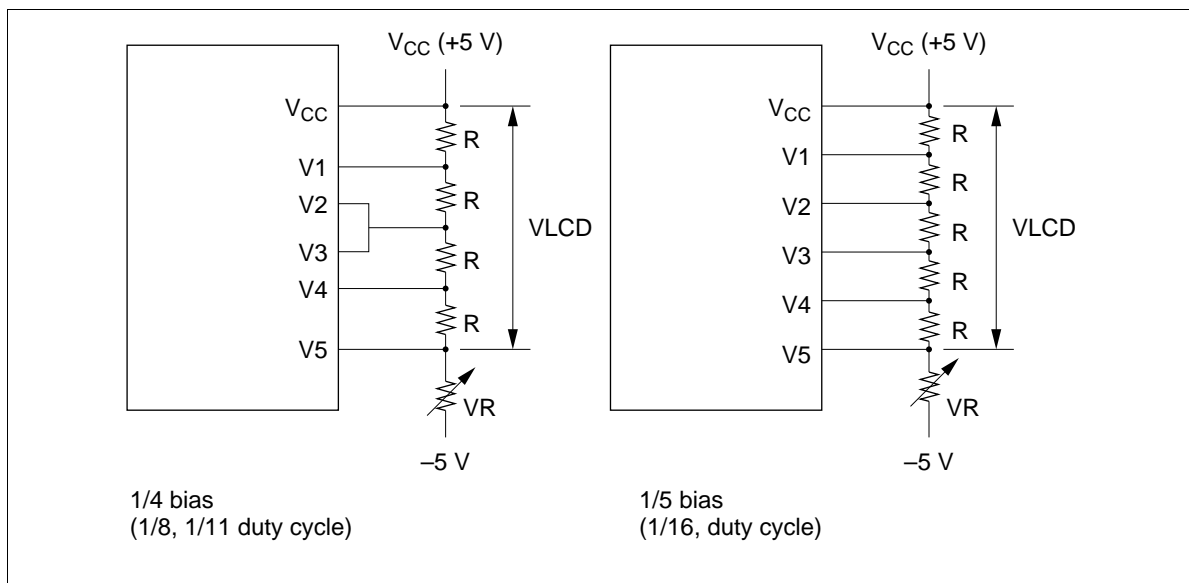
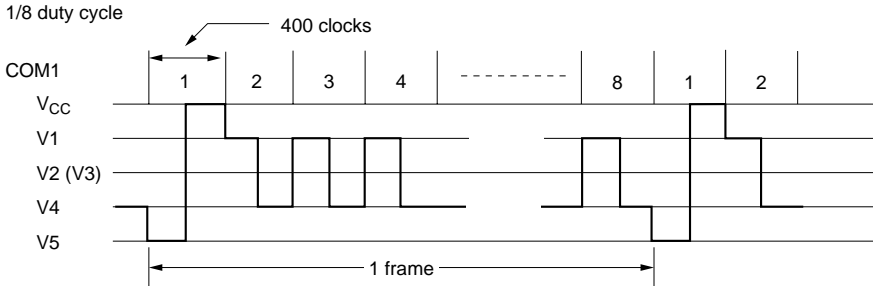


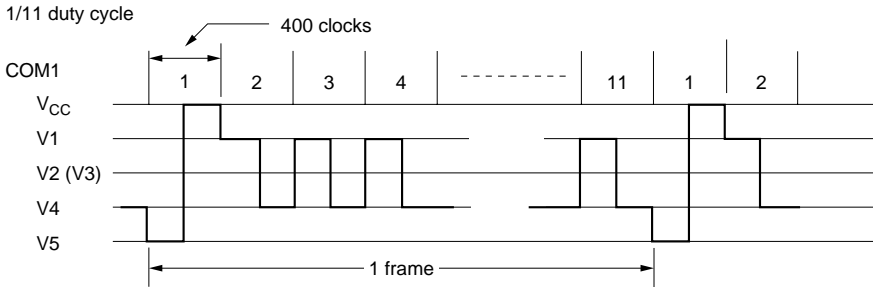
Figure 21 Drive Voltage Supply Example

Relationship between Oscillation Frequency and Liquid Crystal Display Frame Frequency

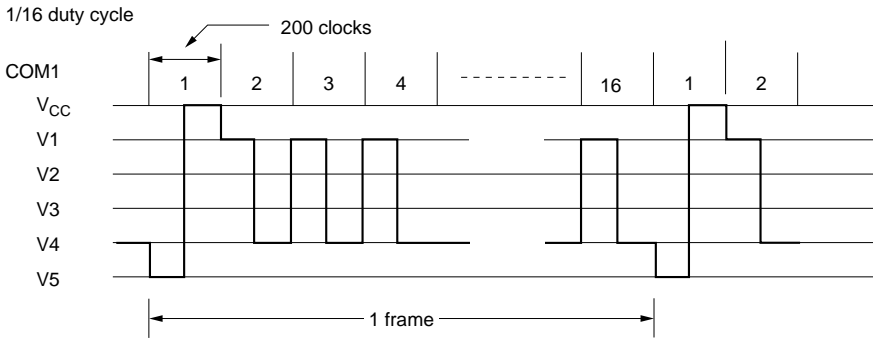
The liquid crystal display frame frequencies of Figure 22 apply only when the oscillation frequency is 270 kHz (one clock pulse of 3.7 μs).



1 frame = $3.7 \mu\text{s} \times 400 \times 8 = 11850 \mu\text{s} = 11.9 \text{ ms}$
 Frame frequency = $\frac{1}{11.9 \text{ ms}} = 84.3 \text{ Hz}$



1 frame = $3.7 \mu\text{s} \times 400 \times 11 = 16300 \mu\text{s} = 16.3 \text{ ms}$
 Frame frequency = $\frac{1}{16.3 \text{ ms}} = 61.4 \text{ Hz}$



1 frame = $3.7 \mu\text{s} \times 200 \times 16 = 11850 \mu\text{s} = 11.9 \text{ ms}$
 Frame frequency = $\frac{1}{11.9 \text{ ms}} = 84.3 \text{ Hz}$

Figure 22 Frame Frequency

Instruction and Display Correspondence

- 8-bit operation, 8-digit × 1-line display with internal reset

Refer to Table 11 for an example of an 8-digit × 1-line display in 8-bit operation. The HD44780U functions must be set by the function set instruction prior to the display. Since the display data RAM can store data for 80 characters, as explained before, the RAM can be used for displays such as for advertising when combined with the display shift operation.

Since the display shift operation changes only the display position with DDRAM contents unchanged, the first display data entered into DDRAM can be output when the return home operation is performed.

- 4-bit operation, 8-digit × 1-line display with internal reset

The program must set all functions prior to the 4-bit operation (Table 12). When the power is turned on, 8-bit operation is automatically selected and the first write is performed as an 8-bit operation. Since DB0 to DB3 are not connected, a rewrite is then required. However, since one operation is completed in two accesses for 4-bit operation, a rewrite is needed to set the functions (see Table 12). Thus, DB4 to DB7 of the function set instruction is written twice.

- 8-bit operation, 8-digit × 2-line display

For a 2-line display, the cursor automatically moves from the first to the second line after the 40th digit of the first line has been written. Thus, if there are only 8 characters in the first line, the DDRAM address must be again set after the 8th character is completed. (See Table 13.) Note that the display shift operation is performed for the first and second lines. In the example of Table 13, the display shift is performed when the cursor is on the second line. However, if the shift operation is performed when the cursor is on the first line, both the first and second lines move together. If the shift is repeated, the display of the second line will not move to the first line. The same display will only shift within its own line for the number of times the shift is repeated.

Note: When using the internal reset, the electrical characteristics in the Power Supply Conditions Using Internal Reset Circuit table must be satisfied. If not, the HD44780U must be initialized by instructions. See the section, Initializing by Instruction.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step No.	Instruction										Display	Operation
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
1	Power supply on (the HD44780U is initialized by the internal reset circuit)										<input type="text"/>	Initialized. No display.
2	0	0	0	0	1	1	0	0	*	*	<input type="text"/>	Sets to 8-bit operation and selects 1-line display and 5 × 8 dot character font. (Number of display lines and character fonts cannot be changed after step #2.)
3	0	0	0	0	0	0	1	1	1	0	<input type="text" value="-"/>	Turns on display and cursor. Entire display is in space mode because of initialization.
4	0	0	0	0	0	0	0	1	1	0	<input type="text" value="-"/>	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	1	0	0	1	0	0	1	0	0	0	<input type="text" value="H_"/>	Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6	1	0	0	1	0	0	1	0	0	1	<input type="text" value="HI_"/>	Writes I.
7											<input type="text"/>	
8	1	0	0	1	0	0	1	0	0	1	<input type="text" value="HITACHI_"/>	Writes I.
9	0	0	0	0	0	0	0	1	1	1	<input type="text" value="HITACHI_"/>	Sets mode to shift display at the time of write.
10	1	0	0	0	1	0	0	0	0	0	<input type="text" value="ITACHI _"/>	Writes a space.

Table 11 8-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset (cont)

Step No.	Instruction										Display	Operation
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
11	1	0	0	1	0	0	1	1	0	1	TACHI M_	Writes M.
12											.	.
											.	.
											.	.
											.	.
13	1	0	0	1	0	0	1	1	1	1	MICROKO_	Writes O.
14	0	0	0	0	0	1	0	0	*	*	MICROKO	Shifts only the cursor position to the left.
15	0	0	0	0	0	1	0	0	*	*	MICROKO	Shifts only the cursor position to the left.
16	1	0	0	1	0	0	0	0	1	1	ICROCO	Writes C over K. The display moves to the left.
17	0	0	0	0	0	1	1	1	*	*	MICROCO	Shifts the display and cursor position to the right.
18	0	0	0	0	0	1	0	1	*	*	MICROCO_	Shifts the display and cursor position to the right.
19	1	0	0	1	0	0	1	1	0	1	ICROCOM_	Writes M.
20											.	.
											.	.
											.	.
											.	.
											.	.
21	0	0	0	0	0	0	0	0	1	0	HITACHI	Returns both display and cursor to the original position (address 0).

Table 12 4-Bit Operation, 8-Digit × 1-Line Display Example with Internal Reset

Step No.	Instruction						Display	Operation
	RS	R/W	DB7	DB6	DB5	DB4		
1	Power supply on (the HD44780U is initialized by the internal reset circuit)						<input type="text"/>	Initialized. No display.
2	Function set 0 0 0 0 1 0						<input type="text"/>	Sets to 4-bit operation. In this case, operation is handled as 8 bits by initialization, and only this instruction completes with one write.
3	Function set 0 0 0 0 1 0 0 0 0 0 * *						<input type="text"/>	Sets 4-bit operation and selects 1-line display and 5 × 8 dot character font. 4-bit operation starts from this step and resetting is necessary. (Number of display lines and character fonts cannot be changed after step #3.)
4	Display on/off control 0 0 0 0 0 0 0 0 1 1 1 0						<input type="text" value=" _"/>	Turns on display and cursor. Entire display is in space mode because of initialization.
5	Entry mode set 0 0 0 0 0 0 0 0 0 1 1 0						<input type="text" value=" _"/>	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
6	Write data to CGRAM/DDRAM 1 0 0 1 0 0 1 0 1 0 0 0						<input type="text" value=" H_"/>	Writes H. The cursor is incremented by one and shifts to the right.

Note: The control is the same as for 8-bit operation beyond step #6.

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset

Step No.	Instruction										Display	Operation
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
1	Power supply on (the HD44780U is initialized by the internal reset circuit)											Initialized. No display.
2	Function set 0 0 0 0 1 1 1 0 * *											Sets to 8-bit operation and selects 2-line display and 5 × 8 dot character font.
3	Display on/off control 0 0 0 0 0 0 1 1 1 0											Turns on display and cursor. All display is in space mode because of initialization.
4	Entry mode set 0 0 0 0 0 0 0 1 1 0											Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CGRAM. Display is not shifted.
5	Write data to CGRAM/DDRAM 1 0 0 1 0 0 1 0 0 0											Writes H. DDRAM has already been selected by initialization when the power was turned on. The cursor is incremented by one and shifted to the right.
6	.											.
7	Write data to CGRAM/DDRAM 1 0 0 1 0 0 1 0 0 1											Writes I.
8	Set DDRAM address 0 0 1 1 0 0 0 0 0 0											Sets DDRAM address so that the cursor is positioned at the head of the second line.

Table 13 8-Bit Operation, 8-Digit × 2-Line Display Example with Internal Reset (cont)

Step No.	Instruction										Display	Operation		
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0				
9	Write data to CGRAM/DDRAM										<table border="1"> <tr><td>HITACHI</td></tr> <tr><td>M_</td></tr> </table>	HITACHI	M_	Writes M.
HITACHI														
M_														
	1	0	0	1	0	0	1	1	0	1				
10														
11	Write data to CGRAM/DDRAM										<table border="1"> <tr><td>HITACHI</td></tr> <tr><td>MICROCO_</td></tr> </table>	HITACHI	MICROCO_	Writes O.
HITACHI														
MICROCO_														
	1	0	0	1	0	0	1	1	1	1				
12	Entry mode set										<table border="1"> <tr><td>HITACHI</td></tr> <tr><td>MICROCO_</td></tr> </table>	HITACHI	MICROCO_	Sets mode to shift display at the time of write.
HITACHI														
MICROCO_														
	0	0	0	0	0	0	0	1	1	1				
13	Write data to CGRAM/DDRAM										<table border="1"> <tr><td>ITACHI</td></tr> <tr><td>ICROCOM_</td></tr> </table>	ITACHI	ICROCOM_	Writes M. Display is shifted to the left. The first and second lines both shift at the same time.
ITACHI														
ICROCOM_														
	1	0	0	1	0	0	1	1	0	1				
14														
15	Return home										<table border="1"> <tr><td>HITACHI</td></tr> <tr><td>MICROCOM</td></tr> </table>	HITACHI	MICROCOM	Returns both display and cursor to the original position (address 0).
HITACHI														
MICROCOM														
	0	0	0	0	0	0	0	0	1	0				

Initializing by Instruction

If the power supply conditions for correctly operating the internal reset circuit are not met, initialization by instructions becomes necessary.

Refer to Figures 23 and 24 for the procedures on 8-bit and 4-bit initializations, respectively.

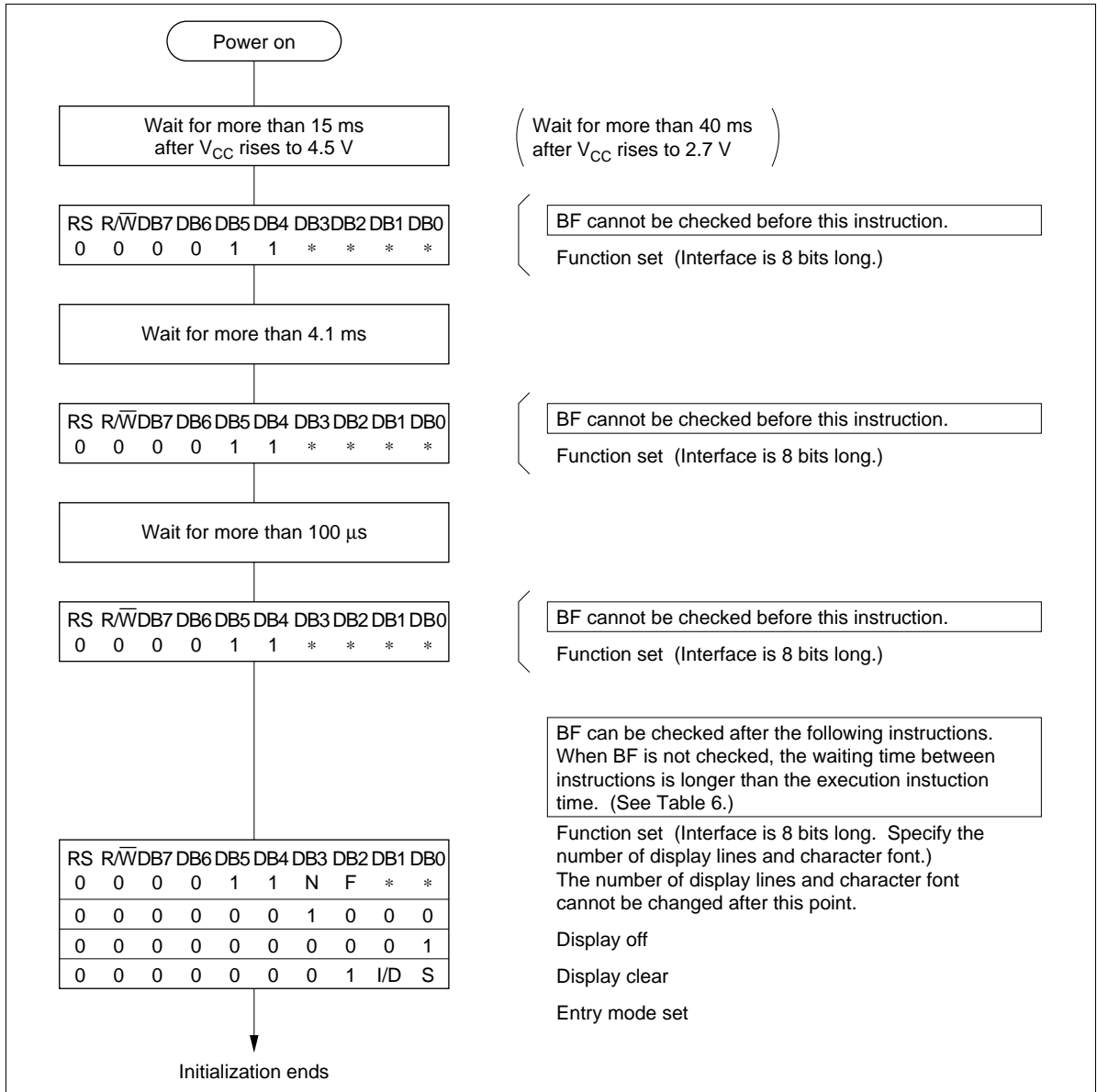


Figure 23 8-Bit Interface

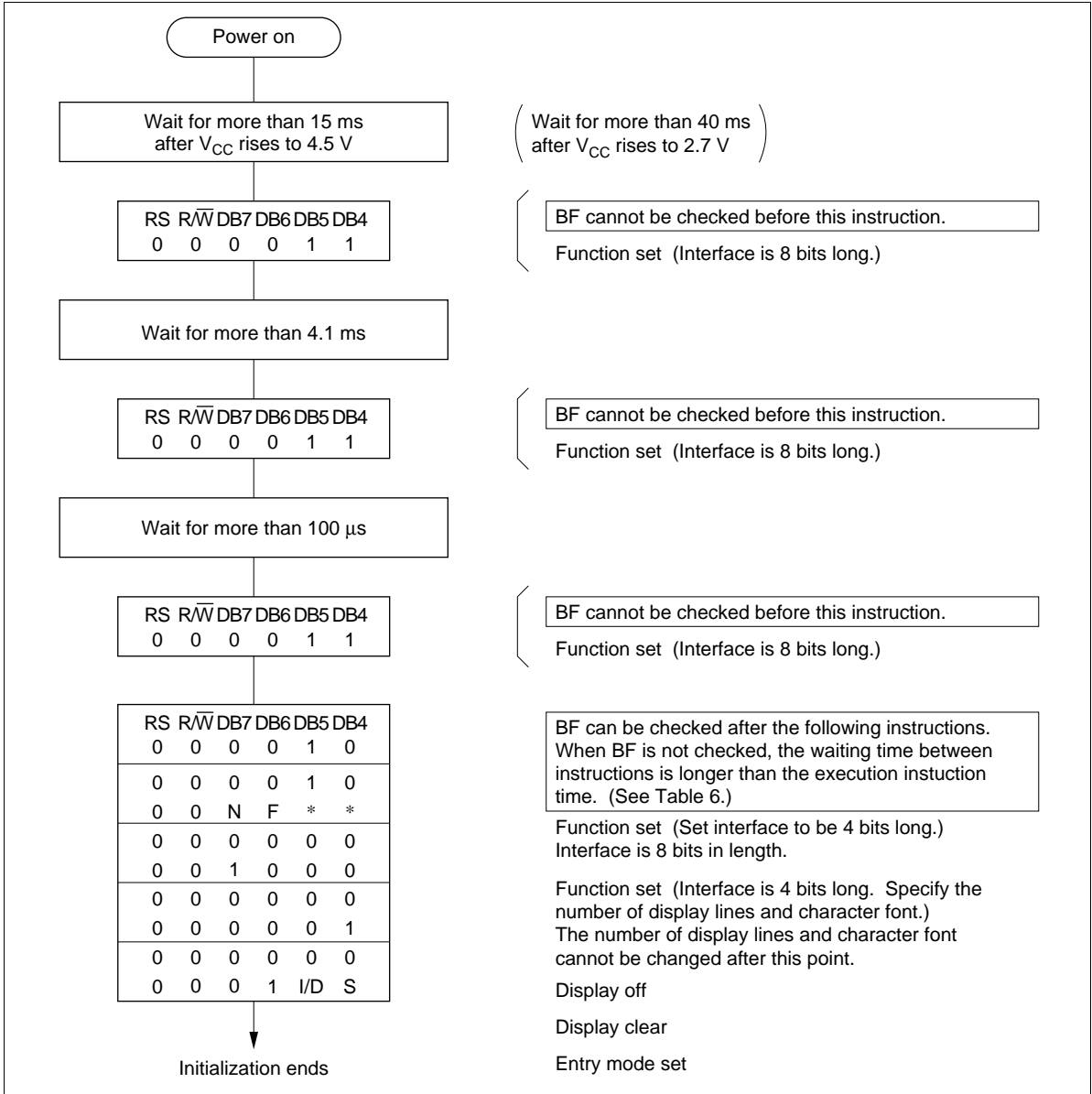


Figure 24 4-Bit Interface

Absolute Maximum Ratings*

Item	Symbol	Value	Unit	Notes
Power supply voltage (1)	V_{CC-GND}	-0.3 to +7.0	V	1
Power supply voltage (2)	V_{CC-V5}	-0.3 to +13.0	V	1, 2
Input voltage	V_t	-0.3 to $V_{CC} + 0.3$	V	1
Operating temperature	T_{opr}	-30 to +75	°C	
Storage temperature	T_{stg}	-55 to +125	°C	4

Note: * If the LSI is used above these absolute maximum ratings, it may become permanently damaged. Using the LSI within the following electrical characteristic limits is strongly recommended for normal operation. If these electrical characteristic conditions are also exceeded, the LSI will malfunction and cause poor reliability.

DC Characteristics ($V_{CC} = 2.7$ to 4.5 V, $T_a = -30$ to $+75^\circ\text{C}^{*3}$)

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	V _{IH1}	$0.7V_{CC}$	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	V _{IL1}	-0.3	—	0.55	V		6
Input high voltage (2) (OSC1)	V _{IH2}	$0.7V_{CC}$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	V _{IL2}	—	—	$0.2V_{CC}$	V		15
Output high voltage (1) (DB0–DB7)	V _{OH1}	$0.75V_{CC}$	—	—	V	$-I_{OH} = 0.1$ mA	7
Output low voltage (1) (DB0–DB7)	V _{OL1}	—	—	$0.2V_{CC}$	V	$I_{OL} = 0.1$ mA	7
Output high voltage (2) (except DB0–DB7)	V _{OH2}	$0.8V_{CC}$	—	—	V	$-I_{OH} = 0.04$ mA	8
Output low voltage (2) (except DB0–DB7)	V _{OL2}	—	—	$0.2V_{CC}$	V	$I_{OL} = 0.04$ mA	8
Driver on resistance (COM)	R _{COM}	—	2	20	k Ω	$\pm I_d = 0.05$ mA, V _{LCD} = 4 V	13
Driver on resistance (SEG)	R _{SEG}	—	2	30	k Ω	$\pm I_d = 0.05$ mA, V _{LCD} = 4 V	13
Input leakage current	I _{LI}	-1	—	1	μ A	V _{IN} = 0 to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/W)	-I _p	10	50	120	μ A	$V_{CC} = 3$ V	
Power supply current	I _{CC}	—	150	300	μ A	R _f oscillation, external clock $V_{CC} = 3$ V, $f_{osc} = 270$ kHz	10, 14
LCD voltage	V _{LCD1}	3.0	—	11.0	V	$V_{CC} - V_5$, 1/5 bias	16
	V _{LCD2}	3.0	—	11.0	V	$V_{CC} - V_5$, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics ($V_{CC} = 2.7$ to 4.5 V, $T_a = -30$ to $+75^\circ\text{C}^{*3}$)

Clock Characteristics

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Note*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		
	External clock rise time	t_{rcp}	—	—	0.2	μs		
	External clock fall time	t_{fcp}	—	—	0.2	μs		
R_f oscillation	Clock oscillation frequency	f_{OSC}	190	270	350	kHz	$R_f = 75\text{ k}\Omega$, $V_{CC} = 3\text{ V}$	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics

Write Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	Figure 25
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er} , t_{Ef}	—	—	25		
Address set-up time (RS, R/W to E)	t_{AS}	60	—	—		
Address hold time	t_{AH}	20	—	—		
Data set-up time	t_{DSW}	195	—	—		
Data hold time	t_H	10	—	—		

Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	1000	—	—	ns	Figure 26
Enable pulse width (high level)	PW_{EH}	450	—	—		
Enable rise/fall time	t_{Er} , t_{Ef}	—	—	25		
Address set-up time (RS, R/W to E)	t_{AS}	60	—	—		
Address hold time	t_{AH}	20	—	—		
Data delay time	t_{DDR}	—	—	360		
Data hold time	t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—		
Clock set-up time		t_{CSU}	500	—	—		
Data set-up time		t_{SU}	300	—	—		
Data hold time		t_{DH}	300	—	—		
M delay time		t_{DM}	-1000	—	1000		
Clock rise/fall time		t_{ct}	—	—	200		

Power Supply Conditions Using Internal Reset Circuit

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time		t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time		t_{OFF}	1	—	—		

DC Characteristics ($V_{CC} = 4.5$ to 5.5 V, $T_a = -30$ to $+75^\circ\text{C}^{*3}$)

Item	Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
Input high voltage (1) (except OSC1)	VIH1	2.2	—	V_{CC}	V		6
Input low voltage (1) (except OSC1)	VIL1	-0.3	—	0.6	V		6
Input high voltage (2) (OSC1)	VIH2	$V_{CC}-1.0$	—	V_{CC}	V		15
Input low voltage (2) (OSC1)	VIL2	—	—	1.0	V		15
Output high voltage (1) (DB0–DB7)	VOH1	2.4	—	—	V	$-I_{OH} = 0.205$ mA	7
Output low voltage (1) (DB0–DB7)	VOL1	—	—	0.4	V	$I_{OL} = 1.2$ mA	7
Output high voltage (2) (except DB0–DB7)	VOH2	$0.9 V_{CC}$	—	—	V	$-I_{OH} = 0.04$ mA	8
Output low voltage (2) (except DB0–DB7)	VOL2	—	—	$0.1 V_{CC}$	V	$I_{OL} = 0.04$ mA	8
Driver on resistance (COM)	RCOM	—	2	20	k Ω	$\pm I_d = 0.05$ mA, VLCD = 4 V	13
Driver on resistance (SEG)	RSEG	—	2	30	k Ω	$\pm I_d = 0.05$ mA, VLCD = 4 V	13
Input leakage current	I_{LI}	-1	—	1	μA	$V_{IN} = 0$ to V_{CC}	9
Pull-up MOS current (DB0–DB7, RS, R/W)	$-I_p$	50	125	250	μA	$V_{CC} = 5$ V	
Power supply current	I_{CC}	—	350	600	μA	R_f oscillation, external clock $V_{CC} = 5$ V, $f_{osc} = 270$ kHz	10, 14
LCD voltage	VLCD1	3.0	—	11.0	V	$V_{CC}-V_5$, 1/5 bias	16
	VLCD2	3.0	—	11.0	V	$V_{CC}-V_5$, 1/4 bias	16

Note: * Refer to the Electrical Characteristics Notes section following these tables.

AC Characteristics ($V_{CC} = 4.5$ to 5.5 V, $T_a = -30$ to $+75^\circ\text{C}^{*3}$)

Clock Characteristics

Item		Symbol	Min	Typ	Max	Unit	Test Condition	Notes*
External clock operation	External clock frequency	f_{cp}	125	250	350	kHz		11
	External clock duty	Duty	45	50	55	%		11
	External clock rise time	t_{rcp}	—	—	0.2	μs		11
	External clock fall time	t_{fcp}	—	—	0.2	μs		11
R_f oscillation	Clock oscillation frequency	f_{OSC}	190	270	350	kHz	$R_f = 91\text{ k}\Omega$ $V_{CC} = 5.0\text{ V}$	12

Note: * Refer to the Electrical Characteristics Notes section following these tables.

Bus Timing Characteristics

Write Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	500	—	—	ns	Figure 25
Enable pulse width (high level)	PW_{EH}	230	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/W to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	10	—	—		
Data set-up time	t_{DSW}	80	—	—		
Data hold time	t_H	10	—	—		

Read Operation

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Enable cycle time	t_{cycE}	500	—	—	ns	Figure 26
Enable pulse width (high level)	PW_{EH}	230	—	—		
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	20		
Address set-up time (RS, R/W to E)	t_{AS}	40	—	—		
Address hold time	t_{AH}	10	—	—		
Data delay time	t_{DDR}	—	—	160		
Data hold time	t_{DHR}	5	—	—		

Interface Timing Characteristics with External Driver

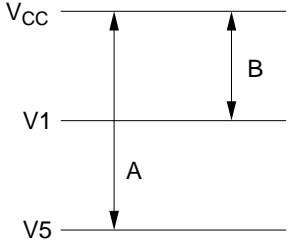
Item		Symbol	Min	Typ	Max	Unit	Test Condition
Clock pulse width	High level	t_{CWH}	800	—	—	ns	Figure 27
	Low level	t_{CWL}	800	—	—		
Clock set-up time		t_{CSU}	500	—	—		
Data set-up time		t_{SU}	300	—	—		
Data hold time		t_{DH}	300	—	—		
M delay time		t_{DM}	-1000	—	1000		
Clock rise/fall time		t_{ct}	—	—	100		

Power Supply Conditions Using Internal Reset Circuit

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Power supply rise time		t_{rCC}	0.1	—	10	ms	Figure 28
Power supply off time		t_{OFF}	1	—	—		

Electrical Characteristics Notes

1. All voltage values are referred to GND = 0 V.



$$A = V_{CC} - V5$$

$$B = V_{CC} - V1$$

$$A \geq 1.5 \text{ V}$$

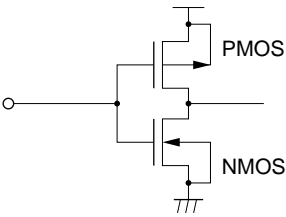
$$B \leq 0.25 \times A$$

The conditions of $V1$ and $V5$ voltages are for proper operation of the LSI and not for the LCD output level. The LCD drive voltage condition for the LCD output level is specified as LCD voltage VLCD.

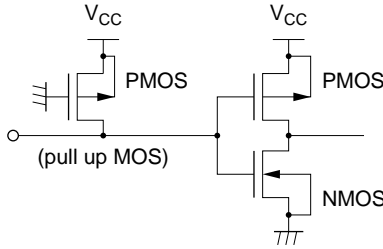
2. $V_{CC} \geq V1 \geq V2 \geq V3 \geq V4 \geq V5$ must be maintained.
3. For die products, specified at 75°C .
4. For die products, specified by the die shipment specification.
5. The following four circuits are I/O pin configurations except for liquid crystal display output.

Input pin

Pin: E (MOS without pull-up)

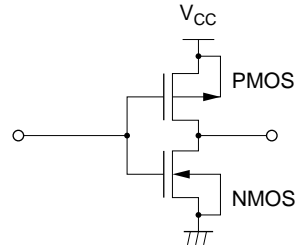


Pins: RS, $R\bar{W}$ (MOS with pull-up)



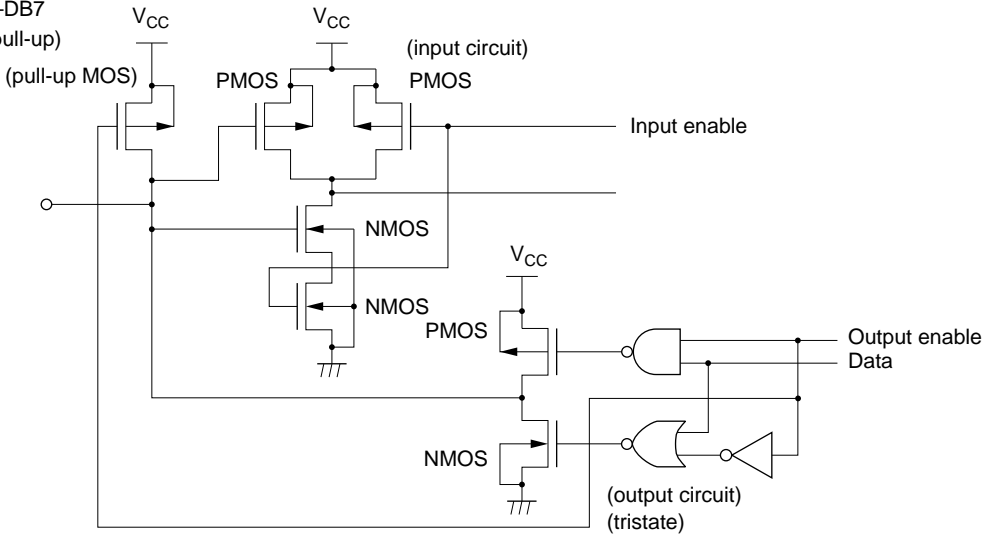
Output pin

Pins: CL1, CL2, M, D

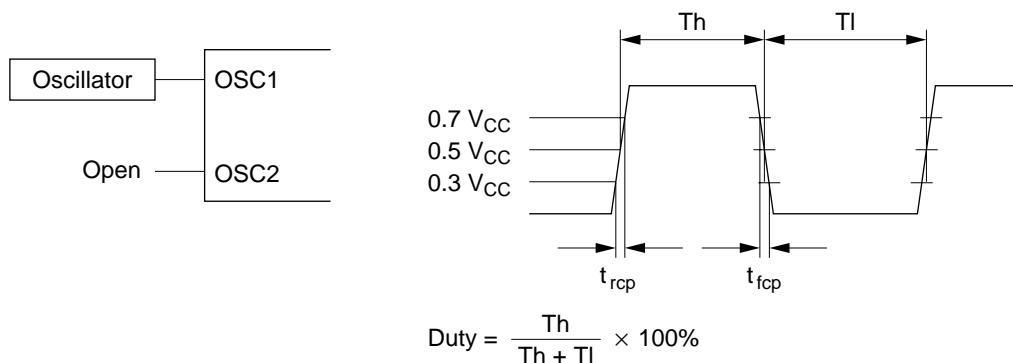


I/O Pin

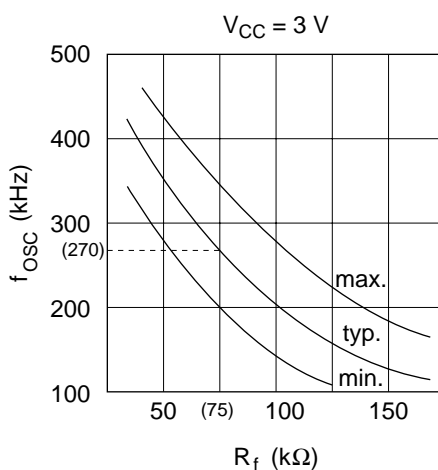
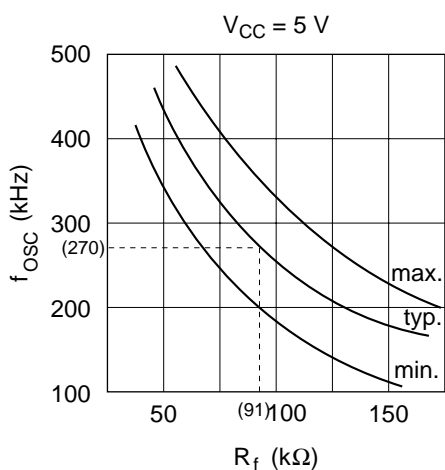
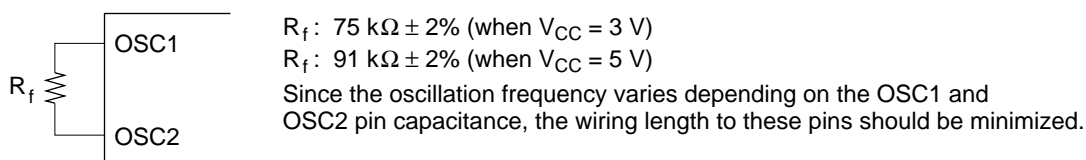
Pins: DB0 - DB7 (MOS with pull-up)



- 6. Applies to input pins and I/O pins, excluding the OSC1 pin.
- 7. Applies to I/O pins.
- 8. Applies to output pins.
- 9. Current flowing through pull-up MOSs, excluding output drive MOSs.
- 10. Input/output current is excluded. When input is at an intermediate level with CMOS, the excessive current flows through the input circuit to the power supply. To avoid this from happening, the input level must be fixed high or low.
- 11. Applies only to external clock operation.



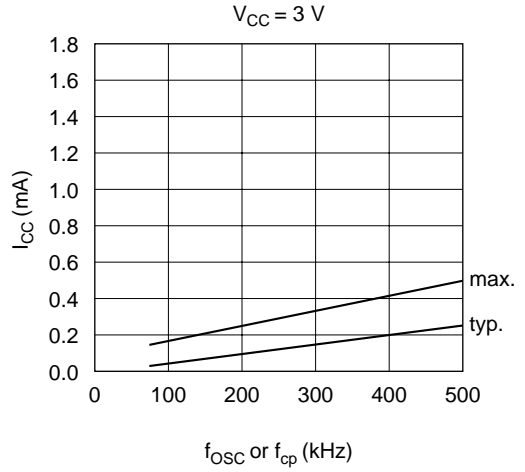
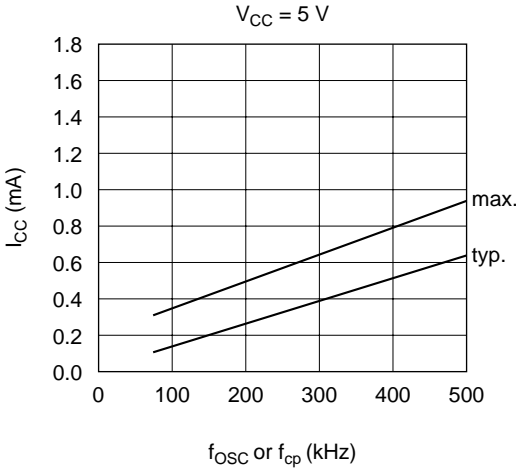
- 12. Applies only to the internal oscillator operation using oscillation resistor R_f .



13. RCOM is the resistance between the power supply pins (V_{CC} , V1, V4, V5) and each common signal pin (COM1 to COM16).

RSEG is the resistance between the power supply pins (V_{CC} , V2, V3, V5) and each segment signal pin (SEG1 to SEG40).

14. The following graphs show the relationship between operation frequency and current consumption.

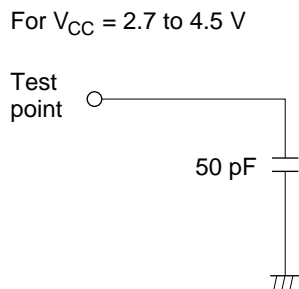
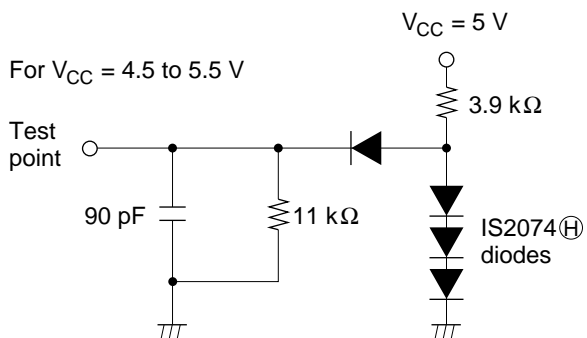


15. Applies to the OSC1 pin.

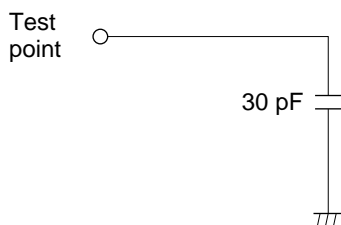
16. Each COM and SEG output voltage is within $\pm 0.15\text{ V}$ of the LCD voltage (V_{CC} , V1, V2, V3, V4, V5) when there is no load.

Load Circuits

Data Bus DB0 to DB7



External Driver Control Signals: CL1, CL2, D, M



Timing Characteristics

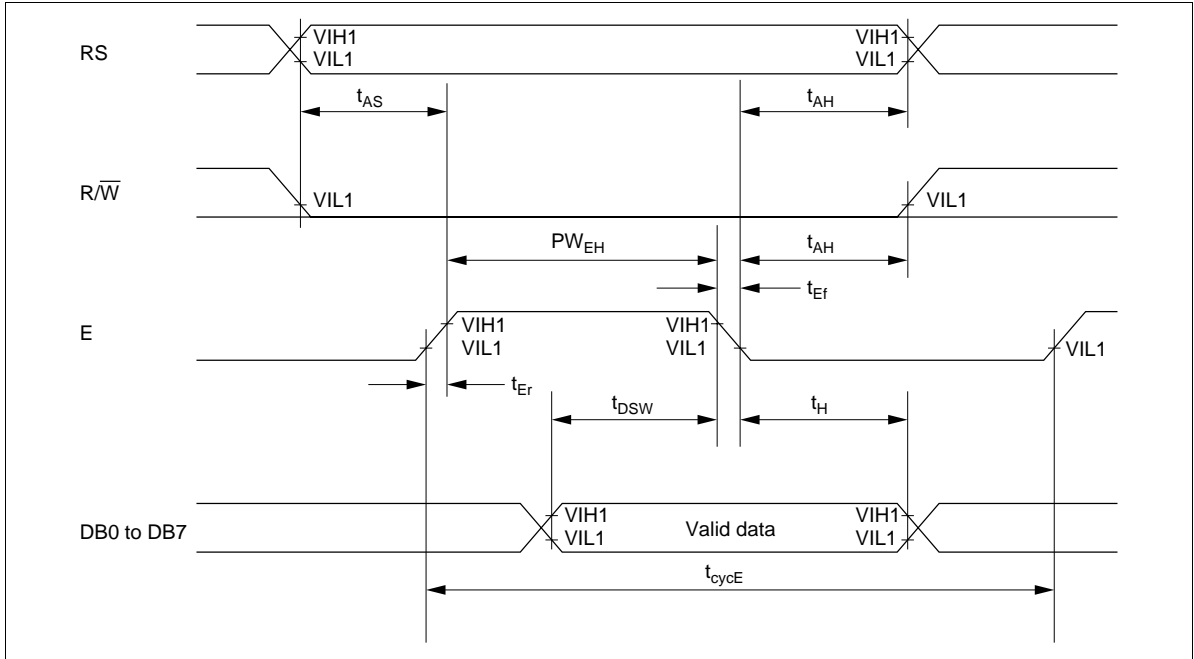
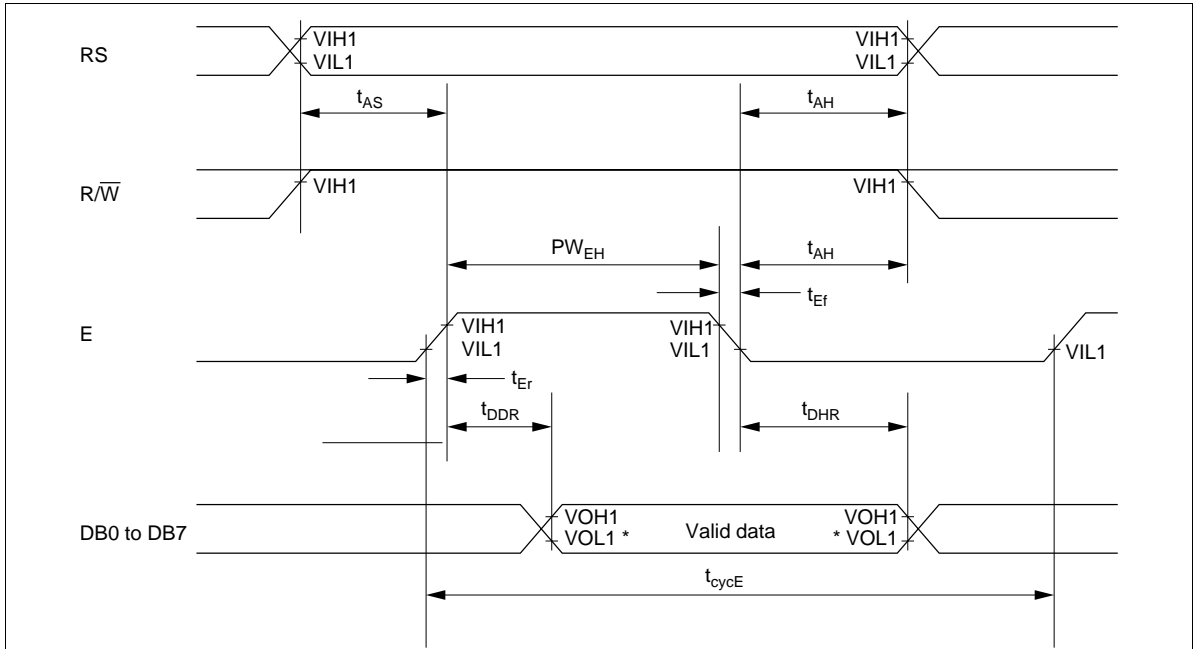


Figure 25 Write Operation



Note: * VOL1 is assumed to be 0.8 V at 2 MHz operation.

Figure 26 Read Operation

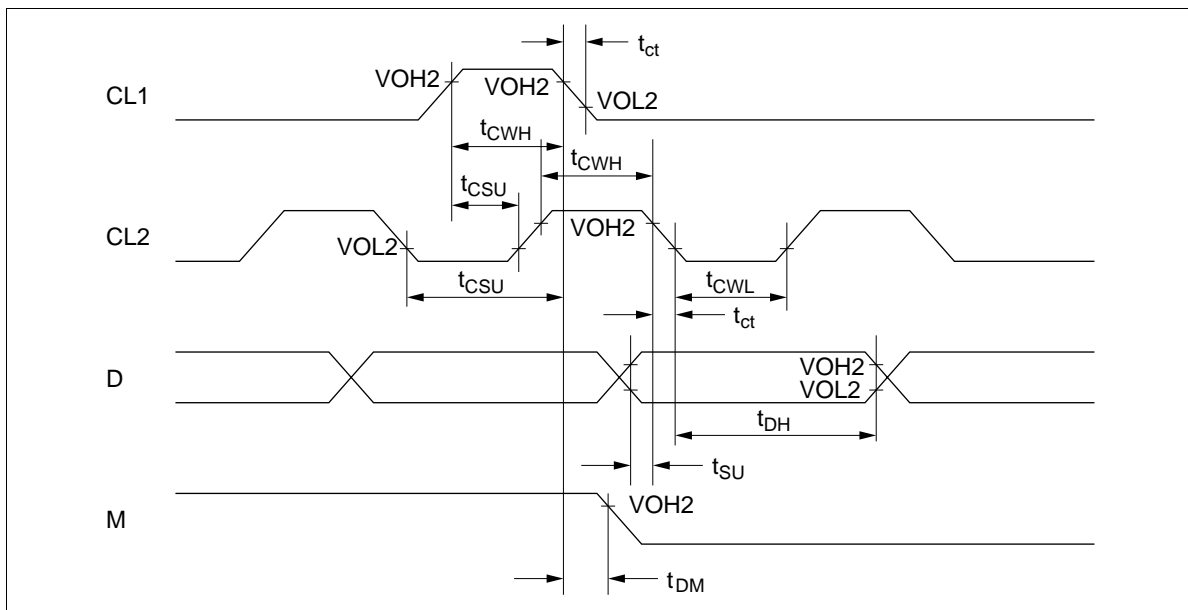
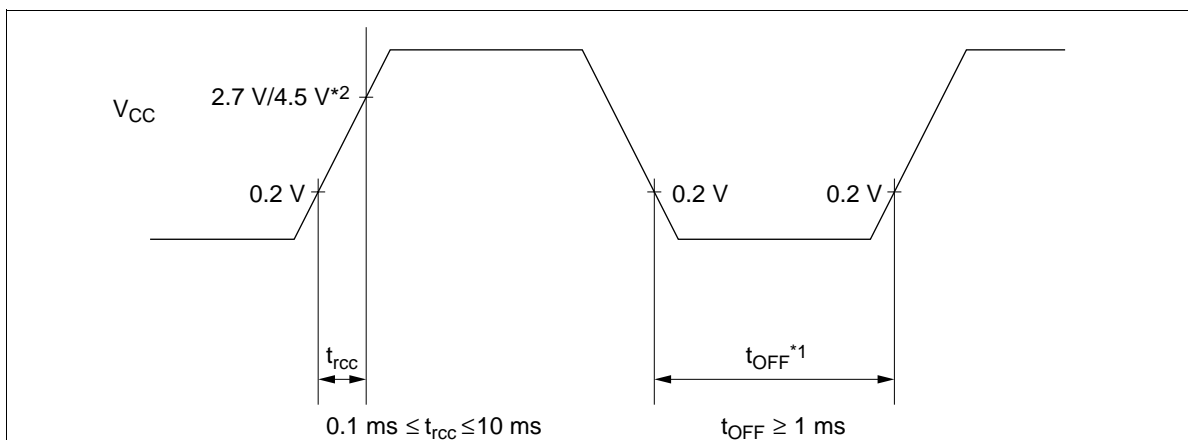


Figure 27 Interface Timing with External Driver



- Notes:
1. t_{OFF} compensates for the power oscillation period caused by momentary power supply oscillations.
 2. Specified at 4.5 V for 5-V operation, and at 2.7 V for 3-V operation.
 3. For if 4.5 V is not reached during 5-V operation, the internal reset circuit will not operate normally.
In this case, the LSI must be initialized by software. (Refer to the Initializing by Instruction section.)

Figure 28 Internal Power Supply Reset

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

HITACHI**Hitachi, Ltd.**

Semiconductor & Integrated Circuits.
 Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan
 Tel: Tokyo (03) 3270-2111 Fax: (03) 3270-5109

URL	NorthAmerica	: http://semiconductor.hitachi.com/
	Europe	: http://www.hitachi-eu.com/hel/ecg
	Asia (Singapore)	: http://www.has.hitachi.com.sg/grp3/sicd/index.htm
	Asia (Taiwan)	: http://www.hitachi.com.tw/E/Product/SICD_Frame.htm
	Asia (HongKong)	: http://www.hitachi.com.hk/eng/bo/grp3/index.htm
	Japan	: http://www.hitachi.co.jp/Sicd/indx.htm

For further information write to:

Hitachi Semiconductor
 (America) Inc.
 179 East Tasman Drive,
 San Jose, CA 95134
 Tel: <1> (408) 433-1990
 Fax: <1> (408) 433-0223

Hitachi Europe GmbH
 Electronic components Group
 Dornacher Straße 3
 D-85622 Feldkirchen, Munich
 Germany
 Tel: <49> (89) 9 9180-0
 Fax: <49> (89) 9 29 30 00

Hitachi Europe Ltd.
 Electronic Components Group.
 Whitebrook Park
 Lower Cookham Road
 Maidenhead
 Berkshire SL6 8YA, United Kingdom
 Tel: <44> (1628) 585000
 Fax: <44> (1628) 778322

Hitachi Asia Pte. Ltd.
 16 Collyer Quay #20-00
 Hitachi Tower
 Singapore 049318
 Tel: 535-2100
 Fax: 535-1533

Hitachi Asia Ltd.
 Taipei Branch Office
 3F, Hung Kuo Building, No.167,
 Tun-Hwa North Road, Taipei (105)
 Tel: <886> (2) 2718-3666
 Fax: <886> (2) 2718-8180

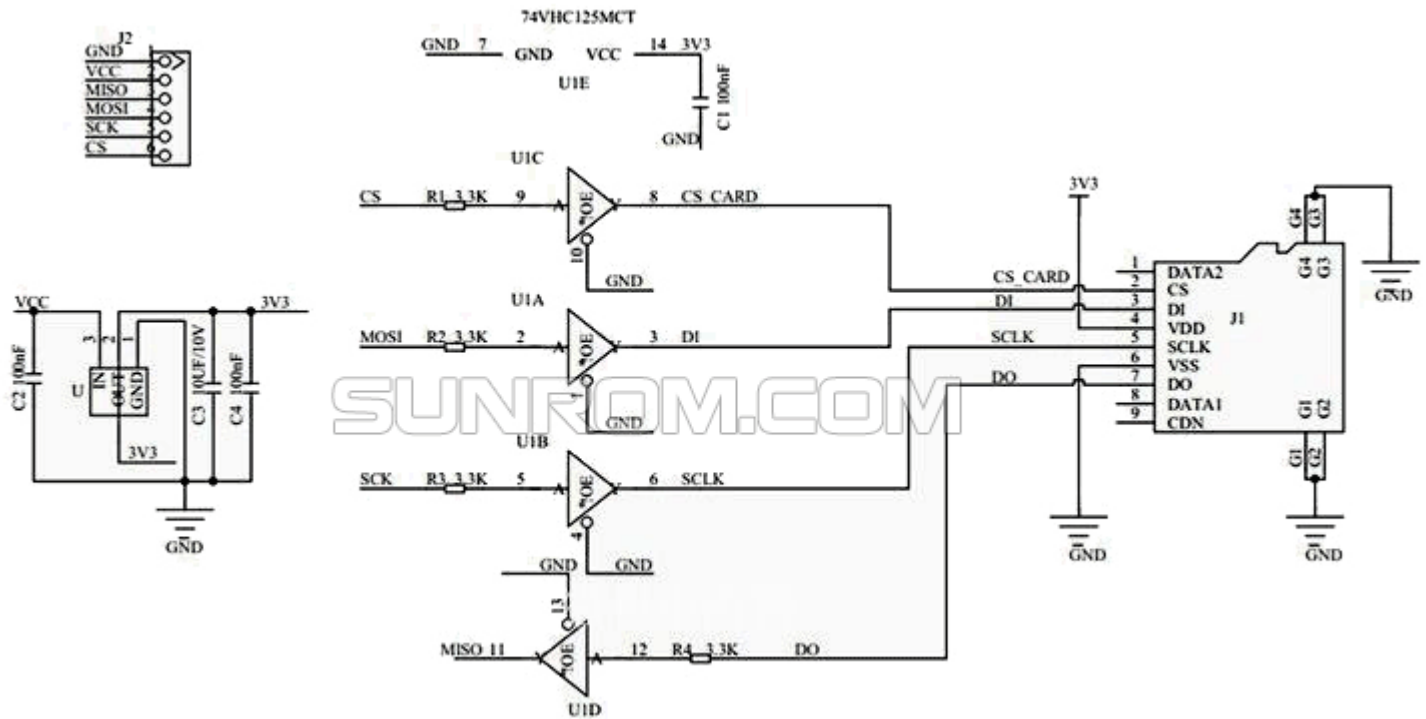
Hitachi Asia (Hong Kong) Ltd.
 Group III (Electronic Components)
 7/F., North Tower, World Finance Centre,
 Harbour City, Canton Road, Tsim Sha Tsui,
 Kowloon, Hong Kong
 Tel: <852> (2) 735 9218
 Fax: <852> (2) 730 0281
 Telex: 40815 HITEC HX

Copyright © Hitachi, Ltd., 1998. All rights reserved. Printed in Japan.

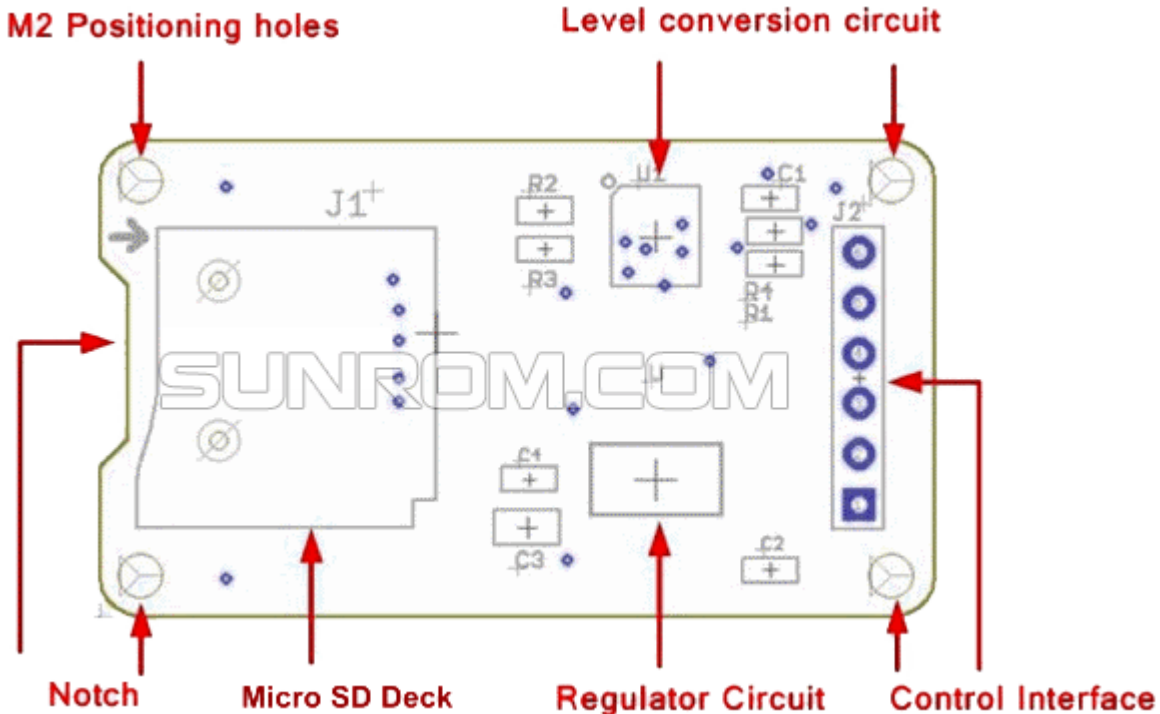
MICRO SD CARD INTERFACING MODULE

On Board Level shifter and 3.3V regulator allows you to easily access micro SD card from 5V microcontrollers through SPI

BOARD SCHEMATIC



DIMENSIONS



SPECIFICATIONS

Product Name	SD Card Interfacing Board
VCC Operating Voltage	5V DC
I/O Interfacing Voltage	3.3V to 5V DC
Communication	SPI
Regulator Chip	AMS1117-3.3
Level Conversion Chip	74VHC125
Mounting Screws	2.2mm Dia for 4x M2

PIN OUT

Pin	Type	Details
CS	Input	SPI Chip Select
SCK	Input	SPI Serial Clock
MOSI	Input	SPI Master Out Slave In
MISO	Output	SPI Master In Slave Out
VCC	Power	5V DC regulated
GND	Ground	Common Ground

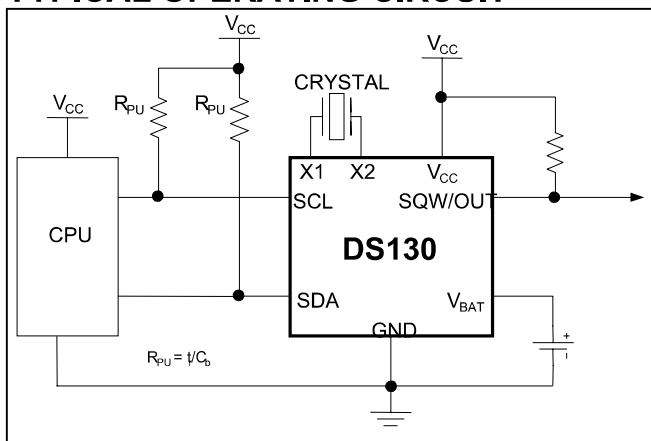
ORDERING DETAILS

Sunrom Part#	Ordering Page
5066	http://www.sunrom.com/m/5066

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

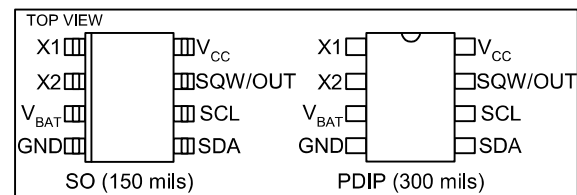
TYPICAL OPERATING CIRCUIT



FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I²C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+Denotes a lead-free/RoHS-compliant package.

*A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial.....	0°C to +70°C
Industrial	-40°C to +85°C
Storage Temperature Range	-55°C to +125°C
Soldering Temperature (DIP, leads).....	+260°C for 10 seconds
Soldering Temperature (surface mount).....	Refer to the JPC/JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS

(T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V _{CC}		4.5	5.0	5.5	V
Logic 1 Input	V _{IH}		2.2		V _{CC} + 0.3	V
Logic 0 Input	V _{IL}		-0.3		+0.8	V
V _{BAT} Battery Voltage	V _{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I _{LI}		-1		1	μA
I/O Leakage (SDA, SQW/OUT)	I _{LO}		-1		1	μA
Logic 0 Output (I _{OL} = 5mA)	V _{OL}				0.4	V
Active Supply Current (f _{SCL} = 100kHz)	I _{CCA}				1.5	mA
Standby Current	I _{CCS}	(Note 3)			200	μA
V _{BAT} Leakage Current	I _{BATLKG}			5	50	nA
Power-Fail Voltage (V _{BAT} = 3.0V)	V _{PF}		1.216 x V _{BAT}	1.25 x V _{BAT}	1.284 x V _{BAT}	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V _{BAT} Current (OSC ON); SQW/OUT OFF	I _{BAT1}			300	500	nA
V _{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I _{BAT2}			480	800	nA
V _{BAT} Data-Retention Current (Oscillator Off)	I _{BATDR}			10	100	nA

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f _{SCL}		0		100	kHz
Bus Free Time Between a STOP and START Condition	t _{BUF}		4.7			μs
Hold Time (Repeated) START Condition	t _{HD:STA}	(Note 4)	4.0			μs
LOW Period of SCL Clock	t _{LOW}		4.7			μs
HIGH Period of SCL Clock	t _{HIGH}		4.0			μs
Setup Time for a Repeated START Condition	t _{SU:STA}		4.7			μs
Data Hold Time	t _{HD:DAT}		0			μs
Data Setup Time	t _{SU:DAT}	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t _R				1000	ns
Fall Time of Both SDA and SCL Signals	t _F				300	ns
Setup Time for STOP Condition	t _{SU:STO}		4.7			μs

CAPACITANCE(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C _{I/O}				10	pF
Capacitance Load for Each Bus Line	C _B	(Note 7)			400	pF

Note 1: All voltages are referenced to ground.**Note 2:** Limits at -40°C are guaranteed by design and are not production tested.**Note 3:** I_{CCS} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 5:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V_{IH(MIN)} of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 6:** The maximum t_{HD:DAT} only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.**Note 7:** C_B—total capacitance of one bus line in pF.

TIMING DIAGRAM

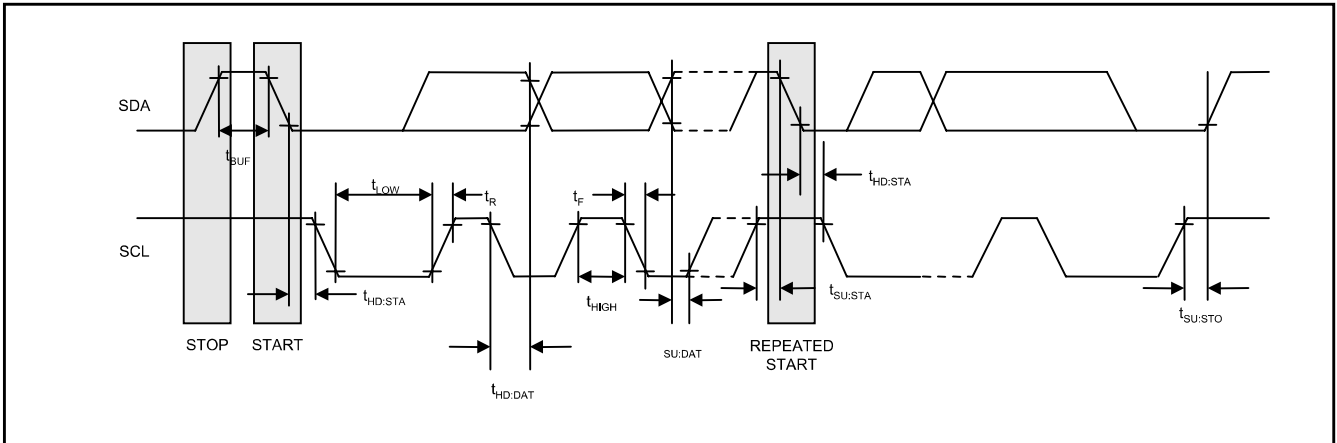
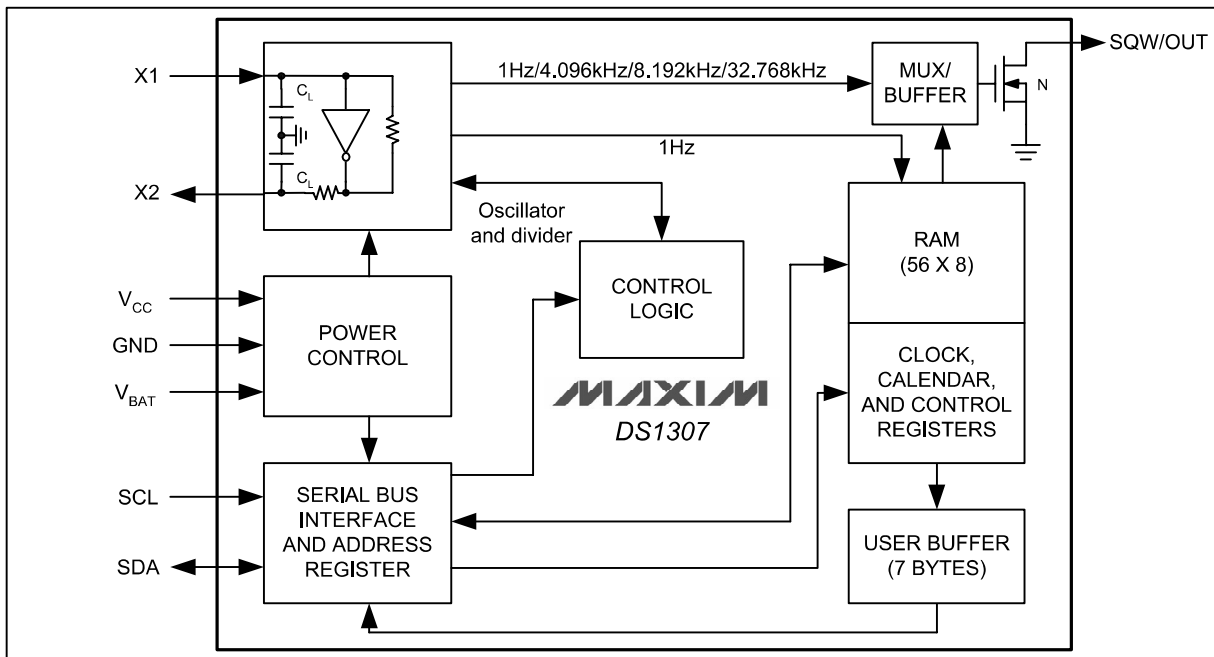
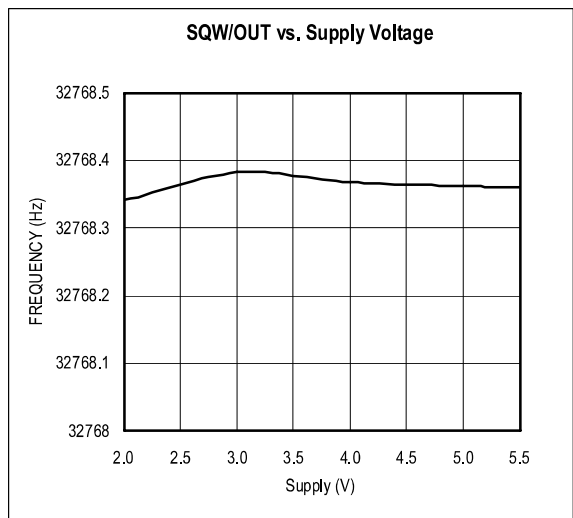
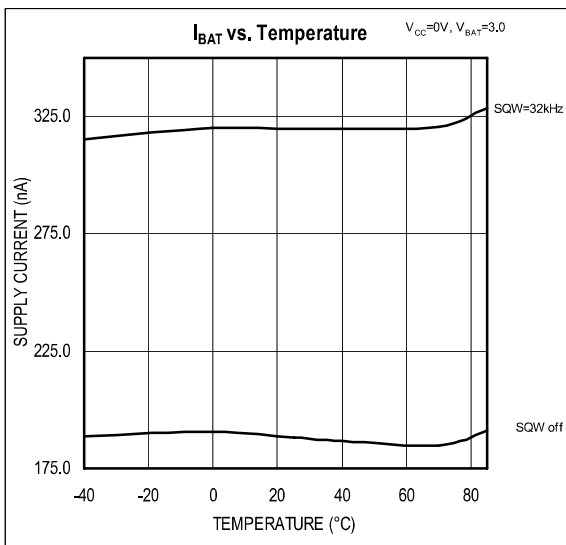
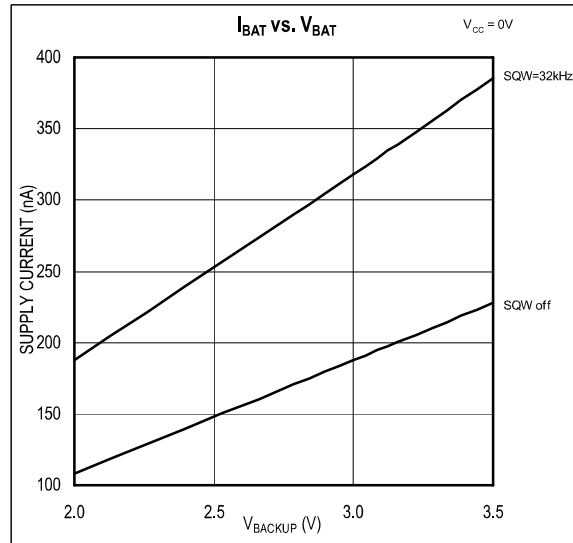
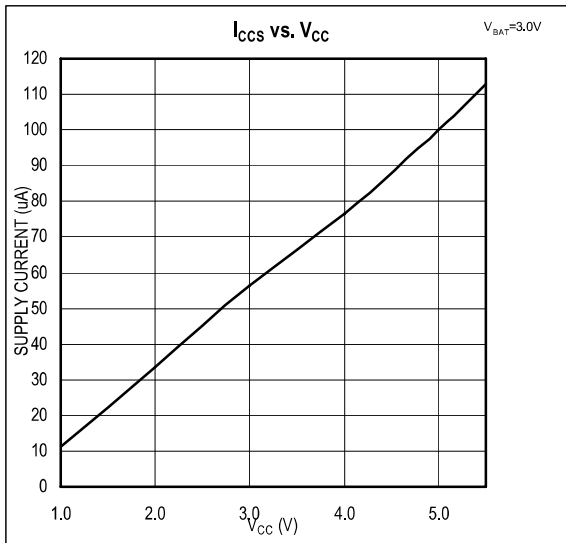


Figure 1. Block Diagram



TYPICAL OPERATING CHARACTERISTICS

(V_{CC} = 5.0V, T_A = +25°C, unless otherwise noted.)



PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1. Note: For more information on crystal selection and crystal layout considerations, refer to <i>Application Note 58: Crystal Considerations with Dallas Real-Time Clocks</i> .
2	X2	
3	V _{BAT}	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V _{BAT} pin may prevent proper operation. If a backup supply is not required, V _{BAT} must be grounded. The nominal power-fail trip point (V _{PF}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V _{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/qa/info/ul/ .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I ² C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} .
6	SCL	Serial Clock Input. SCL is the clock input for the I ² C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} .
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V _{CC} or V _{BAT} applied. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} . If not used, this pin can be left floating.
8	V _{CC}	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V _{CC} is below V _{TP} , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I²C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} +0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the serial RTC.

当VCC低于1.25xVbat时候，DS1307地址复位，拒绝主机的访问

OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1 shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

CLOCK ACCURACY

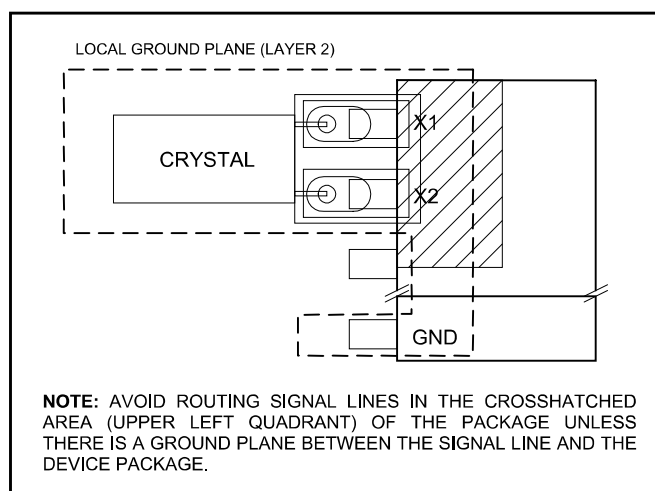
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to Application Note 58: *Crystal Considerations with Dallas Real-Time Clocks* for detailed information.

Table 1. Crystal Specifications*

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	f_0		32.768		kHz
Series Resistance	ESR			45	k Ω
Load Capacitance	C_L		12.5		pF

*The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to Application Note 58: *Crystal Considerations for Dallas Real-Time Clocks* for additional specifications.

Figure 2. Recommended Layout for Crystal



RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on.) Illogical time and date entries result in undefined operation. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled. On first application of power to the device the time and date registers are typically reset to 01/01/00 01 00:00:00 (MM/DD/YY DOW HH:MM:SS). The CH bit in the seconds register will be set to a 1. The clock can be halted whenever the timekeeping functions are not required, which minimizes current (I_{BATDR}).

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12-hour or 24-hour mode-select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20 to 23 hours). The hours value must be re-entered whenever the 12/24-hour mode bit is changed.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any I²C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I²C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

Table 2. Timekeeper Registers

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12 1	10 Hour	10 Hour	Hours			Hours	1–12 +AM/PM 00–23	
		24 0	PM/ AM							
03h	0	0	0	0	0	DAY		Day	01–07	
04h	0	0	10 Date		Date			Date	01–31	
05h	0	0	0	10 Month	Month			Month	01–12	
06h	10 Year			Year			Year	00–99		
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh								RAM 56 x 8	00h–FFh	

0 = Always reads back as 0.

CONTROL REGISTER

The DS1307 control register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Bit 7: Output Control (OUT). This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0. On initial application of power to the device, this bit is typically set to a 0.

Bit 4: Square-Wave Enable (SQWE). This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave. On initial application of power to the device, this bit is typically set to a 0.

Bits 1 and 0: Rate Select (RS[1:0]). These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits. On initial application of power to the device, these bits are typically set to a 1.

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

I²C DATA BUS

The DS1307 supports the I²C protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the I²C bus.

Figures 3, 4, and 5 detail how data is transferred on the I²C bus.

- Data transfer can be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

START data transfer: A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

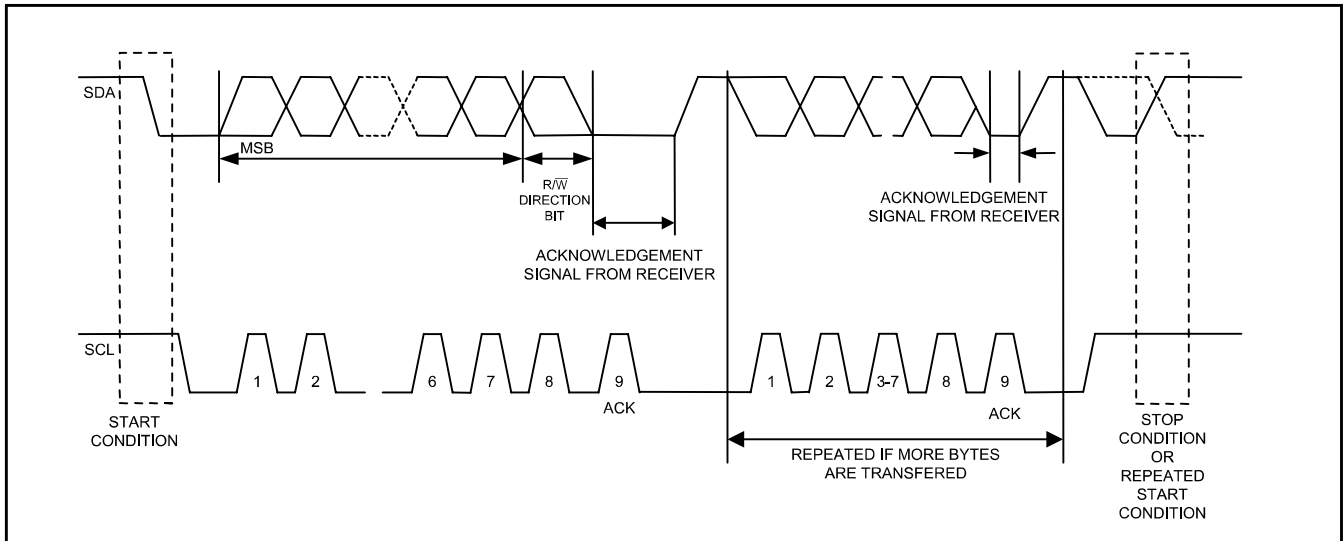
STOP data transfer: A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the I²C bus specifications a standard mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the standard mode (100kHz) only.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Figure 3. Data Transfer on I²C Serial Bus

Depending upon the state of the $\overline{R/W}$ bit, two types of data transfer are possible:

1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a "not acknowledge" is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307 can operate in the following two modes:

1. **Slave Receiver Mode (Write Mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Hardware performs address recognition after reception of the slave address and direction bit (see Figure 4). The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit (R/W), which for a write is 0. After receiving and decoding the slave address byte, the DS1307 outputs an acknowledge on SDA. After the DS1307 acknowledges the slave address + write bit, the master transmits a word address to the DS1307. This sets the register pointer on the DS1307, with the DS1307 acknowledging the transfer. The master can then transmit zero or more bytes of data with the DS1307 acknowledging each byte received. The register pointer automatically increments after each data byte are written. The master will generate a STOP condition to terminate the data write.
2. **Slave Transmitter Mode (Read Mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. The DS1307 transmits serial data on SDA while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (see Figure 5). The slave address byte is the first byte received after the START condition is generated by the master. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit (R/W), which is 1 for a read. After receiving and decoding the slave address the DS1307 outputs an acknowledge on SDA. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The register pointer automatically increments after each byte are read. The DS1307 must receive a Not Acknowledge to end a read.

Figure 4. Data Write—Slave Receiver Mode

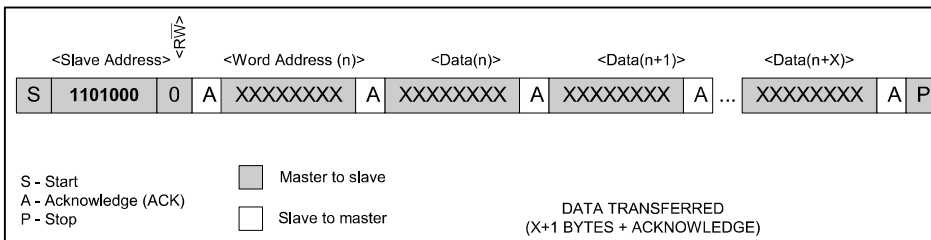


Figure 5. Data Read—Slave Transmitter Mode

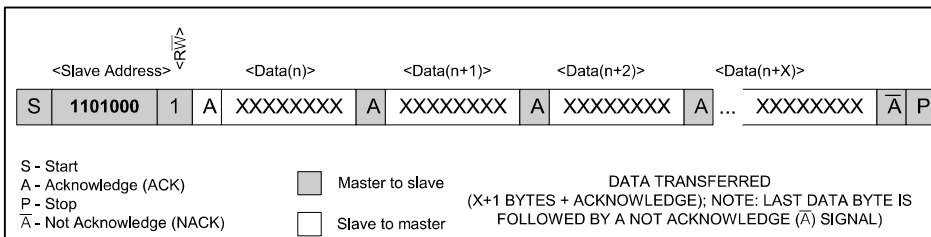
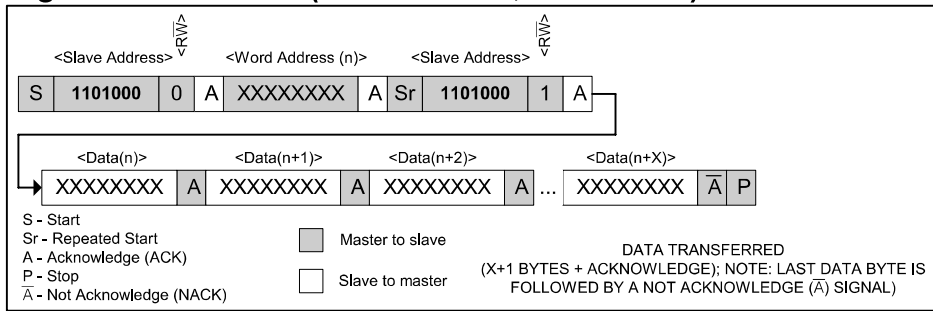


Figure 6. Data Read (Write Pointer, Then Read)—Slave Receive and Transmit



PACKAGE INFORMATION

For the latest package outline information and land patterns, go to www.maxim-ic.com/packages.

PACKAGE TYPE	PACKAGE CODE	DOCUMENT NO.
8 PDIP	—	21-0043
8 SO	—	21-0041

REVISION HISTORY

REVISION DATE	DESCRIPTION	PAGES CHANGED
100208	Moved the <i>Typical Operating Circuit</i> and <i>Pin Configurations</i> to first page.	1
	Removed the leaded part numbers from the <i>Ordering Information</i> table.	1
	Added an open-drain transistor to SQW/OUT in the block diagram (Figure 1).	4
	Added the pullup voltage range for SDA, SCL, and SQW/OUT to the <i>Pin Description</i> table and noted that SQW/OUT can be left open if not used.	6
	Added default time and date values on first application of power to the <i>Clock and Calendar</i> section and deleted the note that initial power-on state is not defined.	8
	Added default on initial application of power to bit info in the <i>Control Register</i> section.	9
	Updated the <i>Package Information</i> section to reflect new package outline drawing numbers.	13