



ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

**Trabajo de Fin de Grado**

**TRAZADOR DE CURVAS I-V DE BAJO COSTE PARA  
CÉLULAS FOTOVOLTAICAS**

Titulación: Grado en Ingeniería Electrónica Industrial y Automática

Alumno: Enrique Pérez Hernández

Tutores: Beatriz Rodríguez Mendoza

Alejandro José Ayala Alfonso

Junio, 2020

## Abstract

The aim of this project is to design and implement a portable device to obtain C-V and P-V characteristics curves from photovoltaic panels, with a following visualization of these. Through a user interface designed for installation on a computer, it is possible to order the measurement device to collect data and transmit it to the computer through USB or Bluetooth connection, obtain the curve in a graph or save it in a file.

The data of these curves will be obtained by connecting the photovoltaic panel to a capacitor. The charge of this element increases its resistance values. When starting this process with the capacitor discharged, its resistance value is null and it short circuits the panel (maximum current). As the charging process continues, the resistance increases and the current decreases, ending with current null at the end of the charging process (equivalent to an open circuit with infinite resistance).

From this process the values of voltage and current will be measured, analogue signals that will be converted to digital by ADC converters, each connected to a microcontroller. One of them will be the master of the measure, order of managing and controlling the process and communication with the computer. The other, the slave, will only be at the master's order and will send the collected data once the reading is finished.

# Índice

<b>CAPÍTULO I: INTRODUCCIÓN GENERAL</b> .....	5
I.1 - Introducción. ....	6
I.2 - Objetivos del proyecto. Descripción general del sistema.....	8
I.3 - Estructura de la memoria.....	10
<b>CAPÍTULO II: PANELES FOTOVOLTAICOS</b> .....	11
II.1 - Introducción .....	12
II.2 - Teoría de la célula fotovoltaica .....	12
II.3 - Tipos de módulos fotovoltaicos .....	14
II.4 - Curva característica.....	14
<b>CAPÍTULO III: CARACTERÍSTICAS DEL MICROCONTROLADOR</b> .....	18
III.1 - Introducción .....	19
III.2 - Características generales de Arduino.....	19
III.3 - Arduino Uno. ....	20
III.3.1 - Tarjeta Arduino Uno .....	20
III.3.2 - Microcontrolador ATmega328P. Introducción .....	21
III.3.3 - Microcontrolador ATmega328P. Patillaje.....	21
III.3.4 - Microcontrolador ATmega328P. Memoria .....	21
III.3.5 - Microcontrolador ATmega328P. Registros.....	22
III.3.6 - Microcontrolador ATmega328P. Reloj.....	22
III.3.7 - Microcontrolador ATmega328P. Interrupciones externas .....	23
III.3.8 - Microcontrolador ATmega28P. Protocolos de comunicación.....	23
<b>CAPÍTULO IV: HERRAMIENTAS DE DISEÑO</b> .....	26
IV.1 - Introducción.....	27
IV.2 - IDE Arduino .....	27
IV.3 - Anaconda .....	27
IV.3.1 - Spyder.....	28
IV.3.2 - PyQt Designer.....	28
IV.4 - KiCad.....	29
<b>CAPÍTULO V: DESCRIPCIÓN DEL SISTEMA</b> .....	30
V.1 - Introducción.....	31
V.2 - Bloques del sistema.....	31
V.2.1 - Entrada del panel FV y sistema de medida.....	31
V.2.2 - Acondicionamiento y adquisición de señal .....	33
V.2.3 - Gestión y control de la medida .....	34
V.2.4 - Alimentación.....	37
V.3 - Elementos Electrónicos .....	38
V.3.1 - Conversor ADC.....	38
V.3.2 - Medida de corriente. Resistencia Shunt .....	39
V.3.3 - Amplificador operacional .....	40
V.3.4 - Módulo de conmutación .....	40
V.3.5 - Buck-Boost.....	41

V.3.6 - Regulador de tensión.....	42
V.3.7 - Módulo Bluetooth.....	42
V.3.8 - Módulo UART-USB .....	43
V.4 - Adquisición de datos .....	43
<b>CAPÍTULO VI: DESCRIPCIÓN DEL SOFTWARE.....</b>	<b>45</b>
VI.1 - Introducción.....	46
VI.2 - Funcionamiento general .....	46
VI.2.1 - Ciclo de trabajo de los microcontroladores .....	46
VI.2.2 - Software de acondicionamiento de datos .....	50
<b>CAPÍTULO VII: INTERFAZ GRÁFICA.....</b>	<b>52</b>
VII.1 - Introducción.....	53
VII.2 - Funcionamiento de la aplicación .....	53
VII.3 - Generación de archivos de datos .....	55
<b>CAPÍTULO VIII: RESULTADOS EXPERIMENTALES.....</b>	<b>58</b>
VIII.1 - Introducción .....	59
VIII.2 - Paneles fotovoltaicos empleados.....	59
VIII.3 - Resultados de la Prueba 1.....	60
VIII.3.1 - Panel 1 .....	60
VIII.3.2 - Panel 2.....	61
VIII.3.3 - Serie panel 1 y 2.....	62
VIII.4 - Resultados de la Prueba 2.....	62
VIII.4.1 - Panel 1 .....	63
VIII.4.2 - Panel 2.....	63
VIII.4.3 - Serie panel 1 y 2.....	64
VIII.5 - Resultados de la Prueba 3.....	64
VIII.5.1 - Panel 1 .....	65
VIII.5.2 - Panel 2.....	65
VIII.6 - Análisis de datos .....	66
<b>CAPÍTULO IX: PRESUPUESTO .....</b>	<b>68</b>
IX.1 - Costes materiales .....	69
IX.2 - Costes mano de obra .....	69
IX.3 - Costes totales .....	70
<b>CAPÍTULO X: APORTACIONES Y CONCLUSIONES.....</b>	<b>71</b>
<b>BIBLIOGRAFÍA .....</b>	<b>73</b>
<b>GLOSARIO.....</b>	<b>75</b>
<b>ANEXOS.....</b>	<b>76</b>

## **CAPÍTULO I: INTRODUCCIÓN GENERAL**

## I.1 - Introducción.

Es indudable que el actual modelo energético a base de tecnologías contaminantes de generación eléctrica es insostenible para el planeta. El cambio climático, el aumento de la temperatura media terrestre, el deshielo de los polos y el aumento de muertes y enfermedades en los seres humanos, son una prueba de ello.

En el año 2018 se alcanzó un nuevo máximo de niveles de concentración de gases de efecto invernadero. La Organización Meteorológica Mundial “OMM” [1] indica que en el caso del CO<sub>2</sub>, uno de los principales gases responsables del calentamiento global, habría que retroceder al menos tres millones de años para encontrar una concentración tan grande en la atmósfera. Estas altas concentraciones son producidas en mayor nivel por la industria, la generación de energía y el transporte, tanto de mercancías como de personas.

La generación de energía eléctrica se caracteriza por transformar cualquier otro tipo de energía (cinética, química, nuclear, solar, térmica, etc.) en energía eléctrica. Encontramos diversos tipos de centrales generadoras de electricidad, tales como nucleares, hidroeléctricas, eólicas, fotovoltaicas, geotérmicas, o las que queman combustibles fósiles, como el petróleo y sus derivados, gas natural o carbón. Estas últimas son las que acarrearán una mayor liberación de gases de efecto invernadero.

Según la OMM, el 85% de la energía global que se produce proviene de minerales fósiles y solo el 15% proviene de energías más limpias como la hidroeléctrica, eólica y solar. Esto demuestra que, si se quieren alcanzar los límites de emisiones impuestos por la unión europea para los años venideros, se debe migrar hacia una producción de energía que se familiarice mejor con el medio ambiente. En la Figura I.1 se puede observar una central eléctrica de carbón, uno de los tipos más contaminantes.



*Figura I.1.* - Central eléctrica de carbón

Con el objetivo de generar energía a base de tecnologías más amigables con el medio ambiente, se encuentran las energías renovables, entre las que destacan las tecnologías eólica, solar fotovoltaica, hidroeléctrica, solar térmica y geotérmica, entre otras.

La energía fotovoltaica proviene de la irradiación solar. A partir de una célula fotovoltaica (Figura I.2), la irradiancia solar incide en ella y genera una corriente eléctrica gracias al efecto fotoeléctrico. Este efecto se lleva a cabo cuando la radiación solar incide sobre el material semiconductor del que está compuesta la célula, generalmente de silicio, que absorbe los fotones de luz con suficiente energía como para originar un “salto de electrones”. Cuando estos electrones

son capturados, el resultado es una corriente eléctrica. Se utiliza este semiconductor por sus propiedades eléctricas y mecánicas, su abundancia y su bajo coste [2].



Figura I.2. - Célula fotovoltaica

Con la intención de hacer los sistemas fotovoltaicos más modulares y versátiles, se han diseñado los módulos fotovoltaicos. Estos agrupan varias células capaces de transformar la energía luminosa incidente en energía eléctrica. Los módulos o paneles fotovoltaicos llegan al mercado en varios tipos de múltiples tamaños y potencias. Son fáciles de unir y de conectar entre sí, con la posibilidad de construir desde pequeños sistemas fotovoltaicos de pocos kilovatios de potencia, hasta grandes sistemas capaces de superar varios cientos de megavatios (Figura I.3).



Figura I.3 - Gran parque fotovoltaico

Aunque profundizaremos en ello en capítulos posteriores, los paneles fotovoltaicos tienen una curva característica (Figura I.4) que define su comportamiento y su potencia máxima de salida en unas condiciones de insolación y temperatura determinadas. En ellas se representa la variación de la corriente que es capaz de suministrar el panel, frente a la tensión entre sus terminales.

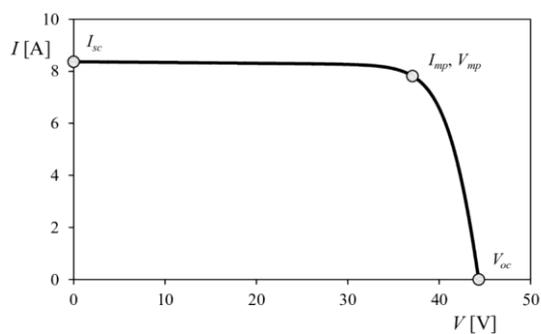


Figura I.4 - Curva característica de los paneles fotovoltaicos

La obtención de esta curva es de vital importancia a la hora de adquirir la máxima eficiencia de la instalación, con la intención de maximizar la potencia que puede ser capaz de suministrar. Para

determinar los puntos de las coordenadas Intensidad-Voltaje que la conforman, se debe utilizar una carga externa variable de tipo resistivo capaz de demandar al panel desde una potencia inicial de cero watios (circuito abierto, carga de  $\infty$  ohmios) hasta llegar al valor máximo que éste pueda suministrar (cortocircuito, cero ohmios) o viceversa. Para simular esta carga, se ha hecho uso de un condensador durante su proceso de carga. Así, partiendo de un condensador inicialmente descargado (con cero voltios entre sus extremos), su resistencia sería nula y cortocircuitaría el panel (máxima corriente). A medida que el proceso de carga continúa, su resistencia aumenta y la corriente disminuye, para terminar con una corriente nula cuando finaliza el proceso de carga (equivalente a un circuito abierto con resistencia infinita).

Durante el transcurso de este proceso, el sistema debe ser capaz de medir y almacenar los valores de voltaje y corriente correspondientes a los datos de la curva característica. Para ello, se digitalizan dos tensiones analógicas. La primera, corresponde a la tensión entre los extremos del panel que, una vez adaptada al rango de operación del Conversor Analógico Digital utilizado (ADC), es digitalizada y almacenada por uno de los dos microcontroladores empleados en el presente proyecto (denominado Maestro).

Un segundo microcontrolador (denominado Esclavo), será el encargado de hacer lo propio respecto a las corriente. Para ello, se hace uso de una "resistencia shunt" que transforma los cambios de intensidad en tensión (con una relación de  $7,5\text{mV/A}$ ) para su digitalización por un segundo ADC.

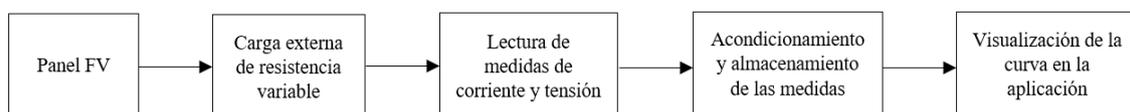
Si bien se han utilizado dos microcontroladores para gestionar, controlar y almacenar las medidas obtenidas de cada uno de sus ADC, será el Maestro el que supervisará todas las acciones propias y las del Esclavo (incluida la comunicación con el usuario), que estará bajo las órdenes del primero, y le enviará sus datos cuando le sean requeridos.

Se ha elegido esta configuración para garantizar que en un mismo instante se obtengan los valores de  $V$  e  $I$  correspondientes a un punto experimental, ya que el empleo de un sólo microcontrolador y un ADC de doble entrada no permitiría muestrear las dos señales en el mismo instante.

Para la visualización de la curva I-V, se ha diseñado una aplicación para ordenador con la que el usuario puede interactuar: ordenar la lectura, visualizar las curvas, obtener los puntos  $I_{sc}$ ,  $V_{oc}$ ,  $P_{max}$ ,  $I_{mp}$  y  $V_{mp}$ , así como ver y almacenar los datos.

El Maestro se podrá conectar al ordenador del usuario mediante USB o Bluetooth, por las que podrá controlar el sistema de adquisición y recibir la información de las medidas realizadas.

En la Figura I.5 se muestra un diagrama de bloques que permite visualizar el flujo que sigue el proceso de medida, desde el panel fotovoltaico, hasta el trazado de la curva en la interfaz.



*Figura I.5 - Diagrama general del proceso*

## **I.2 - Objetivos del proyecto. Descripción general del sistema**

El objetivo del presente proyecto se ha centrado en el diseño y construcción de un dispositivo capaz de trazar dibujar la curva característica de los paneles fotovoltaicos, con la correspondiente medida de tensión y corriente.

El esquema del sistema es simple (Figura I.6) y está compuesto por el dispositivo en cuestión que llevará a cabo el muestreo de los datos, un ordenador conectado vía USB o Bluetooth para visualizar la curva, y uno o varios paneles fotovoltaicos conectados al dispositivo.

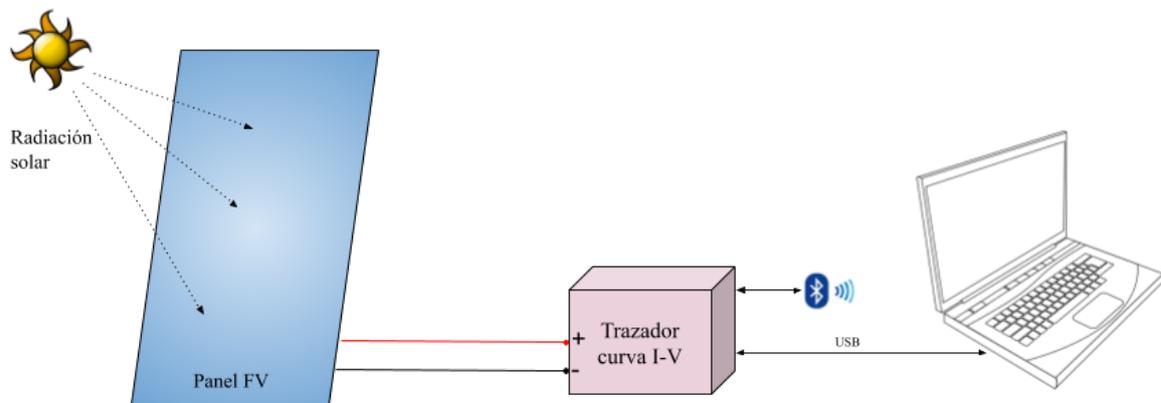


Figura I.6 - Esquema gráfico del sistema

El ordenador tendrá instalado la interfaz gráfica (aplicación *PV Curve Plotter*) realizada para tal fin (que describiremos en capítulos posteriores) mediante la cual, el usuario puede obtener la visualización de esta curva. Dicha orden se enviará al dispositivo de medida que muestreará los datos y, tras el paso de unos segundos, se trazará la curva en la aplicación.

Para la adquisición de los datos de la curva, realizaremos una conmutación del panel fotovoltaico hacia la carga de un condensador. Este condensador servirá como una resistencia variable desde cero ohmios hasta infinito. En el tiempo en el que el condensador se cargue, leeremos y guardaremos los datos pertenecientes a corriente y tensión. Los dispositivos utilizados encargados de gestionar estas lecturas son microcontroladores ATmega328P. Cada uno tendrá conectado un conversor externo analógico-digital para la adquisición de las señales.

Para el desarrollo de este sistema, se ha tomado como punto de partida el diseño por *Chris Satterlee*, bajo el nombre de *IV Swinger 2* [3].

En el mercado, hay varios instrumentos que realizan esta misma función, como el dispositivo mostrado en la Figura I.6.



Figura I.7 - Instrumento comercial trazador de curvas I-V

### **I.3 - Estructura de la memoria.**

Capítulo I – Introducción general: en este capítulo se muestra una visión general del proyecto y se hace una introducción del trabajo realizado, así como sus principales objetivos.

Capítulo II – Paneles fotovoltaicos: se realizará una introducción a la teoría de las células fotovoltaicas, exponiendo sus características y los tipos de módulos que existen. Se continuará con una explicación de las características de la curva I-V fotovoltaica y del motivo de su importancia.

Capítulo III – Características del microcontrolador: se llevará a cabo una justificación del tipo de microcontrolador utilizado y se profundizará en sus características técnicas.

Capítulo IV – Herramientas de diseño: se efectuará una breve descripción de las herramientas de diseño utilizadas, desde la programación del microcontrolador, pasando por el diseño de la aplicación, hasta el desarrollo del circuito electrónico.

Capítulo V – Descripción del sistema: se describirá en profundidad el sistema realizado y los elementos electrónicos que lo conforman. Se explicará, por separado, cada bloque del circuito y su función en la toma de datos.

Capítulo VI – Descripción del software: en este capítulo se definirán las características del software desarrollado para los microcontroladores y la interfaz gráfica.

Capítulo VII – Interfaz gráfica: se explicará el funcionamiento de la aplicación a nivel usuario, y se expondrán todas las funcionalidades que le hemos implementado para un buen uso de la misma.

Capítulo VIII – Resultados experimentales: se ha comprobado el correcto funcionamiento del dispositivo, sometiéndolo a distintas pruebas en diferentes condiciones ambientales y con varios paneles fotovoltaicos, con el posterior análisis de los datos obtenidos.

Capítulo XIX – Presupuesto: se realizará una descripción detallada del coste de este proyecto, tanto de la cuantía de los materiales como de la mano de obra correspondiente.

Capítulo X – Aportaciones y conclusiones: por último, se enunciarán las conclusiones alcanzadas al finalizar el desarrollo de este proyecto. Se añadirán varias aportaciones finales, donde se explicarán los posibles avances y modificaciones que se pueden implementar más tarde, que por cuestión de tiempo y diferentes circunstancias no se pudieron realizar.

## **CAPÍTULO II: PANELES FOTOVOLTAICOS**

## II.1 - Introducción

En este segundo capítulo profundizaremos en la teoría de los módulos o paneles fotovoltaicos. En primer lugar, expondremos brevemente la historia de las células fotovoltaicas, y seguidamente explicaremos su funcionamiento físico, así como los materiales de los que están compuestas. Se indicarán sus características principales, tipos de paneles que se comercializan hoy en día y, por último, terminaremos con una explicación en profundidad de sus curvas características (curva I-V), la cual es el eje sobre el que gira este proyecto.

## II.2 - Teoría de la célula fotovoltaica

Fue en el siglo XIX cuando se descubrió el efecto fotovoltaico (Becquerel, 1839), y se hicieron dispositivos funcionales de selenio (Frits, 1883). No obstante, el primer dispositivo con una capacidad funcional aceptable fue realizada en EE. UU. en el año 1953 por los laboratorios Bell, y se trataba de una célula de silicio de unión PN con un 6% de eficiencia. A lo largo de los años 60, varios científicos especializados en física desarrollaron la teoría fundamental de la célula solar, en la que se incluyeron los materiales idóneos, el espectro de radiación, la acción de la temperatura y de la termodinámica, así como del concepto de eficiencia. En estos años la energía renovable y, especialmente la solar, empieza a tener popularidad, y no fue hasta 1980, que la industria comenzó a madurar y a desarrollar células de unión PN de silicio, con la idea de obtener módulos de tamaño comercial. Hacia los años 90, cuando se consiguieron mejorar los procesos de fabricación, en Europa y América se aumentó el nivel de producción de módulos fotovoltaicos convencionales de silicio cristalino, así como de pequeñas células de a-si (silicio amorfo) para su aplicación en pequeños dispositivos electrónicos (calculadoras, relojes, etc.). Además, durante este periodo, con la llegada al mercado de grandes y eficientes módulos, se lograron construir grandes plantas fotovoltaicas a nivel de megavatios. Este tipo de industria ha sufrido tal evolución hasta el día de hoy que se han conseguido fabricar módulos fotovoltaicos de gran eficiencia y potencia (20%-30%, y más de  $250\text{W}/\text{m}^2$  de potencia) a un precio muy competitivo. Cabe destacar que los fabricantes han diseñado paneles capaces de integrarse en la arquitectura de los edificios, transformados en tejas y cubiertas, incluyendo ventanas y persianas [4].

Por tanto, hoy en día es posible integrar fácilmente estos módulos fotovoltaicos tanto en edificios como en grandes superficies, creando parques solares donde el único cometido es obtener energía a partir de la luz del Sol a un coste muy bajo.

La energía solar fotovoltaica es una tecnología que genera corriente continua por medio de células semiconductoras cuando éstas son iluminadas por un haz de fotones. La estructura típica de la célula solar de unión PN (Figura II.1) es similar a la de los diodos semiconductores.

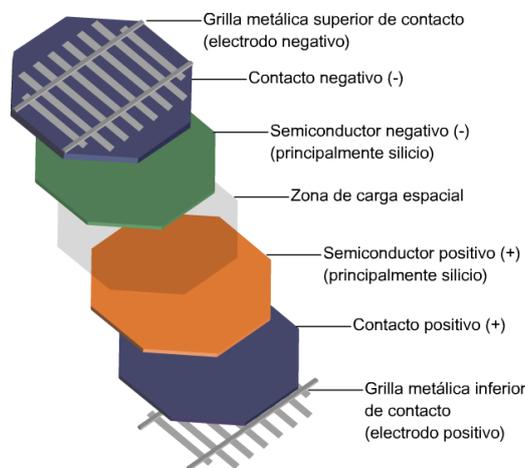


Figura II.1 – Estructura de una célula fotovoltaica

Estas células están hechas de materiales semiconductores que poseen electrones débilmente ligados ocupando una banda de energía denominada “banda de valencia”. Al aplicarse un quantum de energía de cierto valor a un electrón de valencia, el enlace se rompe y el electrón pasa a una nueva banda de energía llamada “banda de conducción”. Estos electrones son transportados a un circuito externo, generando de esta manera una corriente eléctrica, donde pierden la energía y regresan de nuevo a la banda de valencia para adquirir de nuevo la energía que les haga llegar a la banda de conducción [4].

El circuito equivalente de una célula fotovoltaica, mostrado en la Figura II.2, está compuesto por una fuente de corriente continua, un diodo, y las resistencias equivalentes de los cables [5].

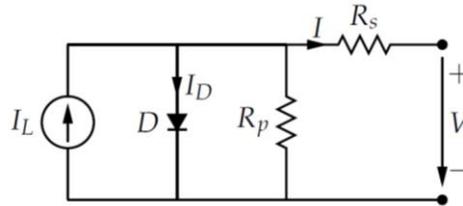


Figura II.2 - Circuito eléctrico equivalente de una célula

En las aplicaciones fotovoltaicas, estas células se interconectan y encapsulan en elementos denominados módulos fotovoltaicos, producto final para ser comercializado. Se suelen realizar varias conexiones de estos módulos para lograr construir desde pequeños hasta grandes parques fotovoltaicos. En estos casos es necesario utilizar dispositivos intermedios entre los módulos y la demanda de energía del usuario, debido que por un lado los paneles sólo son capaces de generar una corriente continua (DC) y de manera usual, la demanda del usuario es en corriente alterna (AC). Por este motivo, debe haber dispositivos que inviertan y controlen la corriente para generar una onda sinusoidal con frecuencia constante.

Por otro lado, también se deben tener en cuenta tanto dispositivos encargados de proteger la instalación y al usuario de posibles cortocircuitos y sobrecargas (fusibles, interruptores, etc.), como de conexiones intermedias de almacenamiento si se requiere la instalación de baterías, ya que si no hay luz solar (por ejemplo, por la noche), los módulos fotovoltaicos no generan energía, y una unidad de almacenamiento puede ser capaz de ofrecer al usuario la energía que demande, si se realiza un correcto dimensionamiento de la misma.



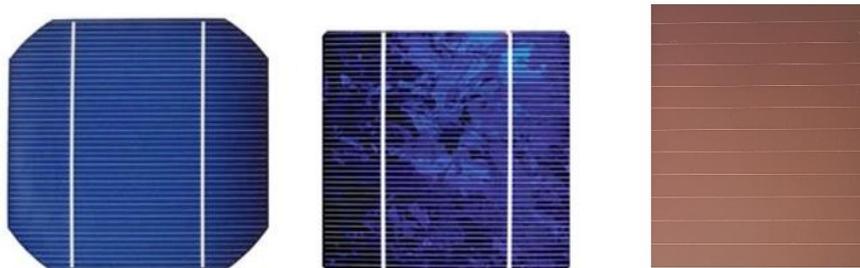
Figura II.3 - Tipos de paneles fotovoltaicos más comunes

“Este tipo de paneles producen electricidad en corriente continua y aunque su efectividad depende tanto de su orientación hacia el Sol como de su inclinación con respecto a la horizontal, se suelen montar instalaciones de paneles con orientación e inclinación fija, por ahorros en mantenimiento. Tanto la inclinación como la orientación, al sur, se fija dependiendo de la latitud y tratando de optimizarla al máximo usando las recomendaciones de la norma ISO correspondiente” [6].

### II.3 - Tipos de módulos fotovoltaicos

Los módulos fotovoltaicos más comercializados son [7]:

- Monocristalinos: estos paneles están compuestos por células monocristalinas. Éstas, a su vez, están formadas por varios materiales entre los que predomina el silicio, encapsulados en un único cristal continuo. Presentan un color azul oscuro muy característico.
- Policristalinos: también llamados multicristalinos, están constituidos por células policristalinas fabricadas a partir de un conjunto de cristales de silicio. Se caracterizan por tener un color azul más intenso y, en comparación con el anteriormente descrito, un menor rendimiento y, por ello, un precio más bajo en el mercado.
- Silicio amorfo: es el menos común en el mercado y están compuestos por células de silicio amorfo. Se caracterizan por tener un alto grado de desorden de átomos en su estructura. Son menos eficientes que las células descritas anteriormente, aunque también más baratas. Este tipo de células es, por ejemplo, el que se emplea en pequeños dispositivos electrónicos como relojes o calculadoras.



*Figura II.4 - Tipos de células: a la izquierda la monocristalina, en el centro la policristalina y a la derecha la de silicio amorfo*

### II.4 - Curva característica

La curva característica de un panel fotovoltaico, o también llamada curva de intensidad-voltaje (abreviadamente curva I-V), representa los valores de tensión y corriente, medidos experimentalmente, de un panel fotovoltaico sometido a unas determinadas condiciones constantes de insolación y temperatura. Variando la resistencia de la carga externa desde cero a infinito, se pueden realizar estas mediciones que conforman la curva característica [8].

En la siguiente imagen se muestran las siguientes curvas: en azul, la variación de la intensidad con respecto al voltaje (I-V), y en violeta, la potencia, que no es más que el producto de la intensidad por la tensión (P-V).

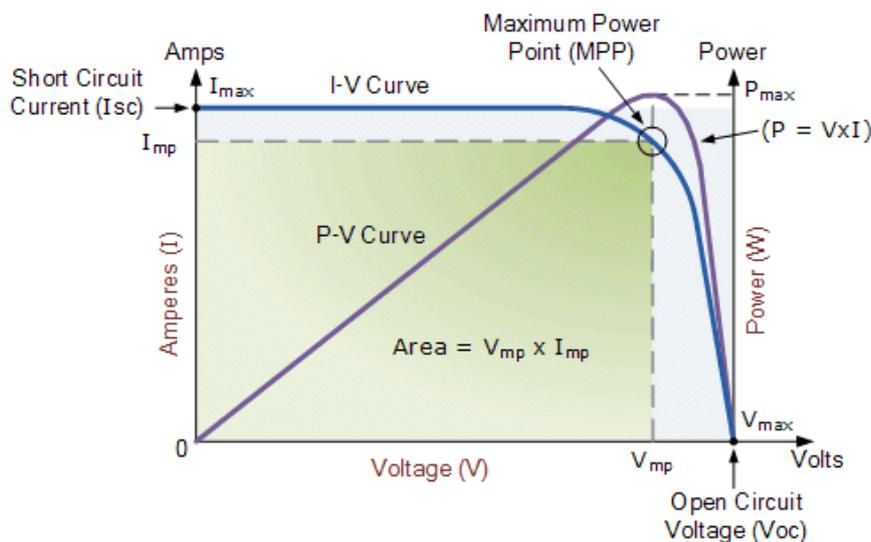


Figura II.5 - Curvas correspondientes a intensidad y potencia respecto al voltaje de un panel FV

Una vez instalados, las condiciones de trabajo reales de los paneles pueden ser muy diferentes a las del laboratorio, por lo que conviene conocer las variaciones que pueden producirse, a fin de efectuar las pertinentes correcciones para maximizar la eficiencia. De este modo, en la instalación pueden producirse sombras que hagan que alguno de los paneles desestabilice la corriente en el resto de los módulos. Además, a lo largo de la vida del panel, alguna de las células puede dejar de funcionar correctamente y no ofrecer la misma energía que sus compañeras. Por lo tanto, estos defectos se podrán visualizar en la curva característica una vez realizada.

Los valores de voltaje, corriente y potencia que cobran especial interés en la gráfica, y que se pueden visualizar en la Figura II.6 son:

- **Voc**, “Open Circuit Voltage”: es el voltaje conocido como de circuito abierto, y corresponde al momento en el que el panel fotovoltaico no está conectado a ninguna carga externa o esa carga tiene una impedancia asociada demasiado alta. Corresponde al valor del voltaje en el que la curva de I-V cruza el eje de abscisas, es decir, el punto donde el corriente se hace cero.
- **Isc**, “Short Circuit current”: es la intensidad máxima que puede ofrecer la célula fotovoltaica. Este punto corresponde al momento en el que la resistencia de la carga llega a un valor de 0 ohmios, o lo que es lo mismo, un cortocircuito entre sus terminales. Corresponde al valor de corriente en el que la curva de I-V cruza el eje de coordenadas, es decir, el punto donde el voltaje es cero.
- **Pmax**, “Maximum Power”: o también denominado MPP (Maximum Power Point), es el punto en el cual la célula fotovoltaica ofrece su máxima potencia. Este valor se corresponde con el pico de la curva de P-V. Con este valor se caracterizan los paneles fotovoltaicos, por su potencia de pico en condiciones STC. Los siguientes valores para describir son obtenidos a partir de las coordenadas de este punto.
- **Vmp**, “Maximum Power Voltage”: también denominado  $V_{pmax}$ , es la tensión correspondiente al punto de máxima potencia  $P_{max}$ .
- **Imp**, “Maximum power current”: o también denominado  $I_{pmax}$ , es la corriente que forma parte de las coordenadas del punto  $P_{max}$ .

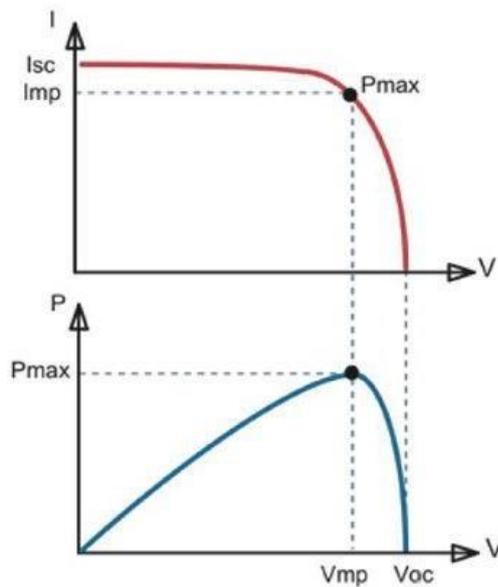


Figura II.6 - Puntos de interés de la curva I-V y P-V

Los regulares encargados de controlar e invertir la corriente eléctrica, también deben ser capaces de regular el punto de máxima potencia *MPP*. De esta manera, si se consigue controlar ese punto bajo todas las circunstancias, la eficiencia del panel será la máxima posible. Si ese punto sufre variaciones, se pueden producir tanto reducciones de corriente como de tensión, algo indeseado por el usuario [9]. De este modo, en el mercado hay reguladores de carga solar de diferentes tipos. Estos son, los reguladores MPPT y PWM.

Los reguladores MPPT maximizan la corriente de salida de los paneles para alimentar las baterías, pues son capaces a adaptar la entrada fotovoltaica a la tensión de éstas. Esto permite a la placa trabajar en su punto de máxima potencia ( $P_{max}$ ) para obtener la máxima eficiencia de producción en cada instante. En cambio, los reguladores PWM únicamente pueden ser utilizados si la tensión de los paneles solares y las baterías es la misma, es decir, con paneles de 24V y baterías de 12V, no se podrá instalar un regulador de carga PWM, la única opción posible será un regulador MPPT.

Por otro lado, la curva de los paneles es dependiente de la irradiancia solar y de la temperatura de las células. La corriente suministrada por el panel es directamente proporcional a la irradiancia solar que recibe en su superficie, es decir, a mayor irradiancia, mayor corriente. Sabiendo esto, las curvas tanto de corriente como de potencia de un mismo panel fotovoltaico se atenuarán en el mismo nivel que se reduce la irradiancia. Este efecto se visualiza en la Figura II.7.

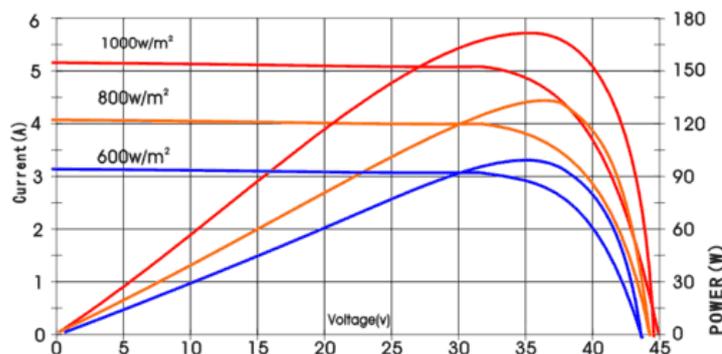


Figura II.7 - Interacción de las curvas con la variación de irradiancia

Además, en un caso semejante al anterior, la temperatura de las células también afecta al rendimiento del módulo solar. En la Figura II.8, podemos observar que, a una mayor temperatura, se obtiene una anticipación de la caída de la curva, disminuyendo los valores de voltaje  $V_{oc}$  y  $V_{mp}$ , y a su vez, el valor del punto de máxima potencia  $MPP$ .

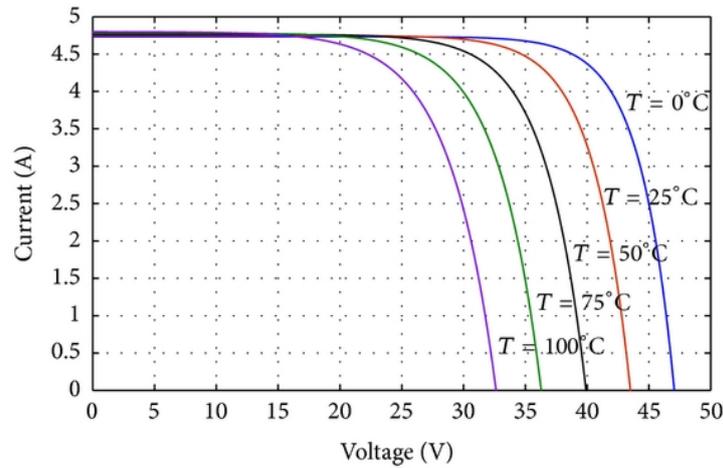


Figura II.8 - Interacción de las curvas con la variación de la temperatura

Por ello, tanto la temperatura como la irradiancia cobran especial importancia a la hora de encontrar el punto en el que se maximice la potencia. Sin embargo, los paneles fotovoltaicos tendrán unas condiciones de trabajo y unos rangos de temperatura e irradiancia limitados, definidos en la hoja de características descrita por el fabricante.

Sin embargo, el punto  $MPP$  y la correspondiente curva I-V sufren distintas variaciones si se realizan varias conexiones de estos módulos en serie o en paralelo. En la Figura II.9, SE observa que, si se realizan conexiones en serie, duplicaremos el voltaje de los paneles, obteniendo la misma corriente resultante. Si las conexiones se realizan en paralelo, encontraremos el caso inverso, donde se sumarán sus corrientes y se mantendrá el mismo voltaje. Esto es debido a que, como ya hemos dicho, las células fotovoltaicas son fuentes generadoras de energía donde se aplica la teoría de circuitos.

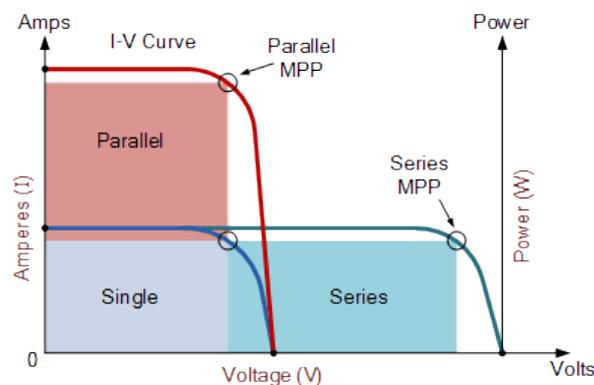


Figura II.9 - Valores del punto MPP con configuración entre paneles serie y paralelo

## **CAPÍTULO III: CARACTERÍSTICAS DEL MICROCONTROLADOR**

### III.1 - Introducción.

En este capítulo se hace referencia al microcontrolador ATmega28P, el principal componente del sistema a nivel de software. Se analizarán sus características y prestaciones a tener en cuenta para la implementación en nuestro sistema.

Los microcontroladores son pequeños y potentes ordenadores de bajo coste. Con un reducido tamaño, pueden estar en el interior de pequeños dispositivos electrónicos con la labor de gestionar su sistema.

El microcontrolador Arduino fue elegido por su capacidad y versatilidad que ofrece para gestionar la información, con un bajo coste.

### III.2 - Características generales de Arduino.

Arduino es una plataforma de creación y desarrollo de electrónica de código abierto. Ésta ofrece una gran variedad de placas de desarrollo, todas ellas basadas en microcontroladores fabricados por Atmel [10]. Por un lado, están las placas basadas en el integrado ATmega328P que es usado en las versiones de Arduino Pro, Mini, Nano y Uno. Por otro lado, están las basados en el microcontrolador ATmega2560, denominado Mega. Generalmente, todas las versiones de Arduino poseen una sola interfaz UART, que es el puerto serie hardware. Con este puerto, el microcontrolador puede comunicarse con un ordenador u otro dispositivo compatible, para establecer una comunicación serial o incluso para programar el microcontrolador.

Todos estos microcontroladores se alimentan a una tensión de 5V, aunque la tarjeta tiene incorporados reguladores lineales de voltaje que pueden transformar tensiones comprendidas entre 5 y 14V a una tensión lineal de 5V, e incluso, 3'3V.

Los microcontroladores de Arduino ofrecen un número determinado de pines de entrada y salida, número que varía en función de la versión. También posee entradas analógicas, integrando para ello un convertor analógico-digital. En cuanto a los pines digitales, permiten ser configurados como entrada o salida a una tensión y corriente máximas de 5V y 40mA, e incluso, establecer varios pines digitales como salida de señales de reloj o PWM. Por último, todos permiten utilizar diferentes protocolos de comunicación que explicaremos en los siguientes apartados.

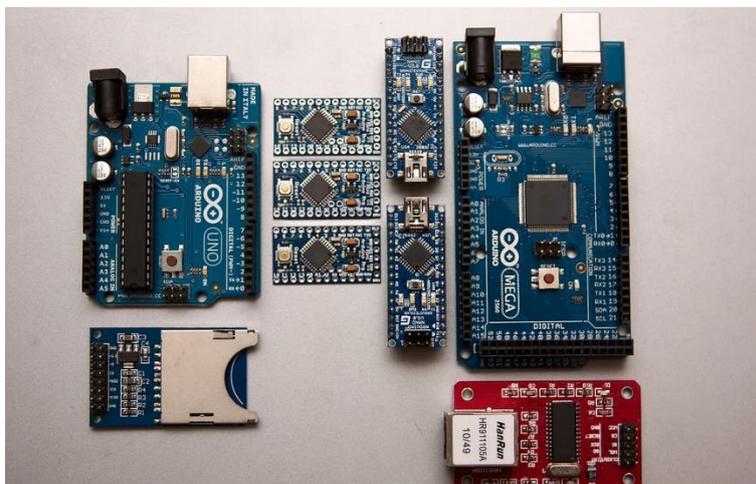


Figura III.1 - Diferentes tipos de placas y módulos de Arduino: Uno, Mega, Nano, etc.

### III.3 - Arduino Uno.

El Arduino Uno es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por *Arduino.cc* [10]. La placa está equipada por varios conjuntos de pines de E/S digitales y analógicas que pueden ser conectados a otros circuitos. Se compone de 14 pines digitales, 6 pines analógicos y programables. Desde la conexión de USB incorporada, se puede adquirir, además de la propia comunicación serial, la alimentación de 5V que necesita el microcontrolador. Esta tensión, como ya comentamos en el apartado anterior, también se puede suministrar mediante una fuente de alimentación externa con voltajes entre 5 y 14V en corriente continua [11]. La versión reducida del *Arduino Uno* se denomina *Arduino Nano*. Ésta tiene un tamaño menor para integrarlo fácilmente en el diseño de un circuito.

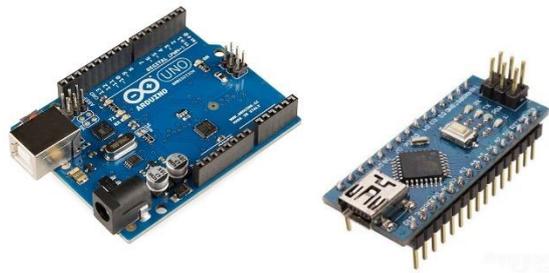


Figura III.2 - Tarjeta de Arduino Uno (izquierda) y Nano (derecha)

En los siguientes subapartados, hablaremos más en profundidad de la tarjeta de Arduino Uno, y de su microcontrolador ATmega328P. Mostraremos sus características principales, así como su patillaje, tipos de memorias, relojes y comunicaciones.

#### III.3.1 - Tarjeta Arduino Uno

Como comentamos anteriormente, la versatilidad de la tarjeta de Arduino es muy amplia. En esta imagen [12] podemos visualizar los diferentes componentes electrónicos que conforman la placa, así como las propias conexiones de entrada/salida.

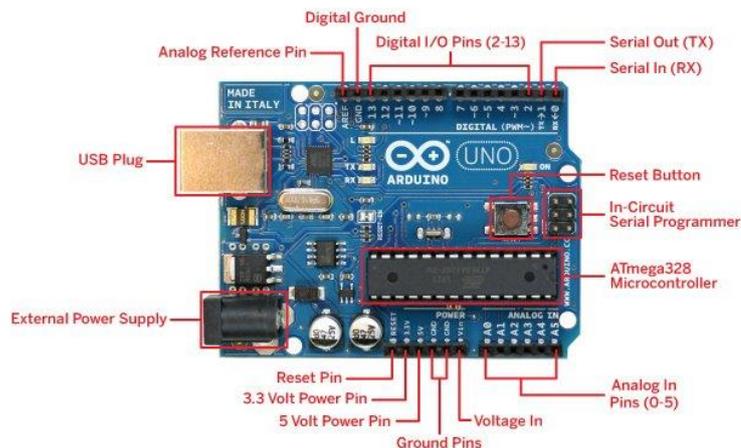


Figura III.3 - Conexiones de entrada y salida de Arduino Uno

Esta tarjeta está gestionada por el microcontrolador ATmega328P. A continuación, hablaremos de sus características.

### III.3.2 - Microcontrolador ATmega328P. Introducción

El ATmega328P es un microcontrolador de alto rendimiento con bajo consumo de energía. Perteneció a la subfamilia “megaAVR”, y está basado en la arquitectura AVR (Arquitectura Harvard 8 bits RISC modificada). Tiene la capacidad de ejecutar grandes instrucciones en un solo ciclo de reloj [13].



Figura III.4 - Encapsulado THT del microcontrolador ATmega328P

A continuación, se describirán algunas características de este microcontrolador.

### III.3.3 - Microcontrolador ATmega328P. Patillaje

El patillaje de este microcontrolador es muy diferente. Los pines que posee tanto de entrada o de salida permiten realizar varias conexiones o comunicaciones diferentes [14].

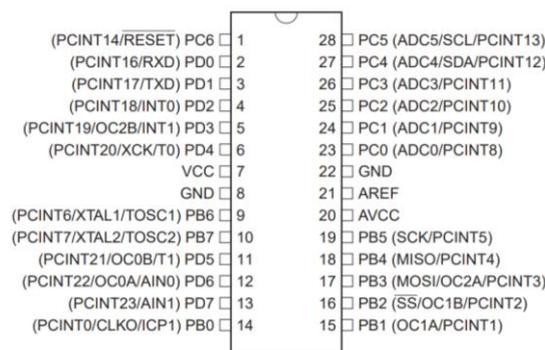


Figura III.5 - Pinout del ATmega328P

### III.3.4 - Microcontrolador ATmega328P. Memoria

El ATmega328P dispone de tres tipos de memorias con diferentes funcionalidades y capacidades [14]. Son estas:

- **Memoria flash (Program Memory):** Memoria de escritura y lectura. Se programa eléctricamente y es la encargada de almacenar las instrucciones que ejecuta el microcontrolador. En el caso del ATmega328P, esta memoria es de 32 KB y se puede reescribir cuando sea necesario.
- **Memoria SRAM (Data Memory):** Memoria de escritura y lectura. Se encarga de guardar los datos obtenidos en cada instante del programa, ya que, una vez se le deja de aplicar tensión, pierde todos los datos almacenados. En el caso del ATmega328P, esta memoria es de 2KB.
- **Memoria EEPROM (Data Memory):** Memoria de escritura y lectura. Se encarga de almacenar aquella información que no queremos que se borre una vez desconectemos el microcontrolador. Esta memoria se programa eléctricamente y se encuentra en un

espacio de datos separados en el cual, se pueden escribir y leer byte a byte. En el caso del ATmega328P, esta memoria es de 1KB.

### III.3.5 - Microcontrolador ATmega328P. Registros

Los registros son memorias de alta velocidad y poca capacidad, integradas dentro del propio microcontrolador. Estos permiten guardar momentáneamente datos que serán usados por el éste, generalmente para realizar operaciones aritméticas o para guardar instrucciones en ejecución o recientemente ejecutadas. El microcontrolador ATmega328P dispone de 32 registros de 8 bits cada uno.

### III.3.6 - Microcontrolador ATmega328P. Reloj

El ATmega328P dispone de un oscilador de cuarzo de 16 MHz. A partir de ese oscilador, parten una serie de señales de reloj internas, encargadas de controlar las velocidades de ejecución de instrucciones, recepción y envío de datos. En la Figura III.6 se muestran estas señales y su distribución.

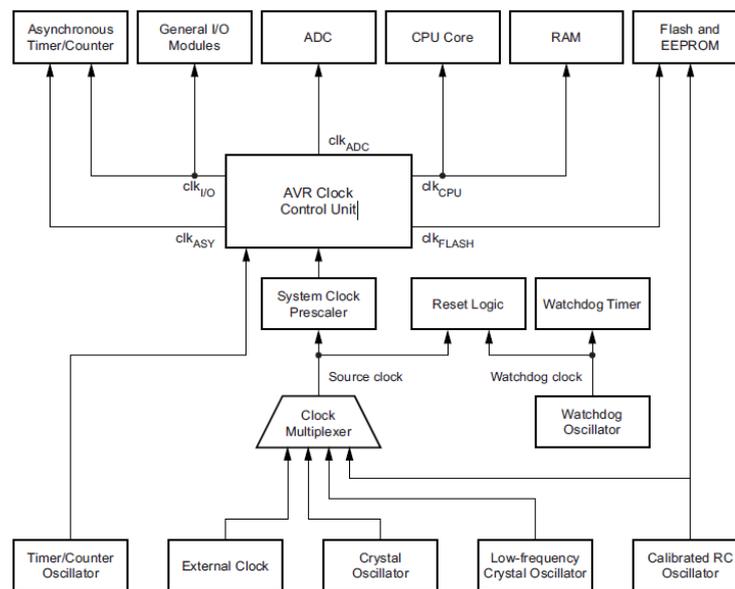


Figura III.6 - Distribución del reloj del ATmega328P

Se puede observar en el diagrama que hay varios tipos diferentes de señales de reloj, y son:

- *clkCPU* (CPU Clock, Reloj de la CPU): es el reloj propio de la CPU del microcontrolador y marca el ritmo a la que se ejecutan las operaciones de la misma.
- *clkI/O* (I/O Clock, Reloj de entrada y salida): es usado por la mayoría de los módulos de entrada y salida, como temporizadores, contadores, SPI, etc.
- *clkFlash* (Flash Clock, Reloj flash): controla las operaciones de la memoria Flash.
- *clkASY* (Asynchronous Timer Clock, Reloj asíncrono): permite que los temporizadores y contadores asíncronos sean controlados por un reloj externo.
- *clkADC* (ADC Clock, Reloj del conversor analógico digital): está dedicado únicamente al conversor analógico digital.

### III.3.7 - Microcontrolador ATmega328P. Interrupciones externas

Para las interrupciones externas (o de hardware) solo hay dos pines dedicados en este microcontrolador. Estas interrupciones son las llamadas INT0 e INT1 que están mapeadas a los pines D2 y D3. Estas interrupciones se pueden configurar en el software con disparadores en Rising o Falling para flancos de subida o bajada, o en nivel LOW. Los disparadores son interpretados por hardware y la interrupción es muy rápida. Permiten alterar el flujo del programa saltando hacia una subrutina del programa. Cuando esta se termine de ejecutar, el programa vuelve de nuevo a donde estaba antes de saltar hacia la subrutina.

### III.3.8 - Microcontrolador ATmega28P. Protocolos de comunicación

Este microcontrolador nos permite usar tres tipos diferentes de protocolos de comunicación distintos, pudiendo elegir entre comunicación en serie o paralelo.

La comunicación en serie (puerto serie, serial o comunicación secuencial) consiste en transmitir los datos bit a bit a través de un único canal o bus. Esto es posible gracias a la UART (Universal Asynchronous Receiver-Transmitter), encargado de gestionar los puertos serie para recibir o transmitir datos. El ATmega328P solo dispone de una UART implementada, y sus pines de comunicación Rx y Tx corresponden respectivamente a los pines D0 y D1 de la tarjeta. Mediante esta conexión serie es posible establecer comunicación entre el microcontrolador y un ordenador, utilizando para ello un conversor Serie-USB. Aunque el microcontrolador solo disponga de una sola conexión serial, si se requiere de otra adicional, es posible crearla vía software, asignando dos pines digitales cualesquiera a Rx y Tx [15].

Por otro lado, la comunicación en paralelo permite transmitir  $n$  bits al mismo tiempo mediante  $n$  canales. Por lo general, este modo de transmisión es más rápido que el anterior, aunque, sin embargo, es más compleja por el gran número de conexiones que se debe tener y por la gran cantidad de pines del microcontrolador que se deben ocupar.

Centrándonos en la comunicación en serie, el microcontrolador es capaz de soportar los siguientes protocolos:

- **I<sup>2</sup>C** (Inter-Integrated Circuit, circuito inter-integrado): es un tipo de bus de datos serial de arquitectura maestro-esclavo (Master-Slave). Permite conectar diferentes dispositivos a una sola línea de comunicación utilizando para ello dos señales, reloj (SCL) y datos (SDA). El ATmega328P puede iniciar una comunicación I<sup>2</sup>C a través de los pines A5 y A4 para SDA y SCL respectivamente [16].

Un posible ejemplo de conexión I<sup>2</sup>C maestro-esclavos es la mostrada en la siguiente imagen (Figura III.7). En ella se puede observar la conexión del maestro que encabeza el bus, y los diferentes esclavos conectados a él mediante las líneas SDA y SCL.

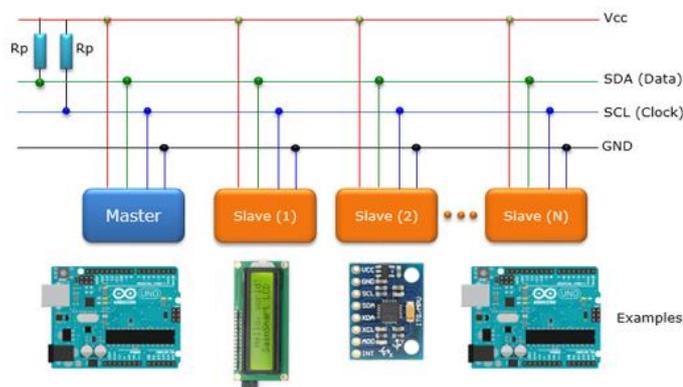


Figura III.7 - Comunicación I<sup>2</sup>C: esquema con conexión de los esclavos a los maestro

Normalmente, el microcontrolador posee la característica propia del maestro y los esclavos serán los dispositivos con los que se comunicará. El maestro puede iniciar la comunicación con los esclavos, además de enviar o recibir datos mientras que los esclavos sólo pueden realizar envíos. Para identificar a los esclavos, es necesario que cada uno tenga asignada una dirección, única dentro de mismo bus. Este tipo de comunicación es síncrona, por lo que se requiere una señal de reloj con el objetivo de mantener los dispositivos perfectamente sincronizados. En la imagen siguiente, observamos la comunicación mediante las líneas SCL y SDA.

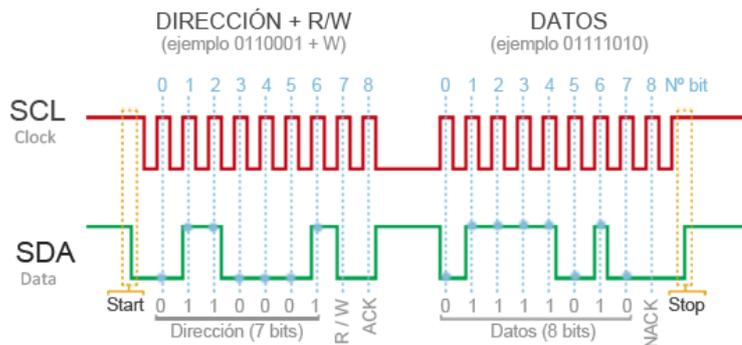


Figura III.8 - Comunicación I2C: señal de datos (SDA) y reloj (SCL)

- **SPI** (Serial Peripheral Interface, Interfaz serial periférica): De igual modo que el bus I2C, el SPI es un método de comunicación basado en protocolo maestro-esclavo. Permite la comunicación simultánea y en ambas direcciones. Para ello se utilizan cuatro señales: *CLK* como reloj para la sincronización, *MOSI* (Master Output, Slave Input), *MISO* (Master Input Slave Output) y *SS* (Select) para indicar que un maestro desea iniciar la comunicación con un esclavo [17].

En la siguiente imagen (Figura III.9) se puede observar el esquema de Maestro-Eslavos con las diferentes conexiones anteriormente descritas para realizar las comunicaciones.

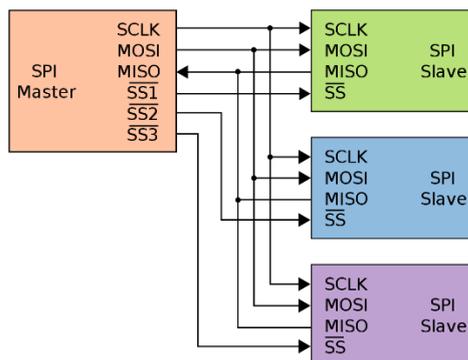


Figura III.9 - Comunicación SPI: esquema con conexión de los esclavos a los maestros

Para comenzar una comunicación, el maestro activa la señal SS (normalmente activa a baja) correspondiente al esclavo y a continuación le envía una orden transformada en una palabra binaria mediante la señal MOSI. El esclavo le responde de la misma

manera, con una palabra binaria, pero esta vez por el canal MISO, manteniendo la sincronización con la señal de reloj. En la siguiente imagen (Figura III.10) observamos esta comunicación.

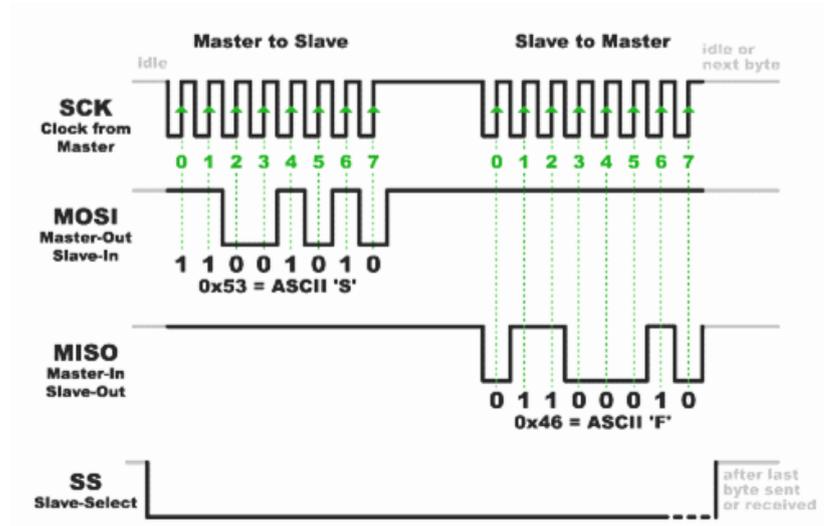


Figura III.10 - Comunicación SPI: señales de datos

## **CAPÍTULO IV: HERRAMIENTAS DE DISEÑO**

## IV.1 - Introducción

Para el diseño y desarrollo de este proyecto, se han utilizado varias herramientas de desarrollo de software, tanto para la programación de los microcontroladores y de la interfaz gráfica, como para el diseño electrónico del circuito. A continuación, haremos una breve descripción de las herramientas utilizadas para tales fines.

## IV.2 - IDE Arduino

El entorno de desarrollo integrado (IDE) de Arduino es una aplicación multiplataforma, disponible para varios sistemas operativos. Está compuesto por un conjunto de herramientas de programación que lo hacen un entorno fácil, accesible e intuitivo a la hora de utilizarlo.

Este IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware a través del puerto serie [18].

Esta aplicación le permite al usuario escribir un código tanto en C como en C++, compilarlo y cargarlo a la tarjeta de Arduino para verificar su funcionamiento. Gracias a su monitor serie, podemos enviar y recibir datos desde la tarjeta hacia el ordenador mediante el sistema de comunicación serie-USB.

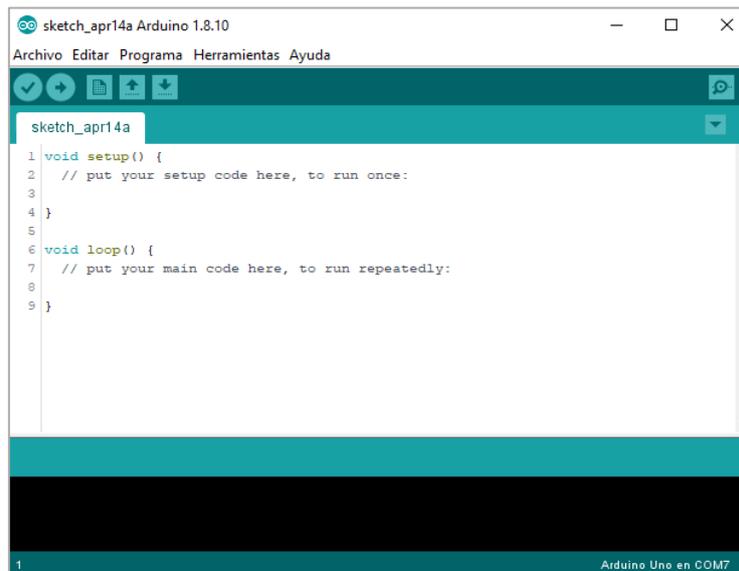


Figura IV.1 - Entorno de programación IDE Arduino

## IV.3 - Anaconda

El entorno de Anaconda es una distribución de código libre y abierto en lenguaje Python disponible para múltiples plataformas. Abarca una serie de aplicaciones, librerías y conceptos diseñados para el desarrollo de la ciencia de datos mediante estos lenguajes. Incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos. Está orientado a simplificar el despliegue y administración de los paquetes de software [19].

Las diferentes versiones de los paquetes se administran mediante el sistema de gestión de paquetes *conda*, el cual lo hace bastante sencillo de instalar, ejecutar y actualizar el software de ciencia de datos.

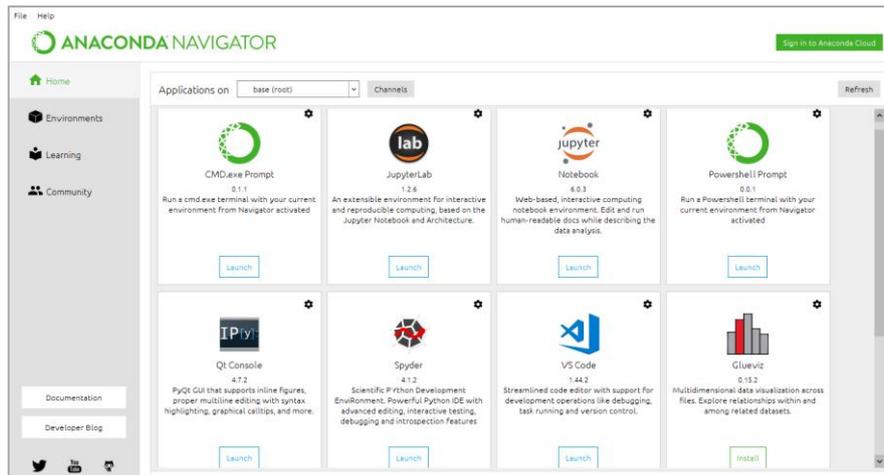


Figura IV.2 - Interfaz de Anaconda

Dentro de este entorno se encuentran dos de las aplicaciones que se han usado en la realización de este proyecto, que son Spyder y PyQt Designer. A continuación, pasamos a explicarlas.

### IV.3.1 - Spyder

Spyder es un entorno de desarrollo integrado multiplataforma (IDE) de código abierto para programación en lenguaje Python. Posee funciones avanzadas de edición, pruebas interactivas, depuración y un entorno informático numérico. Gracias al soporte de IPython (intérprete interactivo mejorado de Python) y bibliotecas populares de Python como NumPy, SciPy, matplotlib o PyQt, este entorno de programación se puede usar como una biblioteca que proporciona widgets potentes relacionados con la consola para nuestras aplicaciones [20].

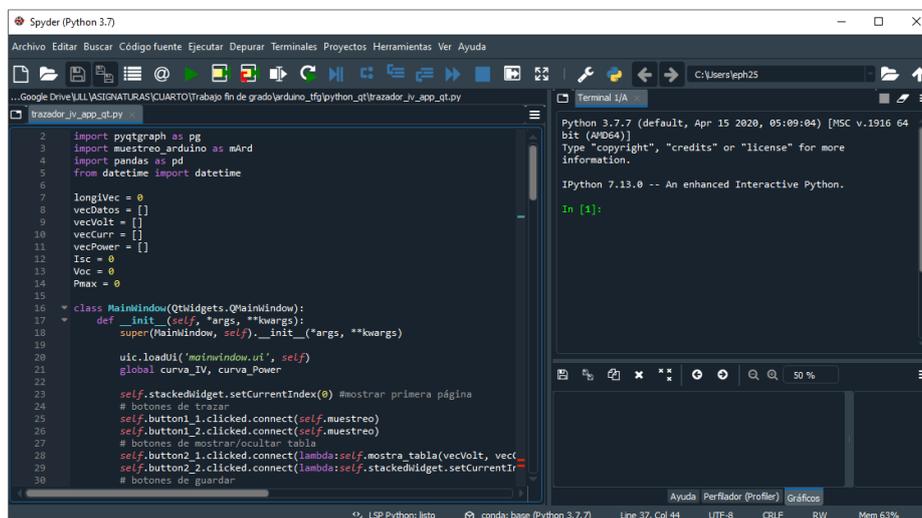


Figura IV.3 - Entorno de programación Spyder

### IV.3.2 - PyQt Designer

PyQt es un kit de herramientas desarrolladoras de widgets para interfaces gráficas (GUI) en lenguaje Python. Dentro de sus funciones, es una de las plataformas más potentes y populares. Para un mejor diseño del sistema, incluye un entorno gráfico e intuitivo con el que poder realizar un diseño de una manera fácil y sencilla. Es posible construir rápidamente una interfaz gráfica sin tener que escribir el código fuente, aunque, por lo contrario, no tiene el poder de mostrar al usuario el código fuente, ni de depurar y compilar la aplicación [21].

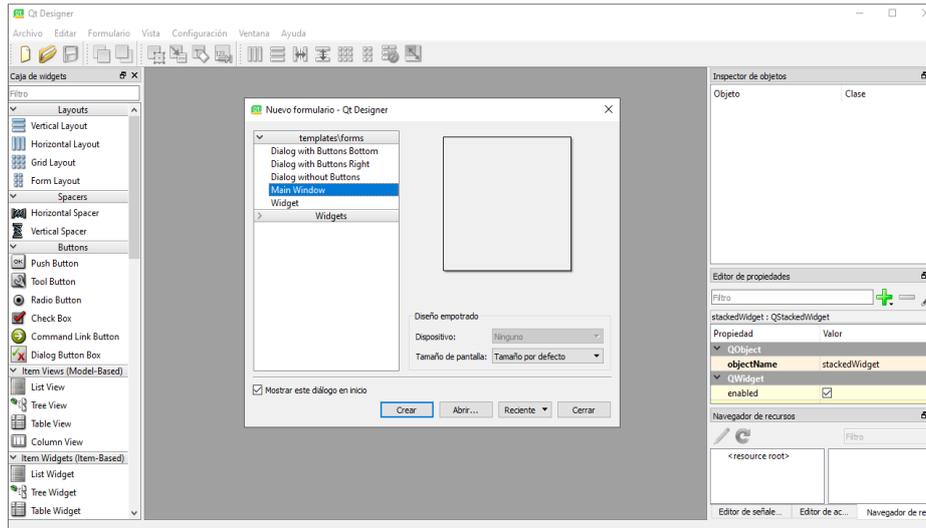


Figura IV.4 - Interfaz de PyQt Designer

La creación de una interfaz gráfica usando Qt Designer comienza con la elección de una ventana de nivel superior para la aplicación. Se pueden elegir añadir varios widgets (pequeñas aplicaciones) a esta ventana creada, editar texto y colores y añadir las distintas funcionalidades.

#### IV.4 - KiCad

La herramienta KiCad es un paquete de software libre creado para el diseño y simulación de circuitos electrónicos. Facilita el diseño de esquemáticos y su conversión a placa de circuito impreso (PCB, Printed Circuit Board) [22].

Cuenta con un entorno integrado para captura esquemática y diseño de PCB. Existen herramientas dentro del software para crear una lista de materiales, ilustraciones, archivos Gerber y vistas 3D de la PCB y sus componentes.

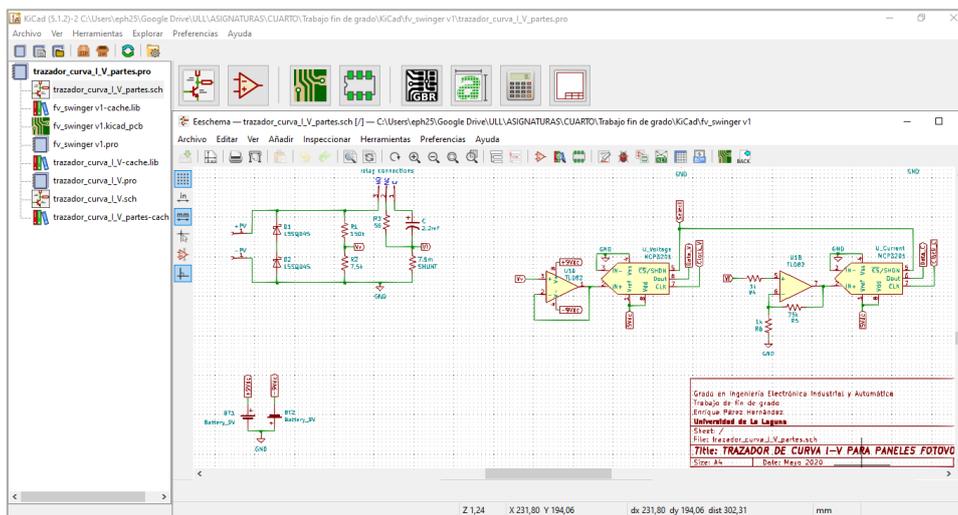


Figura IV.5 - Interfaz de desarrollo KiCad

## **CAPÍTULO V: DESCRIPCIÓN DEL SISTEMA**

## V.1 - Introducción

En este capítulo se realizará una descripción detallada de los bloques y elementos que conforman el sistema del proyecto, así como las características de cada uno de ellos. Comenzaremos con la descripción de los distintos bloques que componen el sistema, continuaremos con los elementos electrónicos utilizados en ellos y terminaremos con el apartado de adquisición de datos, en el que se describirá el ciclo de recolección de los mismos.

## V.2 - Bloques del sistema

El dispositivo de medida diseñado se divide en los bloques mostrados en la Figura V.1. Aunque es un sistema sencillo, se ha decidido separarlos para lograr una mejor explicación.

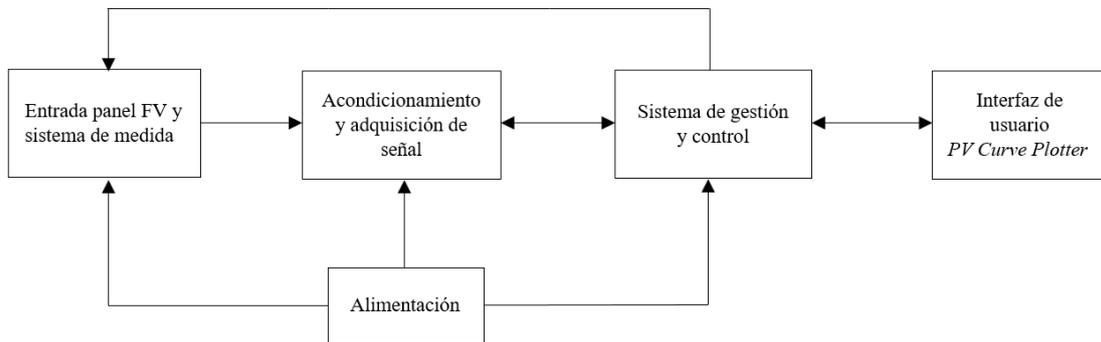


Figura V.1 - Diagrama de los bloques del sistema

A continuación, se hará una descripción más detallada de cada bloque, excepto del de "Interfaz de usuario", dado que se explicará su funcionamiento en profundidad en el Capítulo VII.

### V.2.1 - Entrada del panel FV y sistema de medida

El primer bloque del dispositivo diseñado lo conforma el denominado bloque de "Entrada panel FV y sistema de medida". Su cometido se centra en obtener dos señales de tensión ( $V_v$  y  $V_i$ ) proporcionales a la tensión y corriente, respectivamente, suministradas por el panel o paneles fotovoltaicos que se conectan a su entrada (terminales PV+ y PV- de la Figura V.2). Los diodos Schottky DS1 y DS2 impiden que la corriente circule en sentido contrario al que el circuito ha sido diseñado si, involuntariamente, se realiza una conexión inversa del módulo fotovoltaico.

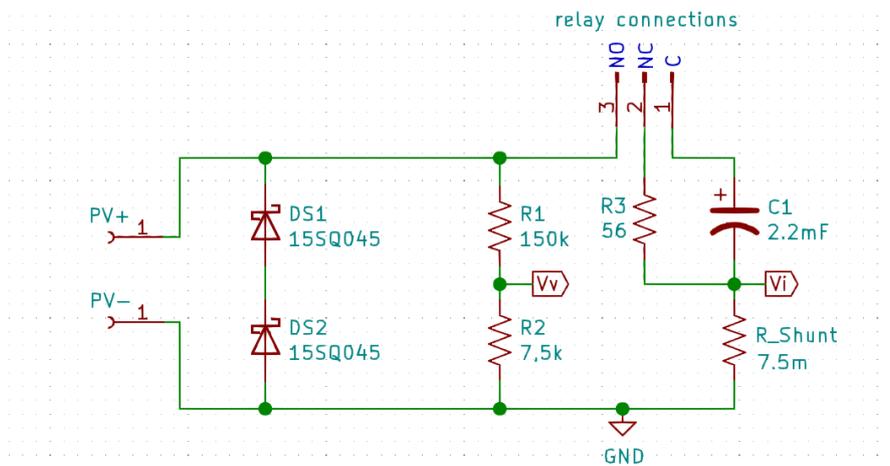


Figura V.2 - Entrada panel FV y acondicionamiento de señal

Teniendo en cuenta que la tensión máxima de entrada para la cual se ha diseñado el circuito es de 100V y, por otro lado, la máxima que soportan los ADC es de 5V, se ha hecho necesario reducir el voltaje en un factor de 20, recurriendo para ello al uso de un partidor de tensión (resistencias R1 y R2 de las Figuras V.2 y V.3) resultando 150kΩ y 7,5 kΩ, respectivamente, para R1 y R2. De esta manera,  $V_v$  será llevada a uno de los ADC, proporcionándonos la tensión de salida de los paneles fotovoltaicos.

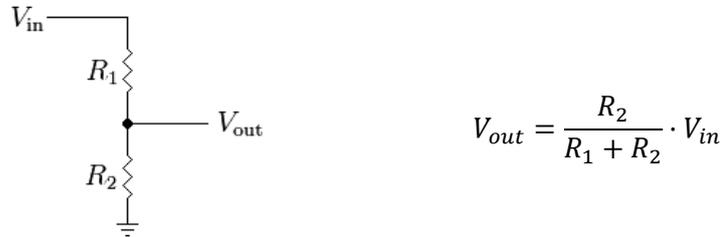


Figura V.3 - Partidor de tensión

En cuanto a la medida de la corriente, se ha implementado una resistencia shunt que proporciona una tensión de 7,5mV/A (salida  $V_i$ , Figura V.2), hasta un máximo de 10A, que será amplificada mediante un amplificador operacional en configuración no inversora en una etapa posterior para adaptarla a la entrada del segundo ADC.

Por último, la parte más importante de este bloque lo conforman el condensador C1 y el relé (Figura V.2). En reposo, los terminales C (Común) y NC (Normalmente Cerrado) del relé están unidos, de forma que C1 se habrá descargado a través de la resistencia de 56 Ω.

Cuando se desea realizar una medida, el microcontrolador Maestro enviará una señal que activará el relé, de modo que, el terminal Común del mismo conmutará para conectarse al NO (Normalmente Abierto). En ese instante, la tensión procedente de los terminales de los paneles FV comenzará a cargar el condensador C1 partiendo de cero voltios entre sus extremos, mientras su resistencia sería nula y cortocircuitaría el panel (máxima corriente). A medida que el proceso de carga continúa, su resistencia aumenta y la corriente disminuye, para terminar con una corriente nula cuando finaliza el proceso de carga (equivalente a un circuito abierto con resistencia infinita).

Durante el tiempo que dura este proceso de carga, se realizan todas las tomas de datos con los conversores ADC, tanto de corriente como de voltaje, recogiendo los valores de señal de las salidas de  $V_v$  y  $V_i$  (Figura V.2). Cuando se termina de realizar la medida, el relé vuelve a su estado de reposo, descargando así el condensador y dejándolo listo para realizar otra medida.

La Figura V.4 muestra el aspecto real del bloque de entrada, donde se encuentra la conexión con la señal fotovoltaica (PV+ y PV-), los diodos Schottky (DS), el condensador de 2,2mF, el relé que permite la conmutación el partidor de tensión, la resistencia shunt y las salidas  $V_v$  y  $V_i$ , que serán llevadas a sus correspondientes ADC.

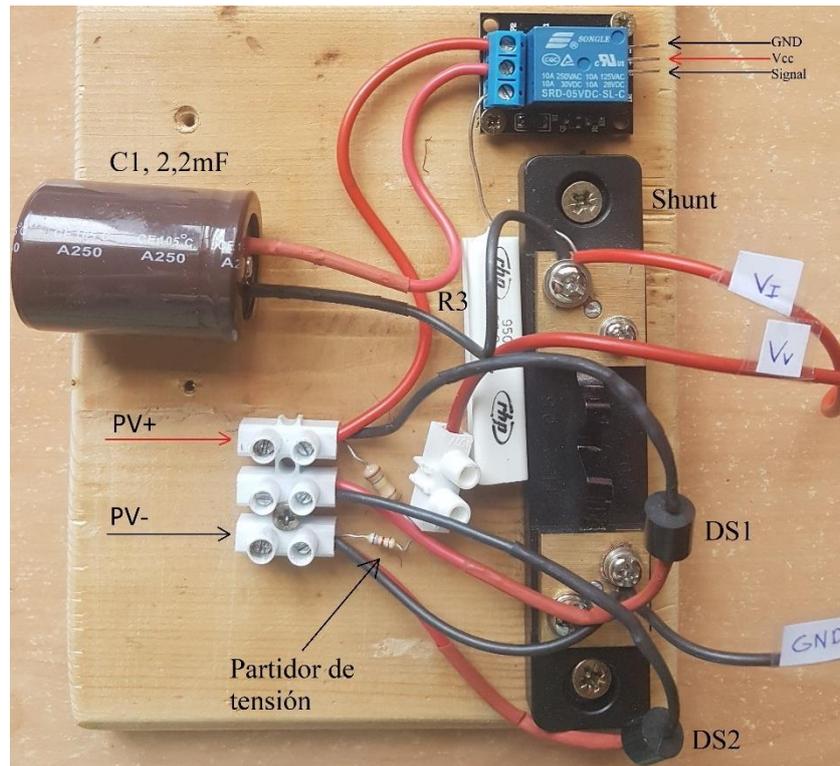


Figura V.4 - Imagen real de la entrada FV con los dispositivos de carga y medición

El tiempo de carga y descarga de un condensador depende de la constante de tiempo, que es el producto de la capacidad por la resistencia en serie asociada, es decir,  $\tau = C \cdot R$ . Se dice que, en este tiempo, el condensador se ha cargado un 63,2%, y en 5 veces este tiempo, se habrá cargado prácticamente en su totalidad [23]. La curva de carga es exponencial, y se puede visualizar en la Figura V.5.

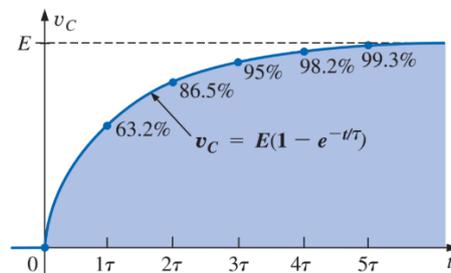


Figura V.5 – Curva de carga del condensador

Por lo tanto, para calcular el valor del condensador en este circuito, hay que tener en cuenta, por un lado, la resistencia parásita ESR del condensador y, por otro lado, la resistencia de los cables, la de los terminales del relé y la resistencia shunt. Conociendo esto, nos ha resultado un condensador de 2,2mF, capaz de cargarse en su totalidad en un tiempo aproximado de 50ms.

### V.2.2 - Acondicionamiento y adquisición de señal

La parte de acondicionamiento y amplificación de señal es de vital importancia a la hora de recoger y almacenar mediciones desde sensores. En este caso, se ha implementado un amplificador operacional doble (TL082, descrito en el apartado 3.3 de este mismo capítulo) en configuración amplificador no inversora para aumentar la ganancia de la señal mediante uno de sus operacionales, y como seguidor de tensión a través del otro, con la intención de eliminar ruidos o perturbaciones que se puedan originar en el circuito. Este amplificador doble acondiciona las

señales medidas  $V_v$  y  $V_i$  provenientes del bloque anterior, para ser llevadas a sus respectivos conversores ADC, los cuales transformarán los valores de las señales de analógicas a digitales.

En la siguiente imagen (Figura V.6) se puede visualizar el esquema electrónico correspondiente a este bloque. En la zona izquierda, se encuentra la parte de acondicionamiento y adquisición de la señal  $V_v$  (voltaje dependiente de voltaje) y en la parte derecha, la perteneciente a  $V_i$  (voltaje dependiente de corriente).

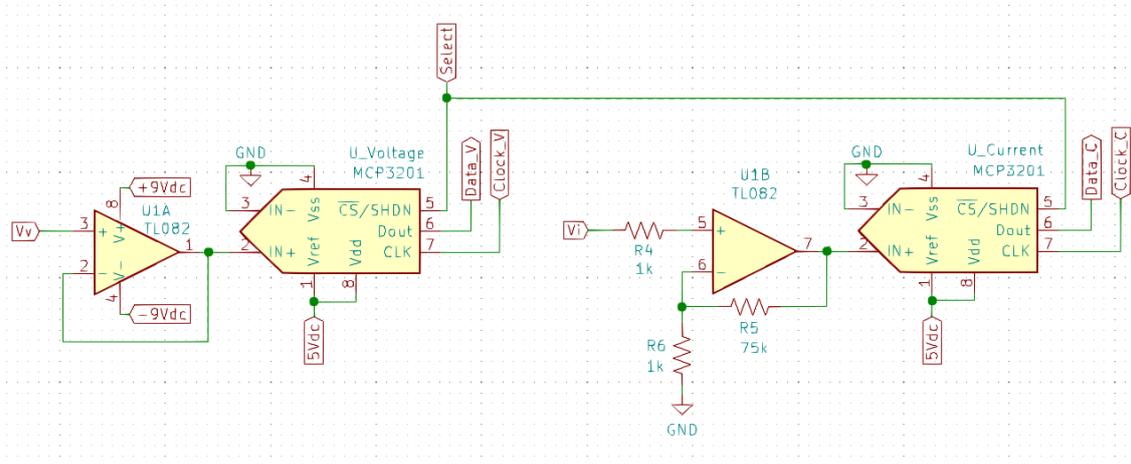


Figura V.6 - Acondicionamiento de señal y conversión A/D

Como máximo, la señal proveniente de la resistencia shunt es de 75mV correspondientes a 10A, voltaje demasiado ínfimo para ser aprovechado por el conversor ADC. De esta manera, se le debe aumentar la ganancia en 66,7 para que esos 75mV correspondan al voltaje máximo que el conversor puede leer, que son 5V. Por lo tanto, la ganancia del amplificador debe ser:

$$G = \frac{5V}{75mV} = 66,7 \rightarrow V_{ADC} = G \cdot V_i \rightarrow G = 1 + \frac{R_5}{R_6} = 1 + \frac{75k\Omega}{1k\Omega} = 75$$

Con estos cálculos se han obtenido, mediante valores comerciales de resistencias, la ganancia del amplificador, que será de  $75_{v/v}$ , aunque sufrirá pequeñas variaciones por la tolerancia de las resistencias.

Una vez que  $V_v$  y  $V_i$  han pasado la parte de amplificación de señal, entran en la de adquisición de señal. Para ello, se han utilizado dos conversores ADC MCP3201 (explicados en el apartado 3.1 de este capítulo), cada uno perteneciente a una de las dos señales, para que puedan ser convertidas y leídas por separado y en el mismo instante de tiempo. A su vez, cada conversor pertenece a un microcontrolador, es decir, el conversor de  $V_v$  está conectado al microcontrolador Maestro, y el de  $V_i$ , al Esclavo.

Para asegurar que los dos ADC convierten las señales en el mismo instante de tiempo, eliminamos el posible desfase entre las medidas de intensidad y voltaje del mismo punto muestreado, empleando una única señal de *Select* (activación del conversor) proveniente del microcontrolador Maestro, que servirá de señal de activación de entrada a los dos conversores. Sin embargo, las salidas digitales de cada conversor, es decir, *Data* (*Dout*, *MISO*) y *Clock*, pertenecerán a su propio microcontrolador. La señal *Data* transporta hacia el microcontrolador el valor de la palabra digital del valor muestreado, y el reloj *Clock* es enviada desde el microcontrolador, y marca la velocidad de lectura y envío de esa palabra.

### V.2.3 - Gestión y control de la medida

El último bloque del circuito corresponde con el control y gestión de la medida, encargado de procesar y almacenar la información. Para ello, se han utilizado dos microcontroladores, por el

motivo de que se debe leer y guardar los valores de los dos conversores al mismo tiempo mediante comunicación SPI. Esta labor no es posible realizarla si solo se utiliza un único microcontrolador.

El circuito implementado para gestionar el microcontrolador utilizado (ATmega328P) es el mostrado en la siguiente imagen (Figura V.7):

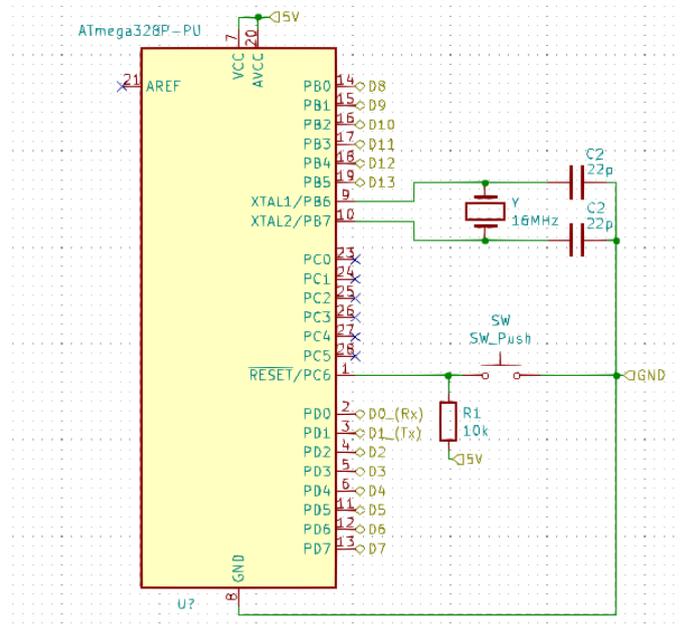


Figura V.7 - Circuito de control de ATmega328P

Se pueden visualizar las conexiones requeridas para que el microcontrolador funcione de igual manera que en la tarjeta de Arduino, con su oscilador de 16MHz, botón de reset, alimentación de 5V, tierra GND, y pines digitales de entrada y salida. Los pines analógicos no los utilizamos en este proyecto, ya que los hemos sustituido por los conversores ADC externos. Estos conversores proporcionan dos bits más de medición (10 bit, 1024 niveles frente a 12 bit, 4096 niveles). En el circuito final del dispositivo deben encontrarse dos circuitos como el que mostramos en la imagen anterior: uno corresponderá al maestro, encargado de gestionar y controlar la lectura, y otro al esclavo, con la única labor de leer y guardar los datos en el momento que el maestro lo ordene, y enviárselos cuando se los pida.

En la siguiente imagen (Figura V.8) se puede visualizar los dos microcontroladores con las correspondientes señales y conexiones entre ellas dos, y, además, los módulos de transmisión de datos mediante Bluetooth y USB. Por otro lado, también se observa la conexión de la señal del relé *SigRelay* conectada al pin D5 del maestro, y la conexión de la señales pertenecientes a cada uno de los conversores (*Data* y *Clock*) en los pines D12 y D13 del maestro en el caso del ADC de voltaje, y en los mismos pines del esclavo, en el caso del de corriente. La señal *Select* corresponderá a una salida desde el pin D10 del maestro. Ésta servirá de entrada de activación de los dos conversores por igual, y para que el esclavo lea los valores de su ADC, será una señal de entrada por su pin D5.

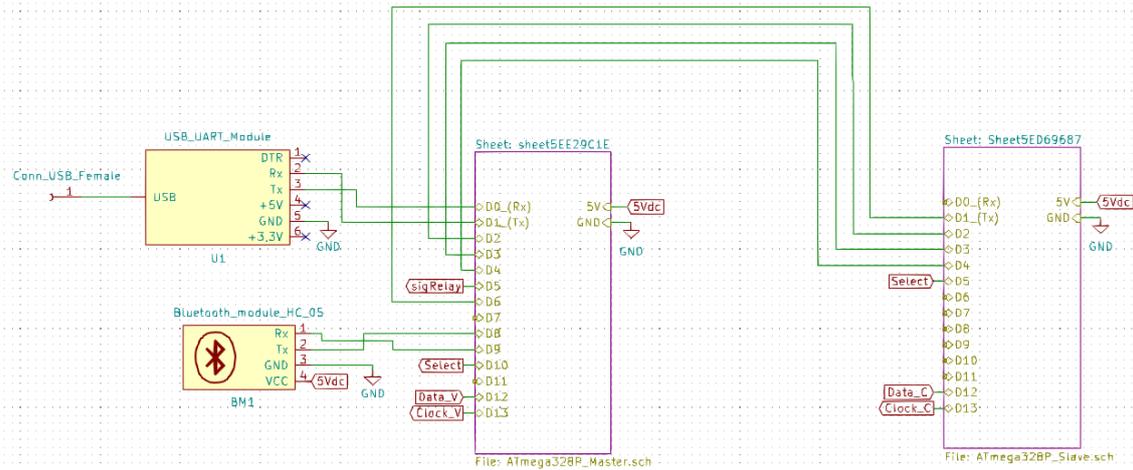


Figura V.8 - Conexión de microcontroladores y módulos de comunicación

Como debe establecerse una comunicación entre los dos microcontroladores, se han empleado dos canales distintos: por un lado, conexión serial (Rx-Tx) y por otro, algunos pines digitales conectados entre sí. La comunicación serial solo es utilizada para enviar los datos en bytes desde el esclavo hasta el maestro. Esta conexión se realiza desde el puerto D1/Tx del esclavo hasta un nuevo puerto serial Rx creado en el pin D6 del maestro. Por otro lado, la comunicación también se realizará mediante tres conexiones, en los que los pines D2, D3 Y D4 de cada tarjeta están conectados entre sí, es decir, el pin D2 del maestro con el D2 del esclavo, y así con las dos conexiones restantes. En la Tabla V.1 explicamos las funciones de estas conexiones.

Pines (M = master S = slave)	Estado a 0	Estado a 1
D2-M → D2-S	Modo reposo, no muestreo	Se ha recibido una señal desde la aplicación de que se debe empezar a muestrear, por lo que el maestro envía esta señal al esclavo para inicializarlo y que esté atento a la señal <i>Select</i> . Estará a 1 todo el tiempo que dure el proceso de muestreo de los más de 200 puntos.
D3-S → D3-M	-	El esclavo ha terminado de muestrear, por lo que avisa al maestro de que está preparado para enviar los datos
D4-M → D4-S	-	El maestro, si está preparado para recibir los datos, y la señal anterior está activada, avisa al esclavo de que envíe los datos.

Tabla V.1 - Órdenes maestro-esclavo mediante pines

A su vez, la comunicación del maestro con el ordenador se puede realizar de dos maneras distintas, mediante USB o Bluetooth. En los puertos D0-Rx y D1-Tx se ha conectado un dispositivo capaz de comunicarse mediante USB (utilizando la conversión serial UART-USB) a un ordenador. Éste recibe el nombre de *CP2102* (se explicará en profundidad en el apartado 3.8 de este capítulo). También se ha implementado otro modo de comunicación mediante un dispositivo capaz de emitir una señal Bluetooth. Éste, que recibe el nombre de *HC-05* (se explicará en el apartado 3.7), estará conectado a los pines D8 y D9 del microcontrolador maestro. Para configurar estos pines, se ha empleado la función de Arduino para crear conexiones seriales (Rx-Tx) mediante software, en otros pines de la tarjeta, a parte de los que trae ya implementados en D0 y D1.

La imagen del circuito implementado en la realidad se puede visualizar en la Figura V.9. En ella, se observa las conexiones de entrada  $V_v$  y  $V_i$  pertenecientes a las salidas del primer bloque “Entrada del panel FV y sistema de medida”. Estas señales llegan como entrada al segundo bloque, “Acondicionamiento y adquisición de señal” y por último el bloque perteneciente a este aparatado, “Gestión y control de la medida” con sus respectivos microcontroladores.

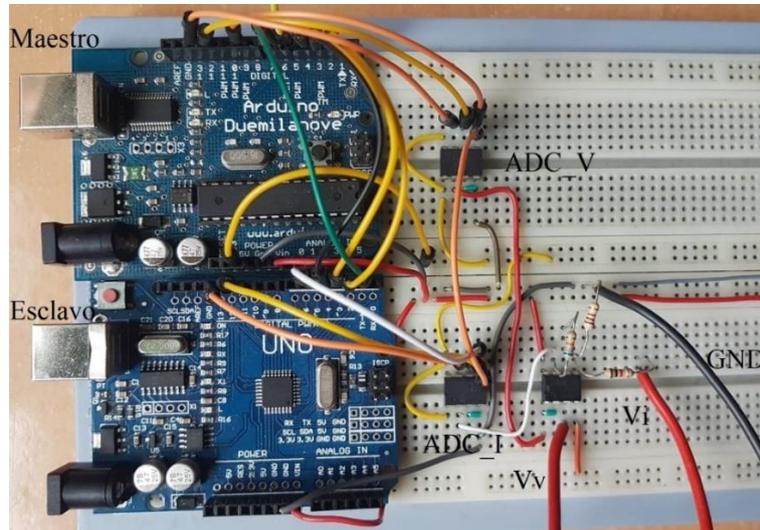


Figura V.9 - Imagen real conexión del circuito

#### V.2.4 - Alimentación

El dispositivo trazador está diseñado para usarlo en labores de campo, por lo que debe ser alimentado mediante pila o batería. Se ha decidido utilizar una pila de 9V, la cual será la principal fuente de alimentación del dispositivo. A partir esta tensión, obtendremos los voltajes necesarios mediante convertidores y reguladores, para alimentar el dispositivo. En la siguiente imagen (Figura V.10), podemos visualizar en la parte izquierda la conexión de la pila al convertidor Buck-Boost, y a continuación, al regulador de tensión utilizado, con las correspondientes salidas.

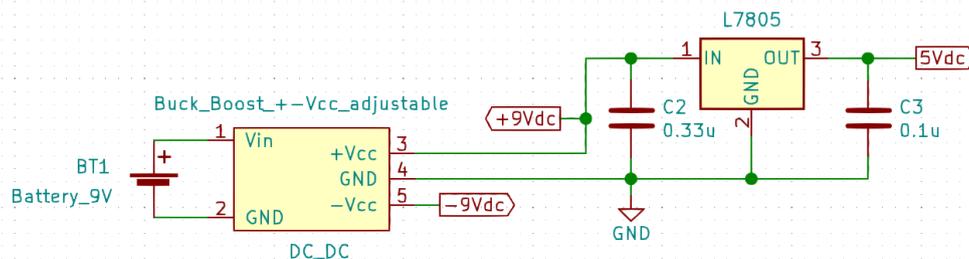


Figura V.10 - Bloque de alimentación

Se ha elegido un convertidor Buck-Boost (Figura V.12) para transformar la tensión de la pila, pues era necesario una tensión de alimentación de  $+V_{cc}$  y  $-V_{cc}$ , ( $\pm 9V$ ).

Cabe destacar que con las salidas de  $+9V_{dc}$  y  $-9V_{dc}$  alimenta al amplificador operacional, aunque los microcontroladores y sus módulos solo admiten 5V, por lo que es necesario un regulador de tensión para realizar una alimentación adecuada. Se ha elegido el regulador LM7805, capaz de controlar tensiones a 5V de salida. Como filtros y siguiendo el datasheet, se ha optado por conectar dos condensadores al regulador, uno de  $0,33\mu F$  a la entrada y otro de  $0,1\mu F$  a la salida, con la intención de eliminar ruidos y perturbaciones.

### V.3 - Elementos Electrónicos

Los elementos electrónicos que forman parte de los bloques del sistema descritos en el apartado anterior, los describiremos a continuación.

#### V.3.1 - Conversor ADC

Para convertir las señales analógicas en palabras digitales, se utilizan los conversores ADC. En nuestro sistema, se ha utilizado el integrado MCP3201, mostrado en la Figura V.11.

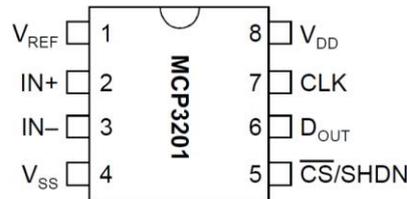


Figura V.11 - Pinout del ADC MCP3201

Este conversor tiene la posibilidad de convertir una única señal analógica a digital con una resolución de 12 bit (4096 niveles) y una tecnología de comunicación serial SPI. Con un rango de voltajes de trabajo y alimentación ( $V_{DD}$ ) que van desde 2.7 a 5.5 Vdc, y una corriente de trabajo de aproximadamente 300  $\mu$ A [24], tiene una velocidad de conversión máxima de 100 ksp/s (100 mil muestras por segundo). Si se alimenta a 5V, tendrá una precisión en la medida de 1,22 mV por nivel. En la siguiente imagen (Figura V.12) se observa el diagrama funcional de este dispositivo, con las entradas y salidas correspondientes, y el esquema del ciclo de conversión.

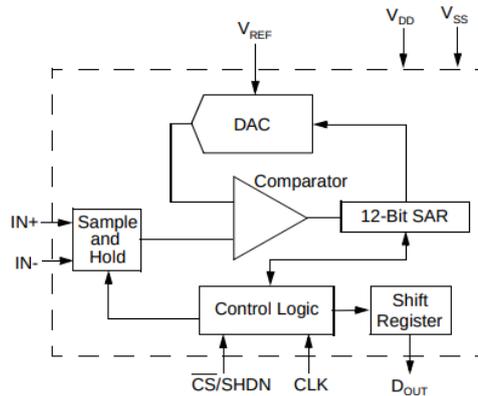


Figura V.12 - Diagrama funcional del MCP3201

Se debe tener en cuenta que no se necesita la línea MOSI (pin 11 de Arduino UNO) del bus SPI, ya que la comunicación es suficiente con la línea CS de activación y la señal de reloj CLK del bus. El microcontrolador no le enviará datos al conversor, sino simplemente con un flanco de bajada de la línea CS, el ADC realiza la conversión y posteriormente le envía los datos al microcontrolador mediante la línea MISO (Dout).

La siguiente imagen (Figura V.13) muestra el diagrama de tiempos de las líneas CS, CLK y Dout.

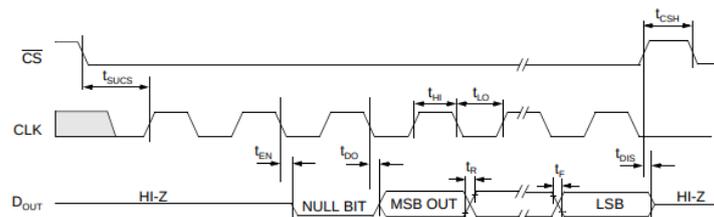


Figura V.13 - Diagrama de tiempos del MCP3201

El microcontrolador baja la línea CS para iniciar la comunicación con el MCP3201. En este punto, la señal de reloj generada por el microcontrolador se utiliza para iniciar la conversión analógico-digital. La siguiente imagen (Figura V.14) ilustra en detalle este ciclo de conversión y transferencia de datos.

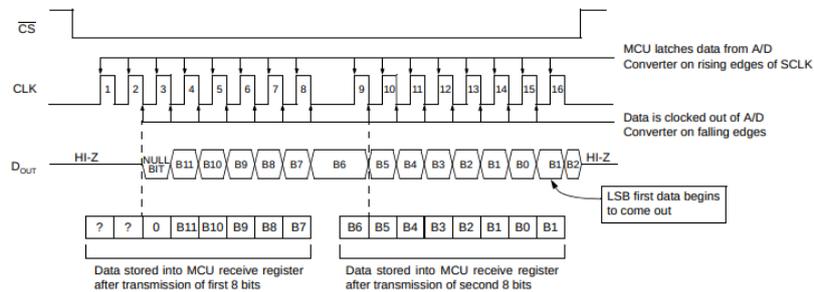


Figura V.14 - Ciclo de conversión y transferencia de datos de MCP3201

Se puede observar que, a partir del tercer pulso de la señal de reloj, se obtiene el bit más significativo (MSB) de los datos digitales en la línea D<sub>out</sub>. Los siguientes pulsos de la señal del reloj permitirán obtener los bits restantes de la palabra digital.

A la hora de recoger y reconstruir los datos digitales de la conversión vía software, se deberán leer primero los 8bits más significativos y luego los 8bits restantes, haciendo una suma lógica y teniendo en cuenta que el bit B1 se repite en el decimosexto reloj. Este bit debe eliminarse desplazando toda la palabra una posición a la derecha [25].

Para terminar la lectura, el microcontrolador coloca la línea CS en el nivel más alto. De esta manera, se puede volver a realizar una nueva conversión.

### V.3.2 - Medida de corriente. Resistencia Shunt

Una resistencia shunt es una carga resistiva de valor mínimo a través de la cual se deriva una corriente eléctrica. Es utilizada para determinar con precisión la intensidad de corriente que fluye a través de un circuito mediante la medición de la diferencia de potencial generada entre sus extremos.

La shunt utilizada (Figura V.15) en este dispositivo tiene un valor de resistencia de 7,5mΩ y está diseñada para soportar una corriente máxima de 10A. Por lo tanto, por cada amperio que circule por ella, generará una diferencia de potencial de 7,5mV [26].



Figura V.15 - Resistencia shunt

### V.3.3 - Amplificador operacional

El TL082 es un amplificador operacional doble (Figura V.16), con dos amplificadores operacionales separados en su interior, de 8 pines y una tensión de alimentación de  $\pm 15V$ . Tiene un Slew Rate de  $13mV/us$  y un ancho de banda de  $3MHz$  [27].

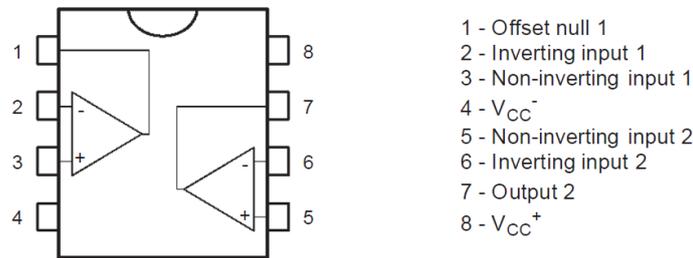


Figura V.16 - Pinout del TL082

Este operacional se ha utilizado con la misión de amplificar la señales provenientes de los parámetros medidos (voltaje y corriente) y eliminar los posibles ruidos que se pudieran generar. Es el paso previo a la conversión analógico-digital.

### V.3.4 - Módulo de conmutación

Un módulo de conmutación (o también comúnmente llamado, relé) es utilizado para intercambiar señales o conexiones entre varios terminales.

El módulo de conmutación de potencia SRD-05VDC-SL-C [28] es el escogido para este proyecto. Es apto para Arduino gracias a su módulo de fácil montaje (Figura V.17) en el que la electrónica de control ya la tiene implementada.

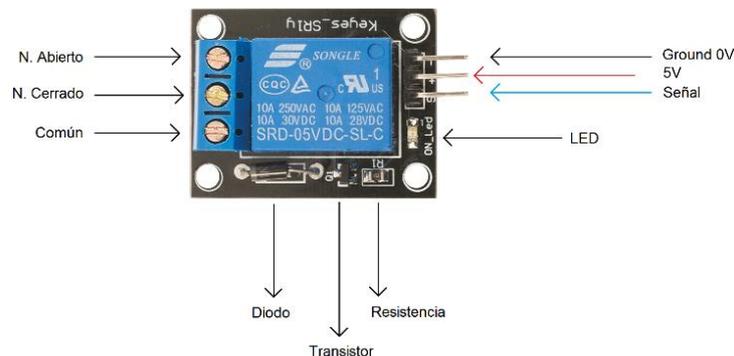


Figura V.17 - Módulo de conmutación

Como se puede observar en la anterior imagen, tiene una tensión de alimentación de  $5V$  en DC y una señal de activación a la misma tensión, por lo que si la señal es  $0V$  (LOW) el relé estará no conmutado, en reposo, y si la señal es  $5V$  (HIGH) el relé realizará la conmutación. Es apto tanto para circuitos en AC como en DC. Tiene una corriente y tensión máxima en AC de  $10A/250VAC$ , y en el caso de DC,  $10A/30VDC$ .

Este relé dispone de tres contactos: *C* (común), *NO* (normalmente abierto) y *NC* (normalmente cerrado). Generalmente, esta es la configuración de los relés mecánicos que se pueden encontrar en el mercado, aunque también encontramos modelos que prescinden del terminal *NC*, como los relés de estado sólido.

Su funcionamiento es el siguiente: al activarse la señal de control, la corriente circula por la bobina del circuito primario generando un campo magnético que hace pivotar una armadura, que a su vez

empuja al contacto móvil, cerrando el circuito con el contacto fijo *NO*, mientras, se separa y abre el circuito con el terminal *NC*. Cuando la corriente del circuito primario cesa, el contacto móvil vuelve a su posición original, abriendo el circuito con el terminal *NO*, y cerrándolo con el terminal *NC* [29]. En la siguiente imagen (Figura V.8) podemos observar el esquema electrónico del módulo de conmutación utilizado.

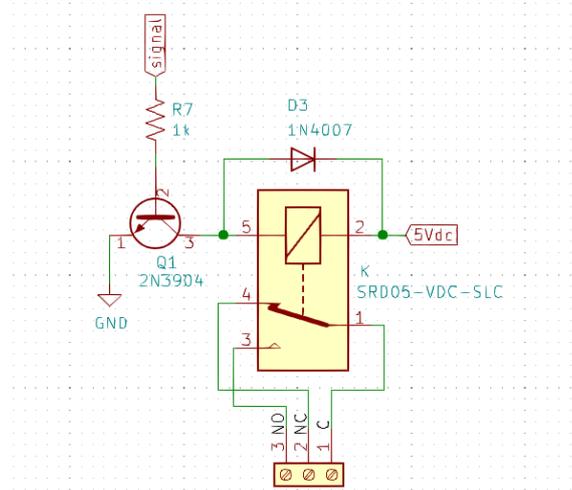


Figura V.18 - Esquema electrónico del módulo de conmutación

En el caso que nos atañe, se utiliza para conmutar el panel FV con el condensador para realizar su carga, o descarga mediante resistencia de potencia. Como es un relé mecánico, el tiempo de conmutación no es despreciable y se debe tener en cuenta. Para calcular este tiempo, le hemos realizado varios pruebas bajo diferentes condiciones y se ha medido los tiempos con un osciloscopio. De esta manera, el tiempo de conmutación ha resultado en torno a los 5ms, y una resistencia entre sus terminales de 100mΩ aproximadamente.

### V.3.5 - Buck-Boost

Un reductor-elevador Buck-Boost es un tipo de convertidor DC-DC que dispone de una salida la cual, su voltaje puede ser mayor o menor al voltaje de entrada. Se utiliza para modificar y estabilizar voltajes a un valor determinado.

La principal ventaja del dispositivo elegido (Figura V.19) es que es capaz de transformar una tensión de entrada entre el rango de 3,6 – 30V, y obtener a la salida +/-Vcc ajustable, que comprende los valores 3,3V, 5V, 6V, 9V, 12V, etc. Esta característica se ha tenido en cuenta a la hora de elegir el dispositivo, pues el amplificador operacional TL082 requería una alimentación de +/-Vcc.



Figura V.19 - Convertidor Buck-Boost

### V.3.6 - Regulador de tensión

Es un dispositivo capaz de ofrecer una tensión constante a su salida a partir de un voltaje de entrada determinado. En este caso, el regulador LM7805 se ha elegido por obtener a su salida una tensión de 5V mediante un voltaje de entrada de entre 0 y 35V [30].

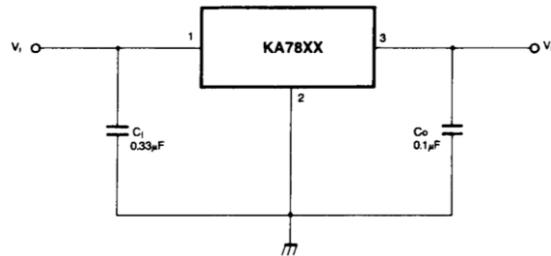


Figura V.20 - Configuración recomendada del LM7805

Este regulador se ha utilizado para alimentar a 5V los dos microcontroladores y sus conversores ADC externos, los módulos de comunicación y el de conmutación.

### V.3.7 - Módulo Bluetooth

Existen módulos compatibles con la tecnología de Arduino, capaces de crear una comunicación y transmisión de datos mediante conexión bluetooth con otro dispositivo compatible. Estos dispositivos suelen ser de bajo precio y fáciles de implementar al microcontrolador.

Bluetooth es un protocolo de comunicación por radiofrecuencia en la banda ISM de 2,4Ghz de corto alcance (de 5 a 10 metros). Generalmente, todos los dispositivos electrónicos modernos ya tienen integrada un módulo de comunicación. Se ha convertido en uno de los mejores métodos inalámbricos de transmisión de datos por su facilidad de uso.

En este proyecto se ha utilizado el módulo HC-05 mostrado en la siguiente imagen (Figura V.9). Éste permite tanto recibir comunicaciones como iniciarlas (master and slave). Por ello, incluso es posible programar el Arduino de forma inalámbrica a través de bluetooth, sin tener que utilizar el cable USB.



Figura V.21 - Módulo bluetooth HC-05

Como se puede ver en la imagen de la derecha, dispone de sólo seis pines para realizar la comunicación, de los cuales solo cuatro son necesarios para realizar una correcta comunicación. Estos son:  $V_{cc}$  (3,6 – 6V),  $GND$  y los pines seriales,  $Rx$  y  $Tx$  [31]. En la parte superior, vemos la antena en forma de serpiente de cobre .

Este tipo de módulos funcionan gracias a los comandos AT, los cuales son usados para obtener la configuración, información y control de estos.

### V.3.8 - Módulo UART-USB

Con la intención de ofrecer al usuario una conectividad vía cable, a parte de la conexión bluetooth, hemos optado por añadir al microcontrolador, un módulo UART con conexión USB. En la siguiente imagen (Figura V.10) podemos visualizar el módulo utilizado, modelo *CP2102*.

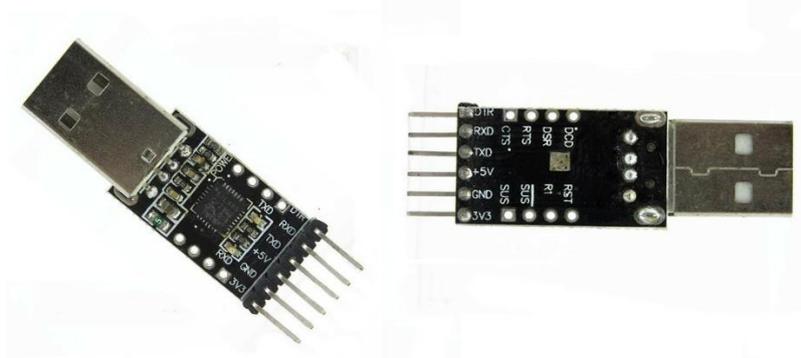


Figura V.22 - Módulo UART-USB Arduino

Para conectar esta placa con el microcontrolador, es necesario tener en cuenta los seis pines con los que cuenta. Estos son: *DTR* (si se requiere un reinicio del microcontrolador), pines seriales de comunicación *Rx* y *Tx*, y pines de salida de alimentación a 5 o 3,3V y *GND*. Por otro lado, el conector USB es de tipo A, con velocidad de transmisión 12Mbps (2.0). El módulo tiene implementado un cristal de cuarzo, un regulador de voltaje a 3,3V y unos buffer de transmisión y recepción de 640 Bytes y 576 Bytes respectivamente [32].

Este convertor facilita la comunicación entre un PC y un microcontrolador utilizando el protocolo USB. Es compatible con Arduino y tiene un bajo coste. Puede funcionar como "programador" del microcontrolador, pues incluye el pin *DTR/RESET* necesario para cargar fácilmente un Sketch. Para ello, solo se necesita usar el clásico protocolo serial UART.

## V.4 - Adquisición de datos

En este apartado se describirán las características del ciclo de la adquisición de los datos referente a la curva característica I-V.

Este proceso comienza cuando el usuario, a través de la interfaz *PV Curve Plotter*, pulsa el botón perteneciente a la orden de obtención de la curva. El ordenador le enviará una señal al microcontrolador maestro a través de conexión serial. Éste, a su vez, enviará un aviso (pin *D2\_M* en alto) al microcontrolador esclavo para que los dos (tanto maestro como esclavo), comiencen a leer datos a la misma vez. El microcontrolador maestro pondrá a baja la señal de *Select* (activación de convertor ADC) para realizar cada conversión y lectura del ADC. Esta señal será enviada a los dos convertidores por igual, pero cada microcontrolador solo recogerá los datos de su convertidor correspondiente. En cambio, el esclavo leerá la señal de *Select* para abrir y almacenar la entrada de los valores que le llegan de su propio convertidor. Se ha realizado de esta manera para asegurar la lectura en el mismo instante de los valores tanto de voltaje como de corriente, con desfase entre tiempos prácticamente nulos.

Se debe tener en cuenta que, antes de la lectura de datos, se debe realizar la conmutación del relé para que conecte la entrada del panel FV con el condensador. Mediante varios experimentos de conmutaciones, hemos conocido que el relé emplea un tiempo de aproximadamente 5ms para la

conmutación. Por lo tanto, previo a realizar la lectura de datos, se enviará esta señal de activación al relé, se espera el tiempo correspondiente y se comienza a leer los datos.

En el software de los microcontroladores se ha especificado que se muestreen 250 puntos de coordenadas I-V en un tiempo definido. Los datos, una vez que se lean, se irán guardando en un vector de longitud definida por el número máximo de puntos a muestrear. Los datos se almacenarán como números enteros de 16 bits.

Los dos microcontroladores, al terminar esta lectura, se comunicarán entre ellos. El esclavo podrá en alto la salida D3-S, que está conectada con la entrada del maestro D3-M. Si el maestro lee que esta entrada está en alto y está preparado para recibir los datos, pondrá en alto la salida D4-M. Esta señal le llegará al esclavo mediante la entrada D4-S, e indica la solicitud de datos al maestro. Por lo tanto, el esclavo comenzará el envío mediante serial de los valores almacenados en su vector de corriente. Estos serán recibidos por el maestro, el cual los almacenará en otro vector de las mismas características. Una vez el maestro tiene almacenados los datos en su memoria, se los envía inmediatamente a la interfaz.

Los datos, en su llegada a la interfaz, conviene ser acondicionados porque llegan a esta tal cual se han leído en los conversores, como una palabra de 12 bits en decimal. Estos valores deben ser transformados a valores de voltaje y corriente correspondientes. Una vez hecho esto, se finaliza pasando un pequeño filtro, borrando de esta manera los valores que no interesan en la gráfica por no estar acordes a los demás. Tras esto, se comprueba si hay suficientes datos para dibujar la curva. Deberán ser más de 120 puntos. Si no se cumple esta condición, se vuelve al principio para volver a realizar la toma de datos, esta vez en la mitad de tiempo que la anterior.

Y así se repetirá unas 10 veces, comenzando la primera toma de datos en aproximadamente en 600ms y acabando la última muestra en unos 0,5ms. Realizamos este barrido de tiempos por el motivo de que el tiempo de carga y descarga del condensador está definida por la constante de tiempo (que es el producto de la resistencia por la capacitancia del mismo). No siempre encontraremos iguales condiciones de resistencia asociada a los cables de los paneles y sus conexiones. Por lo tanto, para asegurar un buen trazado de la curva en cualquier circunstancia, hemos diseñado el trazador para que poco a poco vaya reduciendo el tiempo de lectura si los datos no son válidos, comenzando con un tiempo bastante prudente.

En el momento en que se pase el filtro a los datos y se dé por correcta la lectura, automáticamente se visualizará la curva I-V en la interfaz. Además, se podrá mostrar la curva de potencia, guardar y visualizar la lista de datos obtenidos. Por último, si se requiere realizar una nueva curva por parte del usuario, se podrá activar de nuevo el muestreo mediante la pulsación del botón correspondiente, se borrarán todos los datos y el ciclo comenzará desde el principio.

## **CAPÍTULO VI: DESCRIPCIÓN DEL SOFTWARE**

## VI.1 - Introducción

El software desarrollado para este proyecto consta de dos partes muy diferenciadas. Por un lado, el implementado en los microcontroladores (maestro y esclavo) desarrollado en lenguaje C, y por otro lado el perteneciente a la interfaz gráfica y su tratamiento de datos, elaborado en lenguaje Python. A continuación, pasaremos a describir cada uno de ellos.

## VI.2 - Funcionamiento general

El funcionamiento general es básico y muy simple. Mediante la aplicación *PV Curve Plotter*, el usuario tiene el poder solicitarle los datos de la curva al dispositivo. Esta petición es recibida por el microcontrolador maestro, que informará al microcontrolador esclavo que debe prepararse para el muestreo. Previamente, se deberá realizar una conmutación para conectar la entrada del panel FV con el condensador de carga. Cuando lo esté, comenzará la lectura de cada punto. En el mismo instante de tiempo, el maestro y el esclavo leerán los datos de los conversores ADC. Los datos correspondientes al voltaje los medirá el maestro, y los de corriente, el esclavo. Una vez hayan guardado cada uno sus datos, el esclavo le enviará al maestro su lista de valores de intensidad. En el momento en que el maestro los tenga almacenados en su memoria, se los enviará a la aplicación donde serán dibujados como gráfica I-V.

El esquema hardware del sistema se puede observar en la siguiente imagen (Figura VI.1):

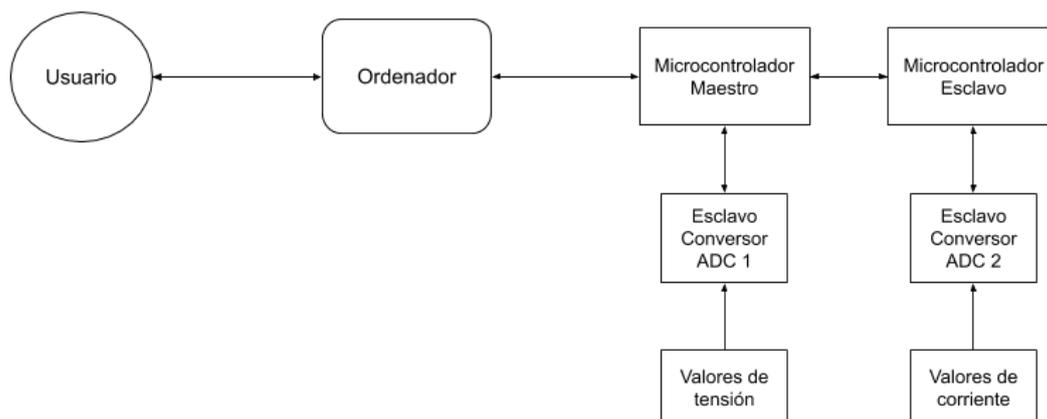


Figura VI.1 - Esquema hardware del sistema

A continuación, describiremos más detalladamente tanto el ciclo de toma de datos de los microcontroladores como la adecuación y posterior visualización de la curva en la aplicación.

### VI.2.1 - Ciclo de trabajo de los microcontroladores

Como explicamos en el Capítulo IV, el desarrollo del software implementado en ambos microcontroladores (Maestro y Esclavo) se ha realizado en el entorno de programación Arduino IDE en lenguaje C. El microcontrolador que tiene el poder de ser maestro se encarga de estar atento a la orden que le llega desde la interfaz para activar al mismo tiempo tanto el muestreo de todos los puntos de su propio ADC como el del esclavo, de ir recogiendo progresivamente los datos correspondientes a su conversor, de los facilitados por el esclavo, y del posterior envío a la interfaz para lograr una correcta representación.

Por otro lado, el microcontrolador esclavo permanecerá a las órdenes del maestro, esperando en todo momento que la señal *Select* pase de alta a baja para leer el valor de su conversor y almacenar los datos, que al finalizar el muestreo de todos los puntos le serán enviados al maestro. El diagrama de flujo correspondiente a esto se puede visualizar en la siguiente imagen (Figura VI.2).

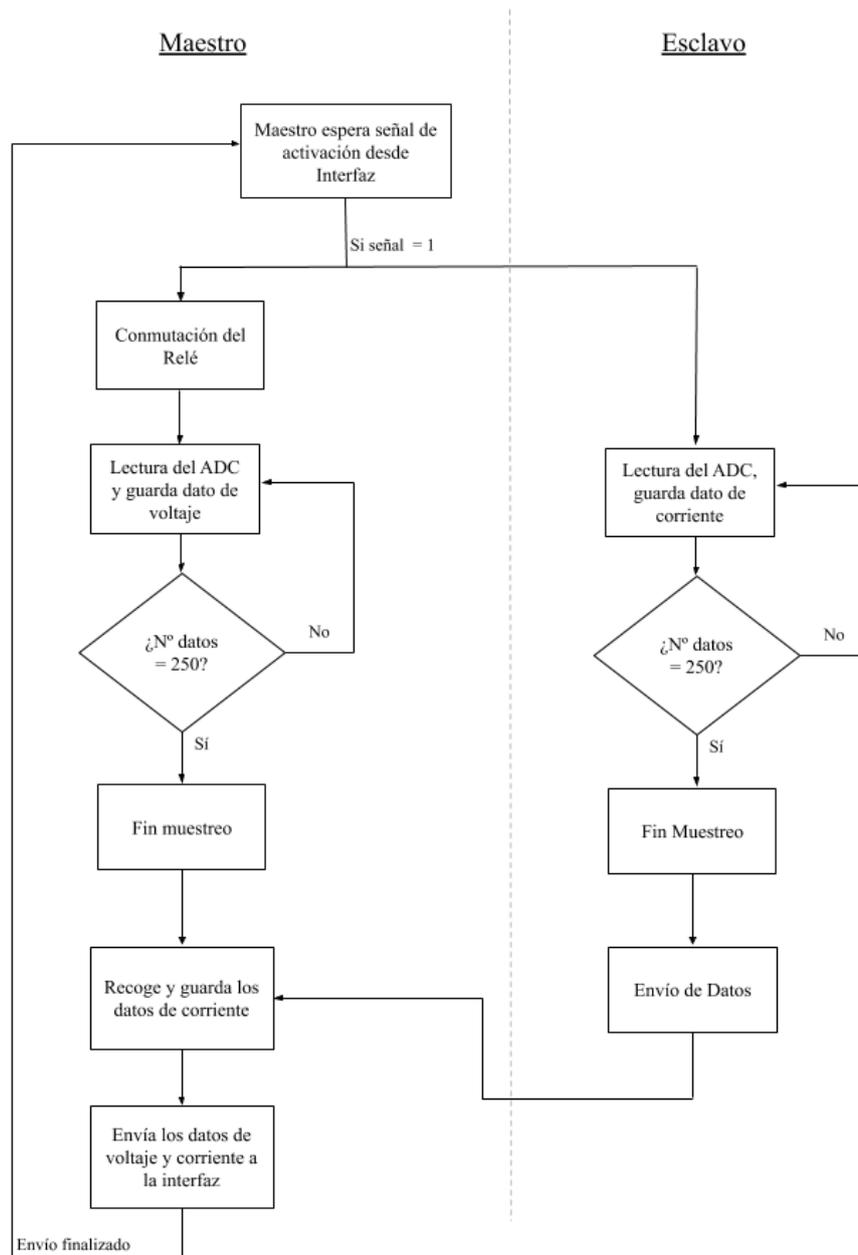


Figura VI.2 - Ciclo general de trabajo de los microcontroladores

Ahora bien, explicaremos con más detalle el ciclo de trabajo de cada microcontrolador.

#### VI.2.1.1 - Software del Maestro

Como bien nombramos en el anterior apartado, el maestro es el responsable de la toma de mediciones. En estado de reposo y después de haber realizado el Setup, deberá estar atento a la petición de datos de la interfaz para comenzar el muestreo de datos (Figura VI.3). Cuando ésta se reciba, inmediatamente comunicaremos al microcontrolador esclavo que debe ponerse en modo muestreo, y se realizará la conmutación del relé. El dispositivo deberá esperar aproximadamente 4,5ms para que el relé realice la conmutación de una manera correcta, y así comenzar el muestreo. La señal de salida *Select*, que en la condiciones iniciales está definido que siempre su estado sea a uno excepto a la hora de activar el conversor ADC, pasará a cero. Esta señal les llegará a los dos conversores al mismo tiempo, por lo que deberán muestrear en el mismo instante de tiempo. Este microcontrolador le enviará una señal de reloj a su conversor para que envíe, mediante el pin

$D_{out}$ , de forma síncrona el valor en digital de la señal convertida. En el momento en que se haya realizado la conversión y almacenado de los datos, el pin *Select* volverá a estar en estado de alta para estar poder recoger el siguiente dato de la misma manera que este . Cabe destacar que los valores muestreados correspondientes a cada conversor se irán guardando en un vector con capacidad de almacenar 250 datos de 16 bit cada uno. De estos 16 bits, solo se utilizarán 12, por el motivo de que es el valor límite de bits de conversión del ADC.

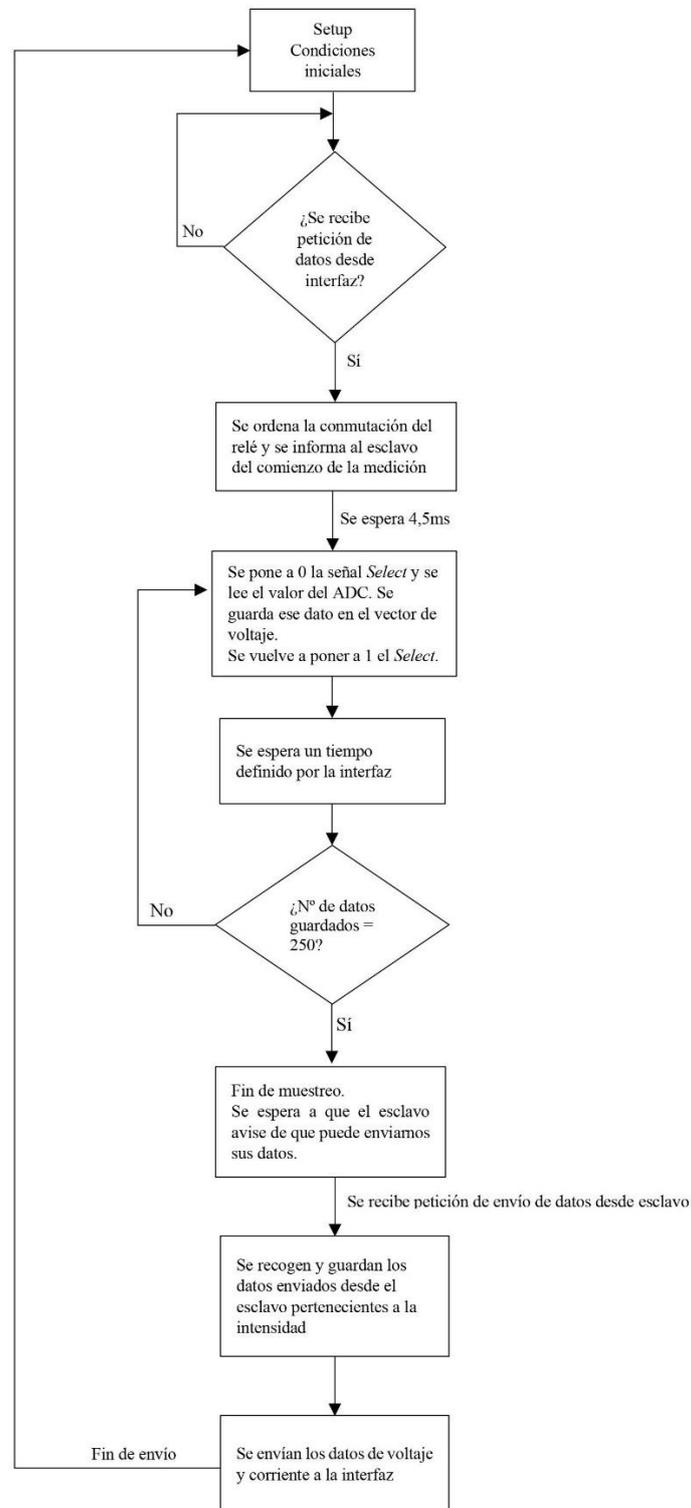


Figura VI.3 - Ciclo de trabajo del maestro

En el momento en que se completan las 250 posiciones de los vectores, el microcontrolador esclavo deberá realizar un envío mediante serial de los datos que ha almacenado. Se ha configurado mediante software una nueva conexión serial Rx-Tx en el maestro para efectuar estos envíos, porque los pines que vienen definidos por defecto se encontraban en uso.

Una vez que el maestro tiene en su memoria los dos vectores correspondientes a las medidas de voltaje y corriente, son enviados a la interfaz.

#### VI.2.1.2 - Software del Esclavo

El ciclo de trabajo del esclavo es semejante al del maestro, pero mucho más simple y con menor número de tareas a realizar. Éste es el encargado de leer las mediciones de corriente, y su ciclo de trabajo está representado en la Figura VI.4.

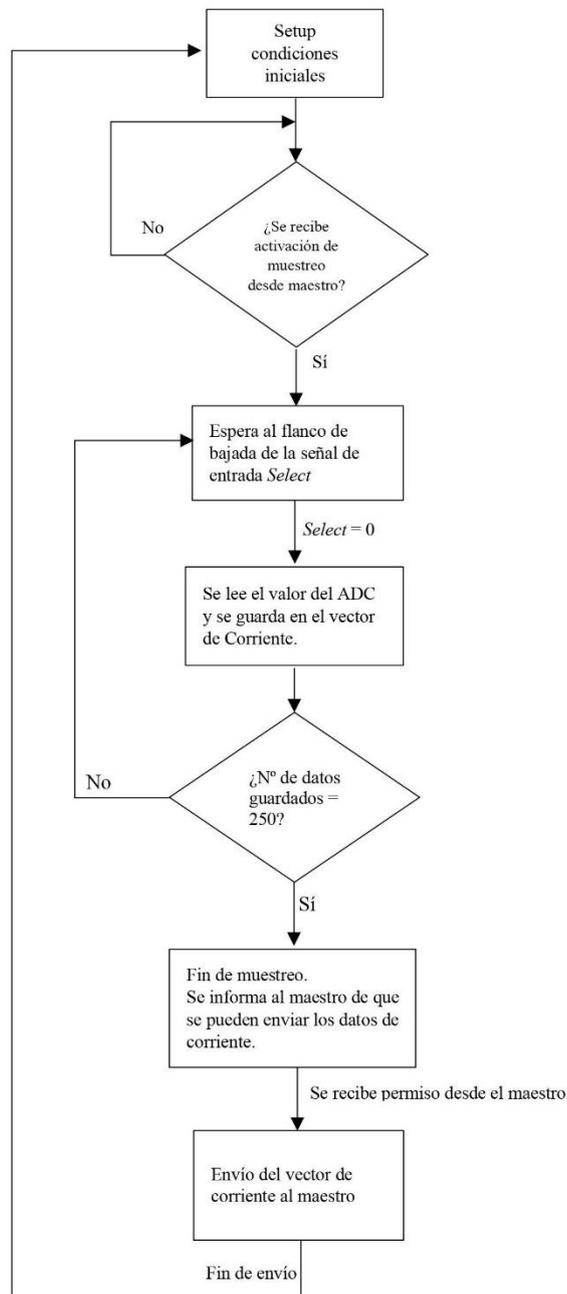


Figura VI.4 - Ciclo de trabajo del esclavo

Tras encontrarse en las condiciones iniciales, estará a la espera de la señal proveniente del maestro que le hará pasar a modo muestreo. En este modo, a su vez, estará atento a cada flanco de bajada de la señal *Select* para efectuar la lectura del valor digital mediante el pin  $D_{out}$  y a la velocidad de reloj (Clk) que éste estipule. Los datos, de igual manera que en el caso del maestro, se irán guardando en un vector de las mismas características.

Al completar el muestreo, es decir, todas las posiciones del vector de corriente deben haber sido completadas por una medición, le será enviado al maestro, y de esta manera, el esclavo volverá a encontrarse en las condiciones iniciales a la espera de una nueva orden de toma de datos.

### VI.2.2 - Software de acondicionamiento de datos

Junto con el código en Python que se ha desarrollado para la aplicación en sí, se ha realizado un breve código capaz de establecer comunicación microcontrolador-PC mediante conexión serial, recoger los datos, filtrarlos y acondicionarlos, y enviarlos a la aplicación (Figura VI.5).

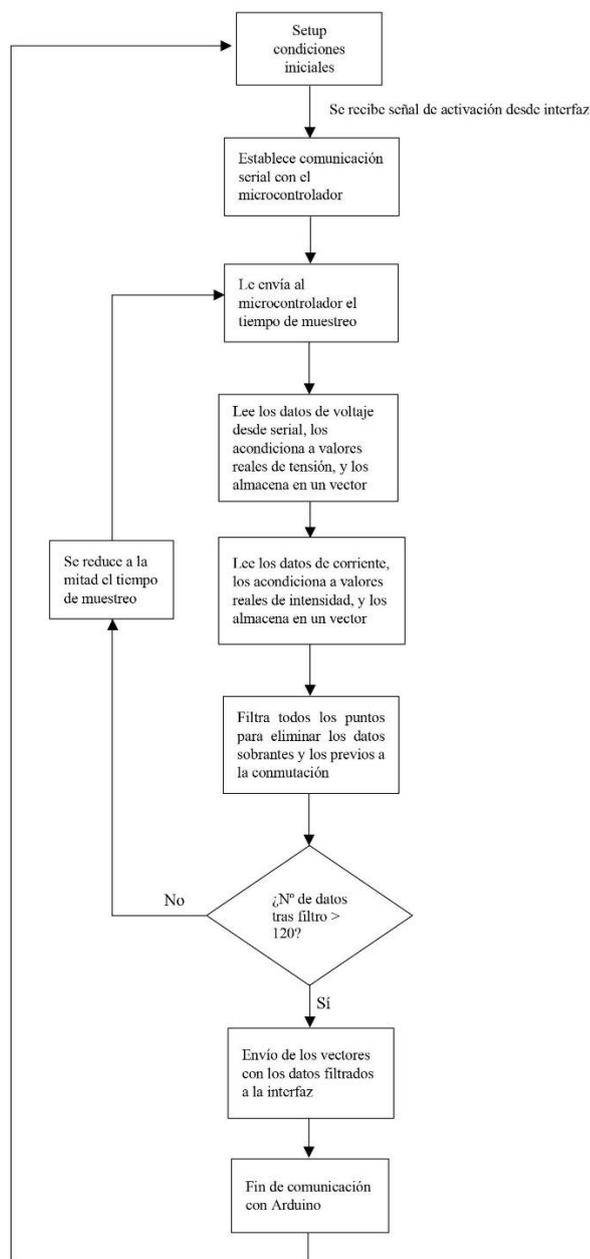


Figura VI.5 - Diagrama de lectura vía serial y acondicionamiento de datos

Por lo tanto, es el paso intermedio de los datos entre el dispositivo de medida y la interfaz. El programa inicialmente establece unas condiciones previas, es decir, inicializa las variables, constantes y vectores, y entra en modo espera, hasta que la aplicación solicite los datos correspondientes a la curva. Cuando llegue este momento, establecerá comunicación con el microcontrolador maestro, y le ordenará la medición de los mismos con un tiempo definido. Este tiempo es de vital importancia porque, como explicamos en apartados anteriores, la constante de tiempo del condensador depende de la resistencia asociada a las conexiones de los paneles fotovoltaicos. Esta resistencia no siempre es igual, puesto que depende de longitud y sección del cable, conectores, uniones, etc. Por lo tanto, si se realizan las conexiones con una resistencia asociada demasiado grande, el tiempo de carga del condensador también aumentará. Esto propicia que se deba ir ajustando el tiempo de muestreo poco a poco hasta encontrar el tiempo exacto donde más del 50% de los puntos medidos por el trazador sean correctos para graficar. Por ejemplo, si el trazador se excede de tiempo para realizar las mediciones, se obtendrán escasos puntos correspondientes a la curva. En cambio, si la lectura se realiza muy rápido, no se le da tiempo al condensador para completar la carga y, por lo tanto, la curva quedaría a medio hacer. Por estos motivos, se debe tardar el tiempo justo en realizar la medida.

Se ha estipulado que se realicen diez mediciones de 250 puntos cada una, comenzando con una toma de datos de 600ms y terminando aproximadamente en 0,5ms. A medida que se van realizando las mediciones, se divide este tiempo.

Tras enviarle el tiempo de medición al microcontrolador, el código entra en modo espera para que el dispositivo realice el proceso de toma de datos. Tras terminarlo, éste los imprimirá en serial. Primero, los valores de voltaje y seguidos de estos, los de corriente. El archivo de Python es capaz de recoger estas lecturas y almacenarlas en un vector para cada una. Los datos, al llegar en valores decimales de 12 bits, es decir, valores entre 0 y 4095, hay que acondicionarlos para determinar el valor real tanto de voltaje como de corriente que se ha medido.

Tras esto, se pasará un filtro a los datos para eliminar las mediciones previas a la conmutación del relé, y los datos sobrantes tras la carga completa del condensador. Tras pasar este filtro deben quedar al menos 120 puntos válidos o más. Si no, se volverá a enviar automáticamente al microcontrolador la orden de volver a realizar la medida, pero esta vez, en la mitad de tiempo de la anterior.

Una vez se hayan obtenido los puntos necesarios (más de 120), se realizará un envío de los mismos a la interfaz para su posterior representación. Tras ello, se cerrará comunicación con el Arduino puesto que el proceso habrá terminado.

## **CAPÍTULO VII: INTERFAZ GRÁFICA**

## VII.1 - Introducción

Con el fin de visualizar las curvas I-V y P-V con los datos adquiridos mediante el dispositivo de medida, se ha diseñado, una interfaz de usuario sencilla y fácil de usar, para este fin. Ésta, tiene la capacidad de posicionar en la gráfica y, unir con líneas, las datos de corriente y potencia en función del voltaje, que conforman las curvas fotovoltaicas. Contiene varias ventanas en las que se pueden visualizar, por un lado, la gráfica y, por otro, la tabla de datos. La aplicación dispone de varios botones que permiten al usuario interactuar con ella.

## VII.2 - Funcionamiento de la aplicación

*PV Curve Plotter* es el nombre dado a esta interfaz. Contiene tres páginas o ventanas: la primera corresponde con la carátula de inicio (Figura VII.1), una segunda en la que se visualiza la gráfica en cuestión y la última en la que se muestra la tabla con los datos obtenidos. Como se puede comprobar en la siguiente figura, sus textos y etiquetas están disponibles únicamente en idioma inglés. Dado que son textos cortos y sencillos, se entienden fácilmente.

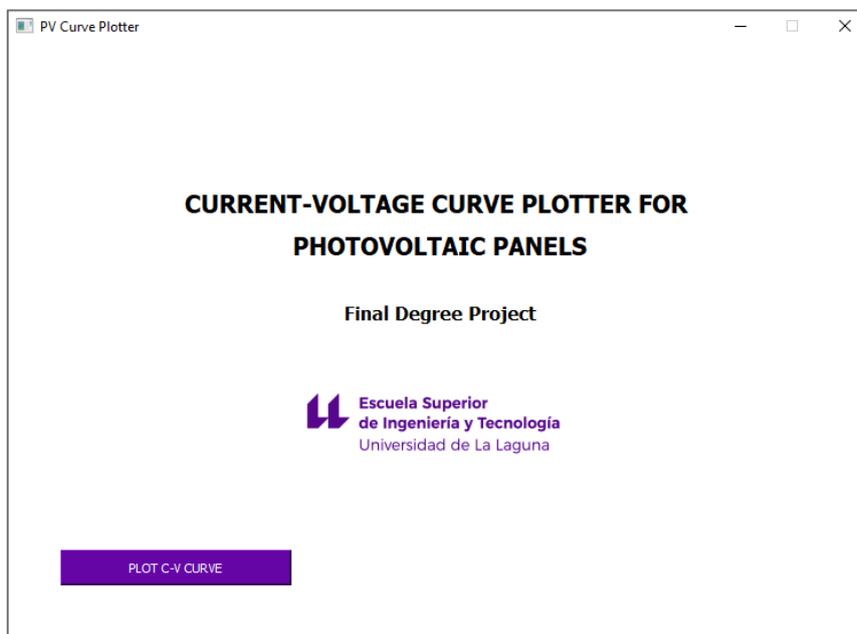


Figura VII.1 - *PV Curve Plotter*: página de inicio

Al presionar el botón *PLOT C-V CURVE* de la esquina inferior izquierda de la página de inicio, se activa la petición de datos, es decir, se envía una señal de activación del muestreo al dispositivo trazador y se vuelcan los datos en la gráfica. En el tiempo que transcurre esta acción (aproximadamente 5 segundos), se muestra en la esquina inferior derecha de la pantalla, un mensaje indicativo (Figura VII.2) referente a esta acción (*Sampling...*, Muestreando).

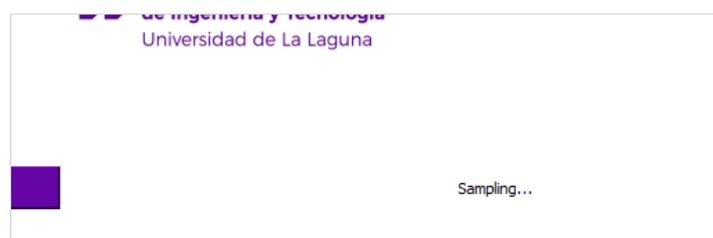


Figura VII.2 - *PV Curve Plotter*: mensaje informativo de muestreo

Si por algún motivo no se puede establecer conexión (USB o Bluetooth) con el dispositivo de medida, o ésta se interrumpe durante el proceso de lectura de los datos, la etiqueta nombrada anteriormente cambiará su texto y mostrará un mensaje relativo a este error (Figura VII.3).

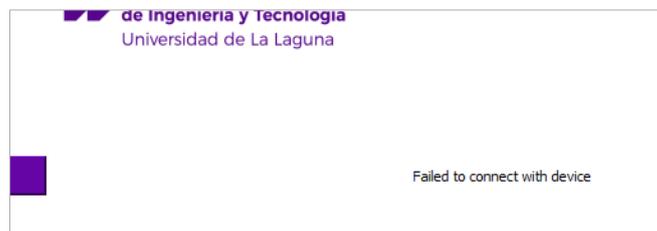


Figura VII.3 - PV Curve Plotter: mensaje informativo de error de conexión

Tras realizarse el muestreo, la interfaz recibirá los datos desde el dispositivo y se trazarán las curvas en la gráfica situada en la segunda página de la aplicación (Figura VII.4).

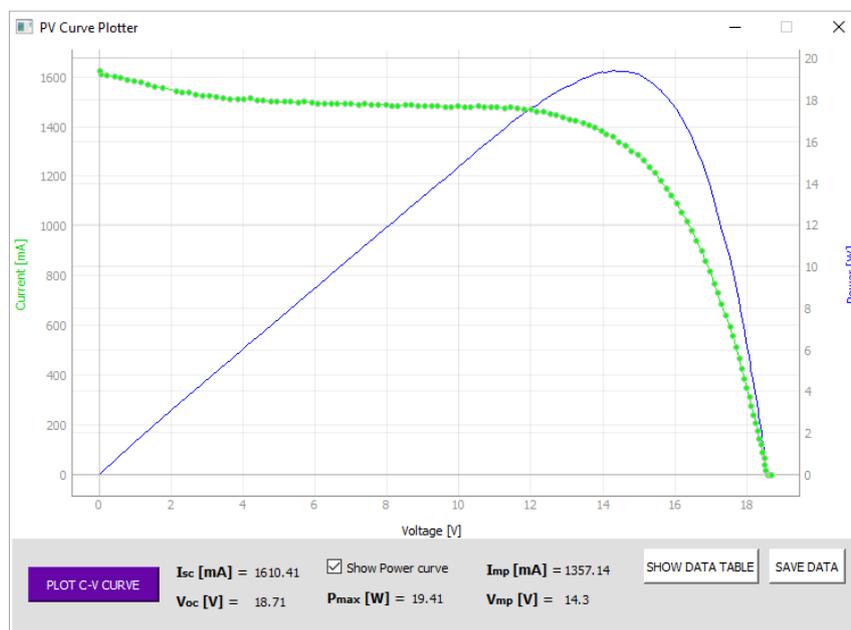


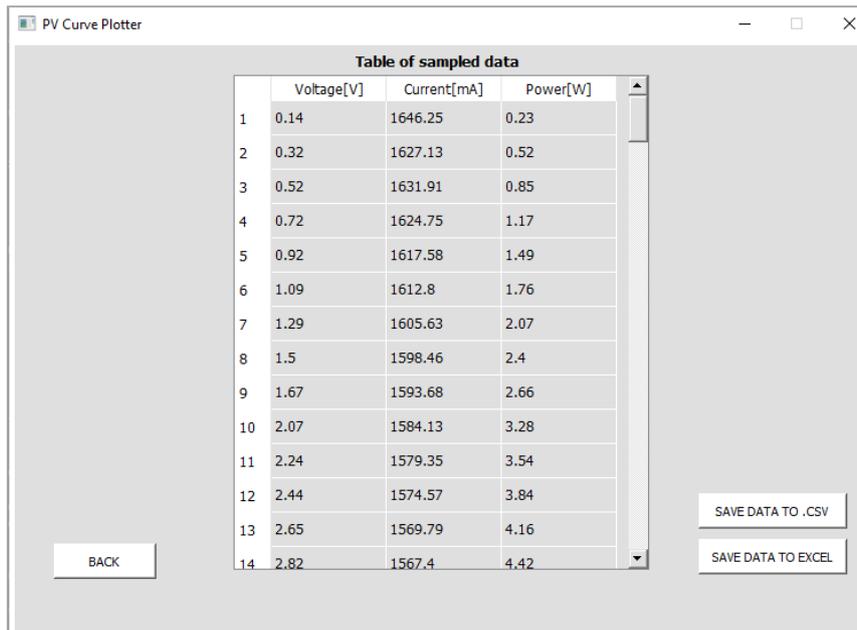
Figura VII.4 - PV Curve Plotter: ventana gráfica de las curvas

En esta ventana se visualizan dos curvas: la de color verde se corresponde a la corriente (en mA, miliamperios) y la de azul a la de potencia (en W, vatios), las dos en función del voltaje.

Además, en la esquina inferior izquierda se dispone de un botón (*PLOT C-V CURVE*) con el que se puede volver a muestrear para obtener una nueva gráfica, borrándose los datos anteriores de la curva. A la derecha de este botón, se encuentran las etiquetas donde se muestran los valores de  $I_{sc}$ ,  $V_{oc}$ ,  $P_{max}$ ,  $I_{mp}$  y  $V_{mp}$  (explicados en el Capítulo II) correspondientes a cada curva.

La curva de potencia se puede mostrar u ocultar mediante la selección de una casilla de verificación (*Show Power Curve*) situada en la zona central inferior de la página. Los botones, que se han diseñado para disponer de un mejor control de la aplicación, los encontramos en la zona inferior derecha. Uno de ellos tiene la función de pasar a la página siguiente y mostrar la tabla de datos (*SHOW DATA TABLE*), mientras que el otro realiza el almacenamiento de los mismos en un archivo Excel (*SAVE DATA*).

En la última página se encuentra una lista (Figura VII.5) correspondiente a los datos de las dos curvas, con todos los puntos de voltaje, corriente y potencia.



The screenshot shows a window titled "PV Curve Plotter" with a standard Windows title bar (minimize, maximize, close). The main content area is titled "Table of sampled data" and contains a table with 14 rows and 4 columns. The columns are labeled "Voltage[V]", "Current[mA]", and "Power[W]". The first column contains row numbers from 1 to 14. The table data is as follows:

	Voltage[V]	Current[mA]	Power[W]
1	0.14	1646.25	0.23
2	0.32	1627.13	0.52
3	0.52	1631.91	0.85
4	0.72	1624.75	1.17
5	0.92	1617.58	1.49
6	1.09	1612.8	1.76
7	1.29	1605.63	2.07
8	1.5	1598.46	2.4
9	1.67	1593.68	2.66
10	2.07	1584.13	3.28
11	2.24	1579.35	3.54
12	2.44	1574.57	3.84
13	2.65	1569.79	4.16
14	2.82	1567.4	4.42

Below the table, there are three buttons: "BACK" on the left, and "SAVE DATA TO .CSV" and "SAVE DATA TO EXCEL" on the right. A vertical scrollbar is visible on the right side of the table.

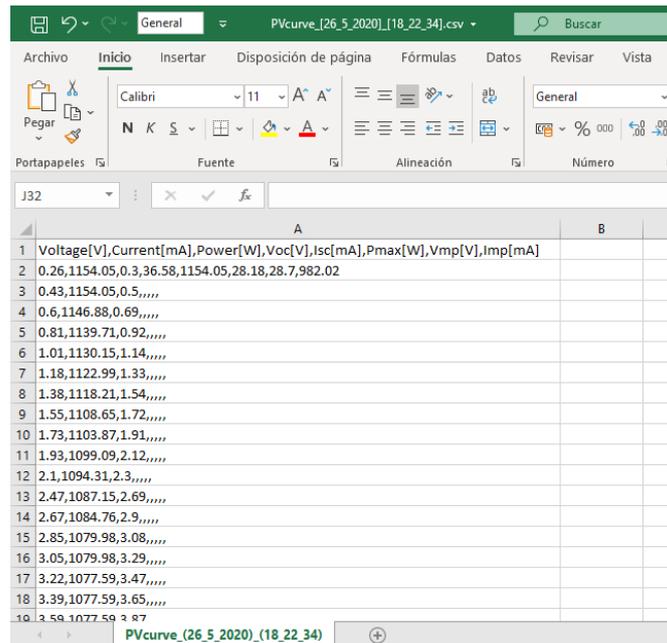
*Figura VII.5 - PV Curve Plotter: ventana de tabla de datos*

Con la barra de desplazamiento (o Scroll) nos podemos desplazar hacia arriba y hacia abajo para visualizar todos los datos. En la esquina inferior izquierda, se localiza un botón (*Back*) que nos devuelve a la página anterior.

### **VII.3 - Generación de archivos de datos**

Como se ha comentado en el apartado anterior, la aplicación tiene la capacidad de almacenar la lista de los datos de los puntos obtenidos en un tipo de documento propio para ello. De esta manera, y por medio de los botones de la zona inferior derecha de la Figura VII.5, disponemos de dos opciones para guardar la lista de datos: una en un archivo con formato \*.csv de software libre (Figura VII.6), mediante el botón SAVE DATA TO .CSV de la Figura VII.5 localizado en la parte inferior derecha de esta. El otro formato del que se dispone es \*.xlsx, propio de Microsoft Excel (Figura VII.7), que se almacena mediante el botón SAVE DATA TO .EXCEL de la Figura VII.5. A través del formato \*.csv, se ha optado por generar un documento de software libre y abierto, no comercial como es el caso de Excel.

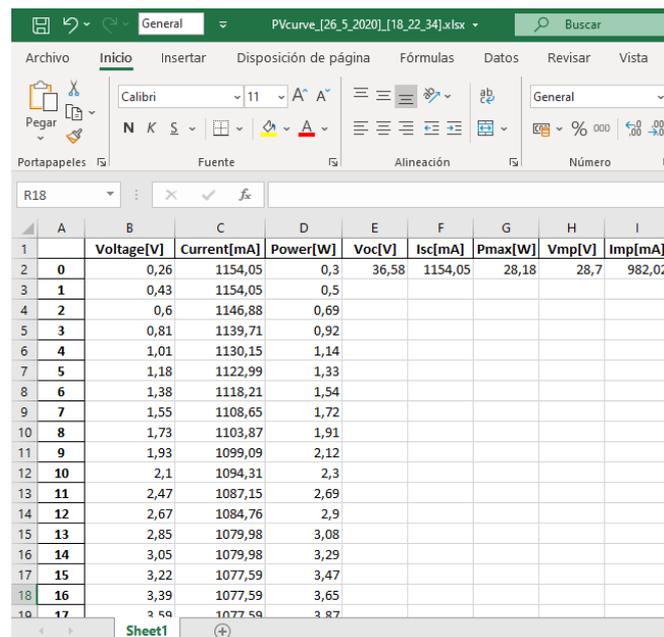
En los dos documentos se guardará toda la lista de puntos de Voltaje, Corriente y Potencia, y por otro lado también los valores de  $V_{oc}$ ,  $I_{sc}$ ,  $P_{max}$ ,  $I_{mp}$  y  $V_{mp}$  correspondientes a cada medida.



The screenshot shows a spreadsheet application with a CSV file named 'PVcurve\_[26\_5\_2020]\_[18\_22\_34].csv'. The data is organized in a single row across columns A and B. Column A contains a list of parameters: Voltage[V], Current[mA], Power[W], Voc[V], Isc[mA], Pmax[W], Vmp[V], and Imp[mA]. Column B contains the corresponding numerical values for each parameter.

	A	B
1	Voltage[V],Current[mA],Power[W],Voc[V],Isc[mA],Pmax[W],Vmp[V],Imp[mA]	
2	0,26,1154,05,0,3,36,58,1154,05,28,18,28,7,982,02	
3	0,43,1154,05,0,5,,,,,	
4	0,6,1146,88,0,69,,,,,	
5	0,81,1139,71,0,92,,,,,	
6	1,01,1130,15,1,14,,,,,	
7	1,18,1122,99,1,33,,,,,	
8	1,38,1118,21,1,54,,,,,	
9	1,55,1108,65,1,72,,,,,	
10	1,73,1103,87,1,91,,,,,	
11	1,93,1099,09,2,12,,,,,	
12	2,1,1094,31,2,3,,,,,	
13	2,47,1087,15,2,69,,,,,	
14	2,67,1084,76,2,9,,,,,	
15	2,85,1079,98,3,08,,,,,	
16	3,05,1079,98,3,29,,,,,	
17	3,22,1077,59,3,47,,,,,	
18	3,39,1077,59,3,65,,,,,	
19	3,59,1077,59,3,87,,,,,	

Figura VII.6 - PV Curve Plotter: Hojas de datos .csv



The screenshot shows an Excel spreadsheet with a file named 'PVcurve\_[26\_5\_2020]\_[18\_22\_34].xlsx'. The data is organized in a table with headers in row 1 and data in rows 2 through 19. The headers are: Voltage[V], Current[mA], Power[W], Voc[V], Isc[mA], Pmax[W], Vmp[V], and Imp[mA].

	A	B	C	D	E	F	G	H	I
		Voltage[V]	Current[mA]	Power[W]	Voc[V]	Isc[mA]	Pmax[W]	Vmp[V]	Imp[mA]
2	0	0,26	1154,05	0,3	36,58	1154,05	28,18	28,7	982,02
3	1	0,43	1154,05	0,5					
4	2	0,6	1146,88	0,69					
5	3	0,81	1139,71	0,92					
6	4	1,01	1130,15	1,14					
7	5	1,18	1122,99	1,33					
8	6	1,38	1118,21	1,54					
9	7	1,55	1108,65	1,72					
10	8	1,73	1103,87	1,91					
11	9	1,93	1099,09	2,12					
12	10	2,1	1094,31	2,3					
13	11	2,47	1087,15	2,69					
14	12	2,67	1084,76	2,9					
15	13	2,85	1079,98	3,08					
16	14	3,05	1079,98	3,29					
17	15	3,22	1077,59	3,47					
18	16	3,39	1077,59	3,65					
19	17	3,59	1077,59	3,87					

Figura VII.7 - PV Curve Plotter: Hojas de datos .excel

Debemos señalar que cuando se realiza la acción para almacenar los archivos, éstos se guardan automáticamente en el mismo directorio donde se encuentra la interfaz. El archivo generado tendrá un nombre compuesto de *PVcurve* seguido de la fecha y hora en la que se realizó el muestreo. Por ejemplo, si se guarda el archivo el día 26 de mayo de 2020 a las 18:22:34, el documento tendrá el nombre de “*PVcurve\_(26\_5\_2020)\_(18\_22\_34)*”.

De esta manera, para verificar que el archivo se ha guardado con éxito, se mostrará un mensaje debajo de los botones de almacenamiento, donde se informa de ello (Figura VII.8).

55.46	2.75
43.51	3.32
38.73	3.57
36.34	3.87

SAVE DATA TO .CSV

SAVE DATA TO EXCEL

Saved .csv file

*Figura VII.6 - PV Curve Plotter: mensaje informativo de archivo guardado (.csv)*

Por último, debemos señalar que no se ha contemplado el hecho de volver a mostrar una curva en la interfaz cuyos datos hayan sido muestreados con anterioridad, ya que esta aplicación está diseñada para ordenar la toma de datos y visualizar las curvas I-V y P-V. Se ha considerado que, si se quiere volver a visualizar las curvas correspondientes de unos datos muestreados y almacenados, el programa de cálculo ya nombrado (Microsoft Excel) tiene las herramientas necesarias para generar dichas gráficas con los datos almacenados. No obstante, se podrá incluir en versiones posteriores y más avanzadas de esta interfaz.

## **CAPÍTULO VIII: RESULTADOS EXPERIMENTALES**

### VIII.1 - Introducción

Para verificar el buen funcionamiento del dispositivo de medida objeto de este proyecto, se han realizado diferentes pruebas o experimentos en distintas condiciones de temperatura e irradiancia, utilizando para ello diferentes paneles fotovoltaicos. A partir de éstas, obtendremos una curva I-V y P-V con unos resultados que analizaremos en cada caso.

La Figura VIII.1 muestra el dispositivo trazador en el momento de realizar una toma de datos, con las diferentes conexiones al panel FV y al PC mediante USB.



Figura VIII.1 - Imagen de conexión del dispositivo para la toma de datos

Se realizarán tres pruebas en distintas condiciones ambientales para corroborar su buen funcionamiento, con un posterior análisis de datos: La primera corresponde con la de mayor irradiancia, la segunda se realizará bajo unas condiciones irradiancia media, y la tercera corresponderá a la más baja de las tres. Cabe destacar que, en las pruebas realizadas, la temperatura ambiente apenas sufre variación. Los valores de irradiancia y temperatura ambiente se obtendrán mediante la aplicación *HD Meteo La Palma* [33] en el momento de realizar las mediciones, observando los datos ofrecidos por las estaciones meteorológicas cercanas. Tomamos los valores de temperatura ambiente como referencia, aunque el valor correcto a considerar (según las normas STC) debe ser la temperatura superficial de las células del panel.

Por otro lado, analizaremos y compararemos los valores obtenidos en cada prueba con los medidos directamente con un multímetro. Estos son  $V_{oc}$  (voltaje en circuito abierto) e  $I_{sc}$  (intensidad de cortocircuito). Los demás valores ( $P_{max}$ ,  $I_{mp}$  y  $V_{mp}$ ) no se compararán con un instrumento comercial de medida, dado que no se ha podido conseguir uno para estas pruebas.

### VIII.2 - Paneles fotovoltaicos empleados

Para la realización de los experimentos, se han utilizado dos paneles fotovoltaicos de la marca Philips RTC BPX 47C (Figura VIII.2) de tipo policristalino fabricadas en los años 90.



Figura VIII.2 - Paneles FV utilizados en la toma de datos

Estos paneles FV son de 12V y con una potencia pico de 33W, aunque con el uso que han llevado en el transcurso de estos años, ha perdido bastante eficiencia. Cada uno de ellos tiene un ancho de 0,4m y un largo de 1m. Se les ha diseñado un soporte para elevar su inclinación 25° desde el suelo (ángulo recomendado para instalaciones FV en esta ubicación geográfica), realizando todas las pruebas con los paneles en dirección hacia el sur. Como se ha podido disponer de dos de ellos, en cada prueba se realizarán tres medidas, es decir, una con cada panel por separado y, otra mediante una conexión en serie de ambos. Estas configuraciones se pueden visualizar en la Figura VIII.3 mediante los esquemas realizados.

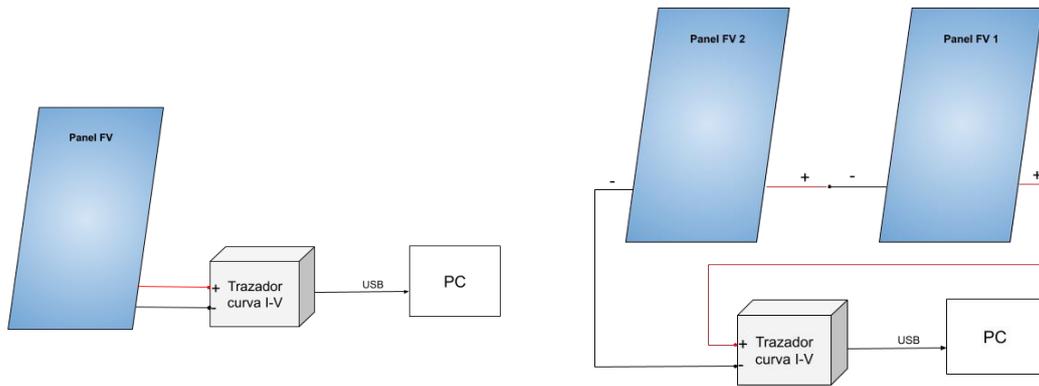


Figura VIII.3 - Esquema de conexiones de los paneles para la toma de datos experimentales

En los siguientes apartados, correspondientes a las pruebas, veremos que en las medidas realizadas con los paneles en serie se obtiene aproximadamente la misma corriente y el doble de voltaje que en las pruebas individuales.

### VIII.3 - Resultados de la Prueba 1

Con una temperatura de 28°C, una irradiancia solar de 800W/m<sup>2</sup>, cielo despejado y a la hora del mediodía en la que la posición del Sol es la más alta, se han obtenido los siguientes resultados:

#### VIII.3.1 - Panel 1

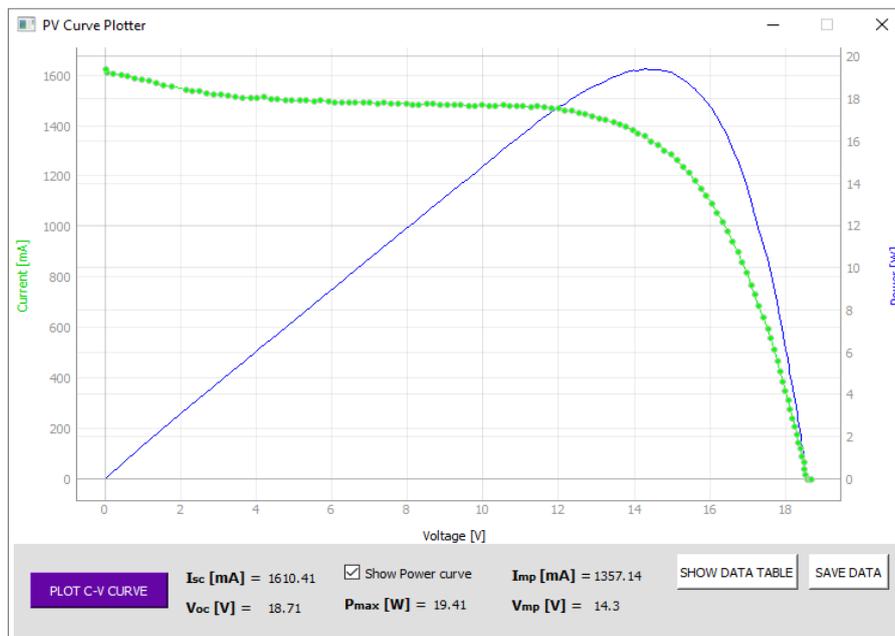


Figura VIII.4 - Curva I-V experimental: panel 1, prueba 1

	Trazador	Multímetro	Diferencia
$V_{oc}$ [V]	18,71	18,65	0,06 (0,32%)
$I_{sc}$ [A]	1,61	1,66	0,05 (-3,01%)
$P_{max}$ [W]	19,41	-	-
$V_{mp}$ [V]	14,30	-	-
$I_{mp}$ [A]	1,36	-	-

Tabla VIII.1 - Tabla de datos experimentales: panel 1, prueba 1

### VIII.3.2 - Panel 2

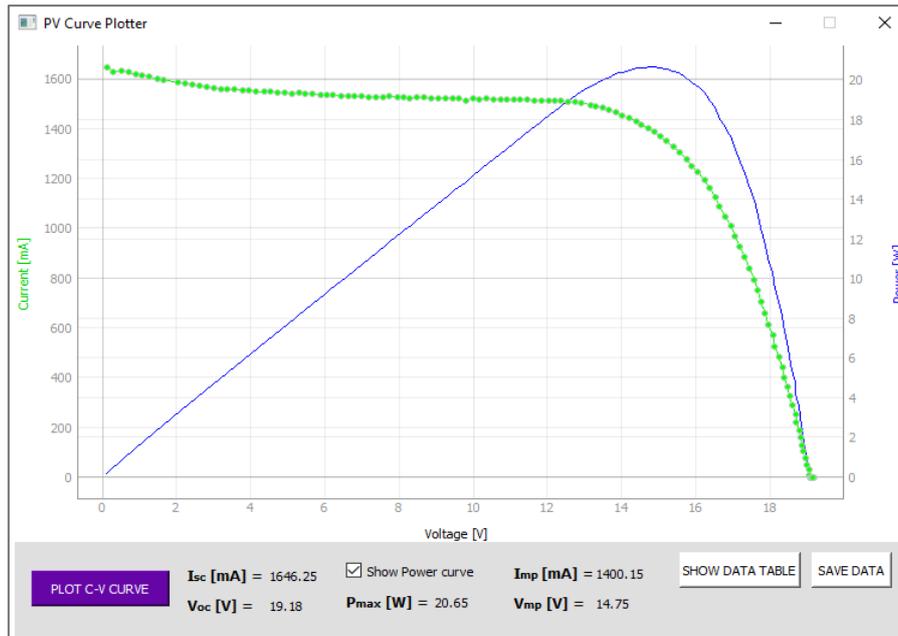


Figura VIII.5 - Curva I-V experimental: panel 2, prueba 1

	Trazador	Multímetro	Diferencia
$V_{oc}$ [V]	19,18	18,60	0,58 (3,12%)
$I_{sc}$ [A]	1,65	1,68	0,03 (-1,79%)
$P_{max}$ [W]	20,65	-	-
$V_{mp}$ [V]	14,75	-	-
$I_{mp}$ [A]	1,40	-	-

Tabla VIII.2 - Tabla de datos experimentales: panel 2, prueba 1

### VIII.3.3 - Serie panel 1 y 2

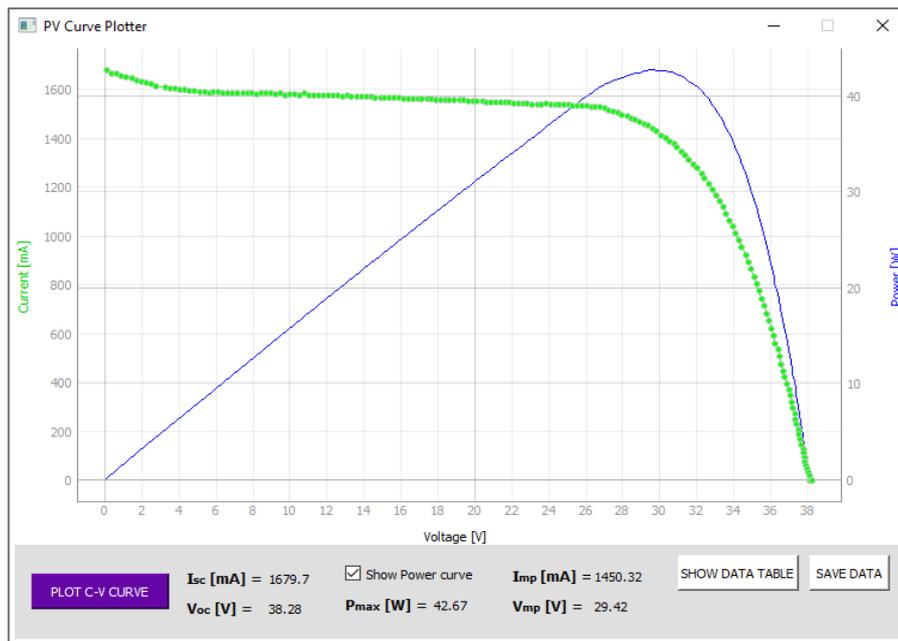


Figura VIII.6 - Curva I-V experimental: paneles en serie, prueba 1

	Trazador	Multímetro	Diferencia
V <sub>oc</sub> [V]	38,28	37,50	0,78 (2,08%)
I <sub>sc</sub> [A]	1,67	1,69	0,02 (-1,18%)
P <sub>max</sub> [W]	42,67	-	-
V <sub>mp</sub> [V]	29,42	-	-
I <sub>mp</sub> [A]	1,45	-	-

Tabla VIII.3 - Tabla de datos experimentales: paneles en serie, prueba 1

### VIII.4 - Resultados de la Prueba 2

Esta segunda prueba se ha realizado bajo una temperatura de 27°C y una irradiancia de 650W/m<sup>2</sup>, con cielo despejado y hacia la mitad de la tarde del día. Se ha obtenido estos resultados:

### VIII.4.1 - Panel 1

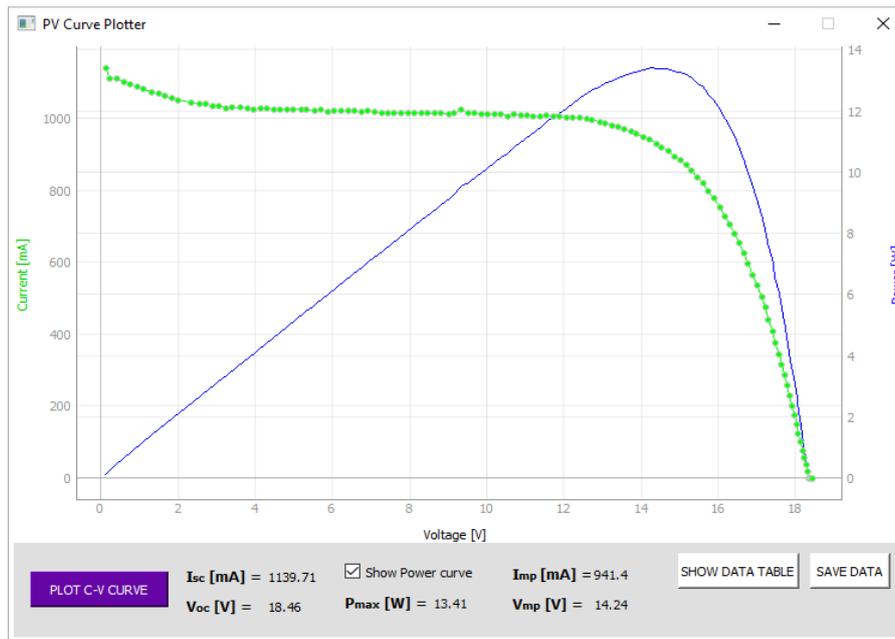


Figura VIII.7 - Curva I-V experimental: panel 1, prueba 2

	Trazador	Multímetro	Diferencia
$V_{oc} [V]$	18,46	18,55	0,09 (-0,49%)
$I_{sc} [A]$	1,14	1,22	0,08 (-6,56%)
$P_{max} [W]$	13,41	-	-
$V_{mp} [V]$	14,24	-	-
$I_{mp} [A]$	0,94	-	-

Tabla VIII.4 - Tabla de datos experimentales: panel 1, prueba 2

### VIII.4.2 - Panel 2

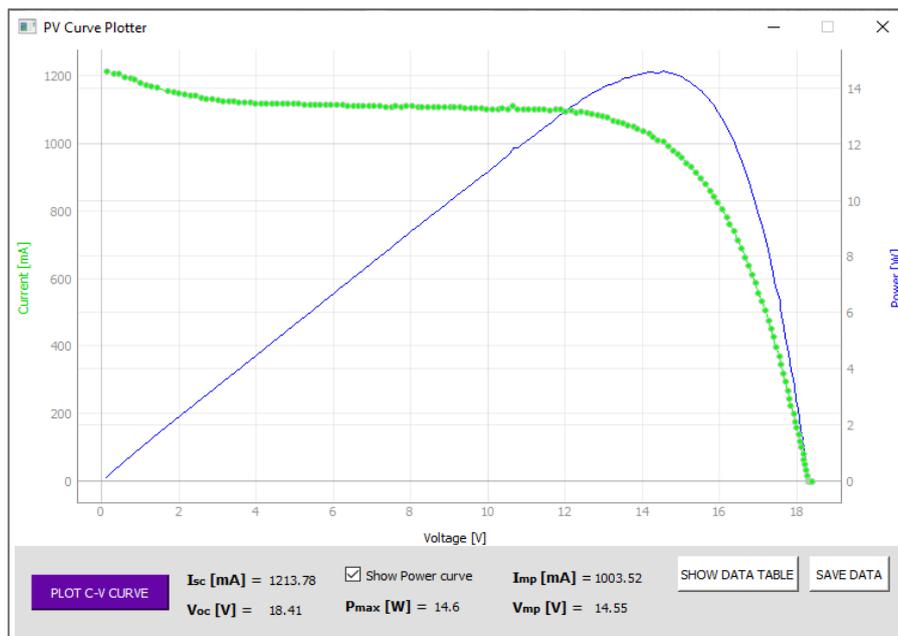


Figura VIII.8 - Curva I-V experimental: panel 2, prueba 2

	Trazador	Multímetro	Diferencia
$V_{oc}$ [V]	18,41	18,45	0,04 (-0,22%)
$I_{sc}$ [A]	1,21	1,26	0,05 (-3,97%)
$P_{max}$ [W]	14,60	-	-
$V_{mp}$ [V]	14,55	-	-
$I_{mp}$ [A]	1,00	-	-

Tabla VIII.5 - Tabla de datos experimentales: panel 2, prueba 2

### VIII.4.3 - Serie panel 1 y 2

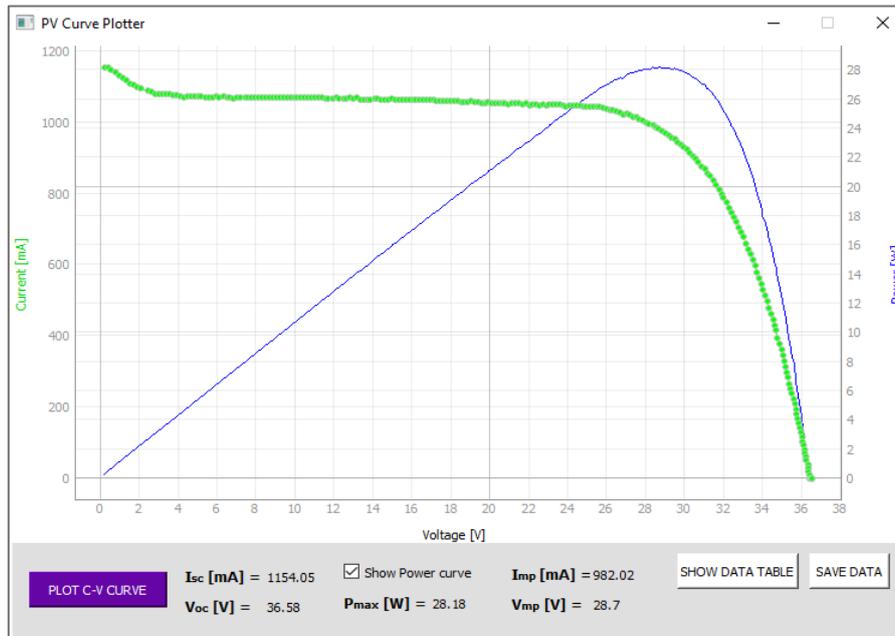


Figura VIII.9 - Curva I-V experimental: paneles en serie, prueba 2

	Trazador	Multímetro	Diferencia
$V_{oc}$ [V]	38,58	36,60	0,02 (-0.05%)
$I_{sc}$ [A]	1,15	1,24	0,09 (-7,26%)
$P_{max}$ [W]	28,18	-	-
$V_{mp}$ [V]	28,70	-	-
$I_{mp}$ [A]	0,98	-	-

Tabla VIII.6 - Tabla de datos experimentales: paneles en serie, prueba 2

### VIII.5 - Resultados de la Prueba 3

Por último, esta tercera prueba se ha realizado bajo unas condiciones ambientales de 26°C y 550W/m<sup>2</sup> de irradiancia en la zona de pruebas, con cielo despejado y hacia la caída del sol, obteniéndose los siguientes resultados:

### VIII.5.1 - Panel 1

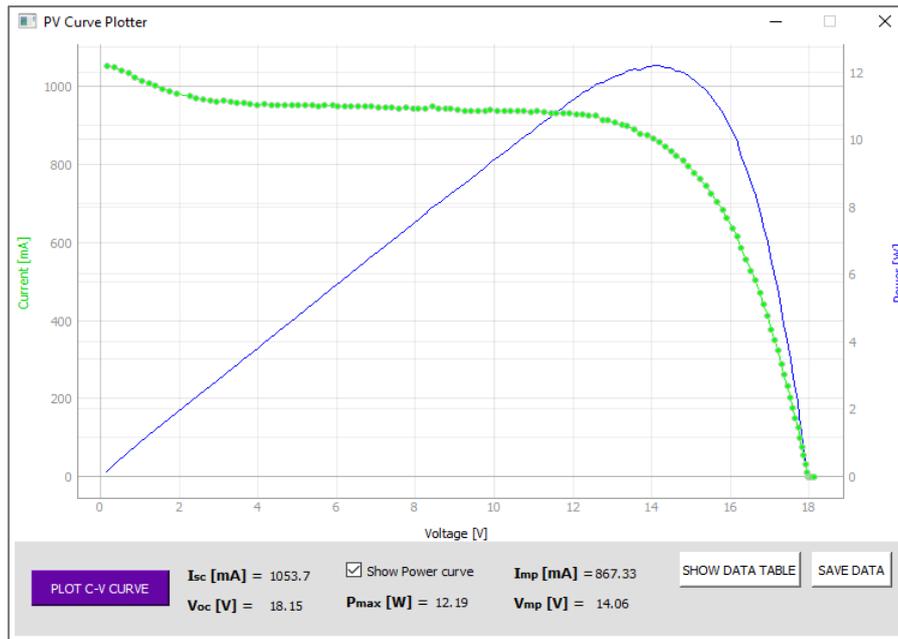


Figura VIII.10 - Curva I-V experimental: panel 1, prueba 3

	Trazador	Multímetro	Diferencia
$V_{oc}$ [V]	18,15	18,27	0,12 (-0,66%)
$I_{sc}$ [A]	1,05	1,14	0,09 (-7,89%)
$P_{max}$ [W]	12,19	-	-
$V_{mp}$ [V]	14,06	-	-
$I_{mp}$ [A]	0,87	-	-

Tabla VIII.7 - Tabla de datos experimentales: panel 1, prueba 3

### VIII.5.2 - Panel 2

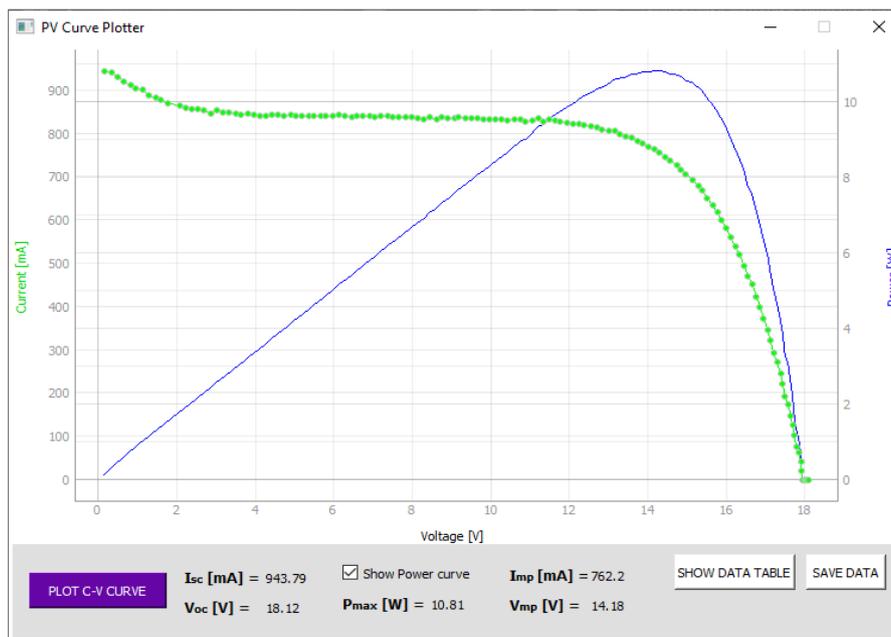


Figura VIII.11 - Curva I-V experimental: panel 2, prueba 3

	Trazador	Multímetro	Diferencia
$V_{oc}$ [V]	18,12	18,30	0,18 (-0,98%)
$I_{sc}$ [A]	0,94	1,08	0,14 (-12,96%)
$P_{max}$ [W]	10,81	-	-
$V_{mp}$ [V]	14,18	-	-
$I_{mp}$ [A]	0,76	-	-

Tabla VIII.8 - Tabla de datos experimentales: panel 2, prueba 3

### Serie panel 1 y 2

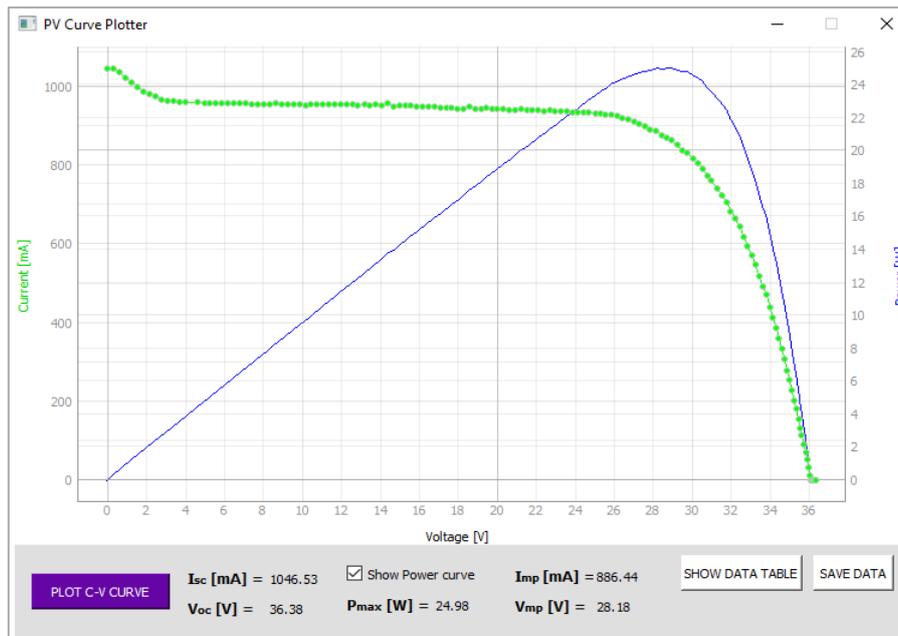


Figura VIII.12 - Curva I-V experimental: paneles en serie, prueba 3

	Trazador	Multímetro	Diferencia
$V_{oc}$ [V]	36,38	36,50	0,12 (-0,33%)
$I_{sc}$ [A]	1,05	1,13	0,08 (-7,08%)
$P_{max}$ [W]	24,98	-	-
$V_{mp}$ [V]	28,18	-	-
$I_{mp}$ [A]	0,89	-	-

Tabla VIII.9 - Tabla de datos experimentales: paneles en serie, prueba 3

### VIII.6 - Análisis de datos

Como se ha podido comprobar en los resultados anteriores con las correspondientes imágenes de las curvas y sus tablas de datos, podemos afirmar que, aunque con ciertas diferencias entre valores, el trazador diseñado realiza de manera correcta las medida y su posterior representación de la curva en la interfaz. Con referencia a lo anterior, las diferencias de medida máximas en estas pruebas han sido del 3% en tensión y 13% en corriente.

En ellas se puede comprobar que, a medida que se va reduciendo la irradiancia, se reduce también la corriente máxima que el panel puede suministrar. En la Tabla VIII.10 se puede visualizar y comparar los resultados de los puntos MPP de potencia máxima obtenidos en las diferentes pruebas.

	<b>Panel 1</b>	<b>Panel 2</b>	<b>Serie Panel 1 y 2</b>
<b>Prueba 1</b>	19,41	20,65	42,67
<b>Prueba 2</b>	13,41	14,60	28,18
<b>Prueba 3</b>	12,19	10,81	25,98

*Tabla VIII.10 - Tabla resultados experimentales: potencias máximas en W (vatios)*

Con estos resultados se puede demostrar que, a medida que se reduce la irradiancia, se reduce el punto de máxima potencia ofrecido por los paneles fotovoltaicos.

Por otro lado, comprobamos que, en las conexiones en serie de los paneles, se suman sus voltajes (18,5V + 18,5V), por lo que obtenemos alrededor de 37V en las tres pruebas, como se explicó en el Capítulo II.

Además, debemos tener en cuenta que, a medida que se realizaron las pruebas, la temperatura ambiente iba decreciendo. El voltaje que se obtuvo también disminuía con ella, acción inversa a lo que explicamos en la teoría de la curva fotovoltaica, Capítulo II. Indicamos que esta diferencia puede deberse a que las pruebas se han realizado bajo condiciones no controladas, con unos paneles antiguos, que pueden ocasionar múltiples fallos, y que, si se requiere afirmar de una manera inequívoca el funcionamiento del dispositivo, se deben realizar en laboratorio bajo condiciones controladas, algo que no se pudo conseguir bajo las circunstancias de realización de este proyecto.

## **CAPÍTULO IX: PRESUPUESTO**

### IX.1 - Costes materiales

Descripción	Cantidad	Coste unid. [€/u]	Coste Total
Microcontrolador ATmega328P	2	4,20	8,40 €
ADC MCP3201	2	5,90	11,80 €
TL082 Amplif. Operacional	1	4,20	4,20 €
Resistencia Shunt 75mV/10A	1	3,50	3,50 €
Módulo relé 5V Arduino	1	4,50	4,50 €
Condensador electrolítico 2,2mF	1	4,80	4,80 €
Diodo Schottky 15SQ045	2	3,40	6,80 €
Módulo Bluetooth HC-05	1	5,10	5,10 €
Módulo UART-USB	1	3,40	3,40 €
Resistencia cerámica 56Ω 11w	1	2,70	2,70 €
Resistencia 1kΩ	2	0,20	0,40 €
Resistencia 7,5kΩ	1	0,20	0,20 €
Resistencia 10kΩ	2	0,20	0,40 €
Resistencia 75kΩ	1	0,20	0,20 €
Resistencia 150kΩ	1	0,20	0,20 €
Condensador 22pF	2	0,25	0,50 €
Condensador 0,1uF	1	0,30	0,30 €
Condensador 0,33uF	1	0,30	0,30 €
Oscilador 16MHz	2	0,70	1,40 €
Buck-Boost	1	5,50	5,50 €
Regulador LM7805	1	1,40	1,40 €
Pila 9V	1	4,50	4,50 €
Cable protoboard	1	3,00	3,00 €
Cable silicona	1	3,50	3,50 €
Protoboard	1	15,50	15,50 €
Conector	2	0,20	0,40 €
<b>Total coste materiales</b>			<b>92,90 €</b>

### IX.2 - Costes mano de obra

Descripción	Cantidad [h]	Coste unid. [€/h]	Coste Total
Tiempo de análisis	70	65	4.550,00 €
Tiempo de codificación	220	70	15.400,00 €
Tiempo de implementación	180	35	6.300,00 €
Tiempo de documentación	60	20	1.200,00 €
<b>Total coste mano de obra</b>			<b>27.450,00 €</b>

### IX.3 - Costes totales

Descripción	Coste Total
Costes materiales (CM)	92,90 €
Costes mano de obra (CMO)	27.450,00 €
Gastos generales: 6% (CM+CMO)	1.652,57 €
Beneficio Industrial: 13% (CM+CMO)	3.580,58 €
<b>Coste total del proyecto</b>	<b>32.776,05 €</b>

## **CAPÍTULO X: APORTACIONES Y CONCLUSIONES**

The objective of this project has been to design and implementation of an electronic device for the characterization of photovoltaic panels with the following properties:

1. It allows obtaining the I-V and P-V characteristic curves for voltage values up to 100V and 10A.
2. From the previous curves, the system is able to obtain the corresponding values of the following parameters: Voc ("Open Circuit Voltage"), Isc ("Short Circuit Current"), Pmax ("Maximum Power"), Vmp ("Maximum Power Voltage") and Imp ("Maximum power current"), of enormous importance in the field of photovoltaic energy.
3. The control of the device by the user, is done through a friendly interface installed on a computer, designed for this project. The connection between the device and the computer can be made via USB or wirelessly via Bluetooth.
4. The interface allows to visualize the obtained curves and to store the experimental data, for additional processing or study, both in Excel file and in free software CSV.
5. Its low cost, simplicity and ease of construction.

Posibles mejoras del dispositivo:

1. Para obtener una mejor información de las curvas, sería importante dotar al dispositivo de sendos sensores que permitieran medir la temperatura superficial del panel, así como la irradiancia solar. Existen módulos en el mercado que realizan dichas funciones, fáciles de conectar a un microcontrolador y a un bajo coste. La utilización de ambos sensores es de gran importancia para obtener la curva con unos datos acorde a las normas STC, con una temperatura de célula e irradiancia solar determinadas.
2. Realizar un ajuste del dispositivo de medida efectuando la toma de datos en un ambiente controlado de irradiancia y temperatura, (por ejemplo, en un laboratorio) con los instrumentos de medida necesarios para ello, un panel o célula fotovoltaica y una lámpara que sea capaz de suministrar, al menos, 1000W/m<sup>2</sup>. Si en estas pruebas también se hace uso de un instrumento trazador de curvas I-V comercial, se podrán comparar los resultados y comprobar si el dispositivo diseñado realiza correctamente todas las medidas.
3. Dotar al dispositivo de un módulo Wi-Fi (basado en el ESP8266, con un coste inferior a los 5€), lo que permitiría su gestión vía Internet.
4. Desarrollar la misma interfaz de usuario para dispositivos móviles. Mediante ésta se podrá ordenar la toma de datos al instrumento y visualizar la curva obtenida, mediante comunicación Bluetooth o vía Wi-Fi.
5. Desarrollar y diseñar la PCB empleando tecnología SMD para reducir su tamaño final. Además, para una mejor presentación del dispositivo, diseñar una caja mediante impresión 3D donde alojar la PCB con sus módulos y conectores, con el fin de mejorar su transporte y manejo.
6. Implementar un módulo de almacenamiento de datos, mediante tarjeta SD, para realizar las medidas sin necesidad de conexión con el ordenador.
7. Realizar el cambio del módulo de conmutación mecánico a uno de estado sólido para mejorar eficiencia, seguridad y rapidez de conmutación.
8. Implementar en la interfaz gráfica la opción de abrir documentos de datos de curvas ya almacenadas para una nueva visualización.

## BIBLIOGRAFÍA

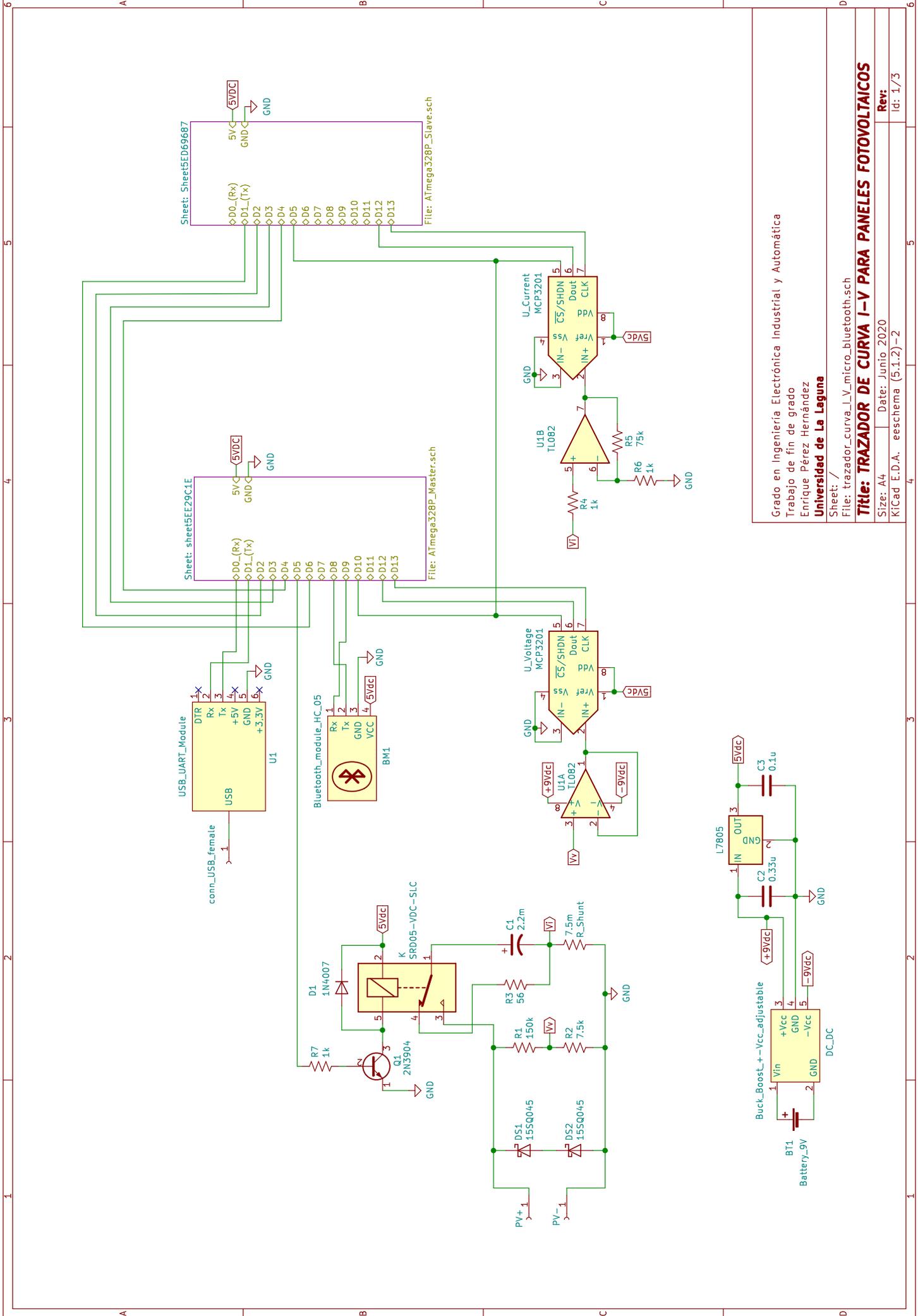
- [1] Organización meteorológica Mundial. *Datos sobre la contaminación*. Disponible en: <https://public.wmo.int/es> [05/2020]
- [2] EcoSolar. *La célula fotovoltaica*. Disponible en: <https://www.ecosolaresp.com/la-celula-fotovoltaica/> [04/2020]
- [3] Instructables Circuits. *IV Swinger 2*. Disponible en: <https://www.instructables.com/id/IV-Swinger-2-a-50-IV-Curve-Tracer/> [11/2019]
- [4] Colegio Oficial de Ing. Telecomunicación, (2007) *Energía solar fotovoltaica*, Inforama S.A.
- [5] Oscar Perpiñán L., (2018) *Energía solar fotovoltaica*, Creative Commons
- [6] Wikipedia. *Panel Fotovoltaico*. Disponible en: [https://es.wikipedia.org/wiki/Panel\\_fotovoltaico/](https://es.wikipedia.org/wiki/Panel_fotovoltaico/) [04/2020]
- [7] Ricardo Guerrero L., J. Manuel Martínez D., (2012) *Renewable Energies and CO2*, Springer
- [8] Energetika. *Curva I-V*. Disponible en: <http://www.energetika.com.ar/Curva%20I-V.html> [12/2019]
- [9] Autosolar. *Curva I-V*. Disponible en: <https://autosolar.es/blog/aspectos-tecnicos/la-curva-de-intensidad-voltaje-y-la-de-potencia-voltaje-de-un-panel-solar-el-papel-del-regulador-de-carga> [12/2019]
- [10] Arduino. *¿Qué es Arduino?* Disponible en: <https://arduino.cl/que-es-arduino/> [01/2020]
- [11] Internet archive. *Introducción a Arduino*. Disponible en: [https://web.archive.org/web/20180403132515/http://www.princeton.edu/~ffab/media\\_downloads\\_files/IntroArduinoBook.pdf](https://web.archive.org/web/20180403132515/http://www.princeton.edu/~ffab/media_downloads_files/IntroArduinoBook.pdf)<https://aprendiendoarduino.wordpress.com/tag/esquemas-electricos/> [01/2020]
- [12] Aprendiendo Arduino. *Entradas y salidas del Arduino*. Disponible en: <https://aprendiendoarduino.wordpress.com/2019/08/28/entradas-y-salidas-en-arduino/> [01/2020]
- [13] All Datasheet. *ATmega328P*. Disponible en: <https://www.alldatasheet.es/datasheet-pdf/pdf/392244/ATMEL/ATMEGA328P.html> [01/2020]
- [14] Instituto Newton C. Braga. *Conociendo el microcontrolador ATmega328P*. Disponible en: <https://www.incb.com.mx/index.php/articulos/78-microcontroladores-y-dsps/2546-conociendo-el-microcontrolador-nucleo-core-atmega328p-de-arduino-uno-mic019s> [02/2020]
- [15] OpenWebinars. *Comunicación serie con Arduino*. Disponible en: <https://openwebinars.net/blog/tutorial-arduino-comunicacion-serie/> [02/2020]
- [16] Tbem. *Protocolo I2C*. Disponible en: <https://teslabem.com/nivel-intermedio/fundamentos/> [01/2020]
- [17] SparkFun. *Protocolo SPI*. Disponible en: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all> [01/2020]
- [18] Arduino. Software. Disponible en: <https://www.arduino.cc/en/main/software> [12/2019]
- [19] Anaconda. Software. Disponible en:

- <https://www.anaconda.com/products/individual> [02/2020]
- [20] Spyder. Software. Disponible en:  
<https://www.spyder-ide.org/> [02/2020]
- [21] Medium. *Primeros pasos con PyQt Designer*. Disponible en:  
<https://medium.com/@hektorprofe/primeros-pasos-en-pyqt-5-y-qt-designer-programas-gr%C3%A1ficos-con-python-6161fba46060> [04/2020]
- [22] KiCad. *Software*. Disponible en:  
<https://kicad-pcb.org/> [04/2020]
- [23] Robert L. Boylestad, (2004) *Introducción al análisis de circuitos*, Pearson Educación
- [24] All Datasheet. *MCP3201*. Disponible en:  
<https://pdf1.alldatasheet.com/datasheet-pdf/view/303761/MICROCHIP/MCP3201.html>  
[01/2020]
- [25] Lógica Programable. *Conversor ADC MCP3201*. Disponible en:  
<https://www.logicaprogrammabile.it/convertitore-a-d-mcp3201/> [01/2020]
- [26] HWPro. *Datasheet resistencia shunt*. Disponible en:  
<http://www.hwpro.cz/oc/image/catalog/datasheets/FL2.pdf> [02/2020]
- [27] All Datasheet. *TL082*. Disponible en:  
<https://pdf1.alldatasheet.com/datasheet-pdf/view/25384/STMICROELECTRONICS/TL082.html>  
[04/2020]
- [28] All Datasheet. *SRD-05VDC-SL-C*. Disponible en:  
<https://www.alldatasheet.com/datasheet-pdf/pdf/1131944/SONGLERELAY/SRD05VDCSLC.html> [02/2020]
- [29] Luis Llamas. *Módulo relé de Arduino*. Disponible en:  
<https://www.luisllamas.es/arduino-salida-rele/> [02/2020]
- [30] All Datasheet. *Regulador LM7805*. Disponible en:  
<https://pdf1.alldatasheet.es/datasheet-pdf/view/105702/FAIRCHILD/LM7805.html> [04/2020]
- [31] Prometec. *Módulo Bluetooth HC-05*. Disponible en:  
<https://www.prometec.net/bt-hc05/> [05/2020]
- [32] NayLamp. *Módulo CP2102 Conversor USB a Serial*. Disponible en:  
<https://naylampmechatronics.com/conversores-ttl/79-modulo-cp2102-conversor-usb-a-serial-ttl.html> [05/2020]
- [33] HD Meteo La Palma. *Estaciones meteorológicas de la isla a tiempo real*. Disponible en:  
<https://lapalma.hdmeteo.com/> [05/2020]

## GLOSARIO

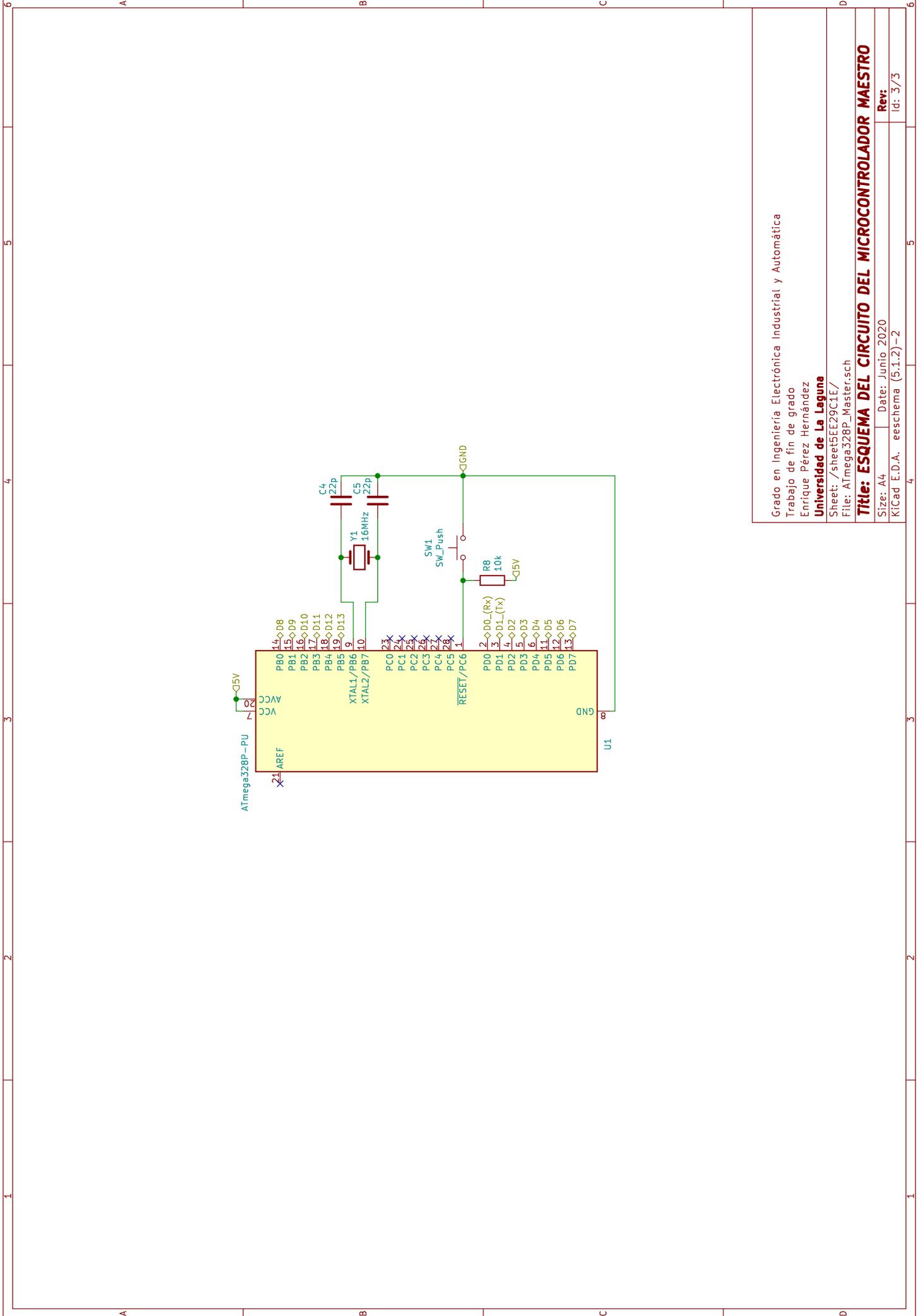
1. ADC: Conversor Analógico-Digital
2. CPU: Unidad Central de Procesamiento
3. CSV: Archivo de Valores separados por comas
4. EEPROM: Memoria de Solo Lectura Programable y Borrable Electrónicamente
5. FV: Fotovoltaica
6. GND: Tierra
7. GUI: Interfaz Gráfica de Usuario
8. I2C: Circuito Inter-Integrado
9. IDE: Entorno Desarrollo Integrado
10. Imp: Corriente en el Punto de Máxima Potencia
11. Isc: Corriente de Cortocircuito
12. I-V: Corriente - Voltaje
13. MISO: Salida del Esclavo y Entrada al Maestro
14. MOSI: Salida del Maestro y Entrada al Esclavo
15. MPPT: Seguidores del Punto de Máxima Potencia
16. OMM: Organización Meteorológica Mundial
17. P-V: Potencia - Voltaje
18. PC: Ordenador Personal
19. PCB: Placa de circuito impreso
20. Pmax: Punto de máxima potencia
21. PV: Fotovoltaica
22. PWM: Modulación por Anchura de Pulso
23. RX: Recepción
24. SCL: Señal de Reloj Serial
25. SD: Secure Digital
26. SDA: Señal de Datos
27. SDRAM: Memoria de Acceso Aleatorio Síncrona y Dinámica
28. SMD: Dispositivos de Montaje Superficial
29. SPI: Interfaz Periférica Serial
30. SS: Señal de Selección
31. STC: Condiciones Estándar de Test
32. TX: Transmisión
33. UART: Transmisor-Receptor Asíncrono Universal
34. USB: Bus Universal en Serie
35. Vmp: Voltaje en el Punto de Máxima Potencia
36. Voc: Voltaje de Circuito Abierto
37. XLSX: Archivo de Microsoft Excel

## **ANEXOS**



Grado en Ingeniería Electrónica Industrial y Automática  
 Trabajo de fin de grado  
 Enrique Pérez Hernández  
**Universidad de La Laguna**

Sheet: /  
 File: trazador\_curva\_v\_micro\_bluetooth.sch  
**Title: TRAZADOR DE CURVA I-V PARA PANELES FOTOVOLTAICOS**  
 Size: A4 | Date: Junio 2020  
 KiCad E.D.A. eschema (5.1.2)-2



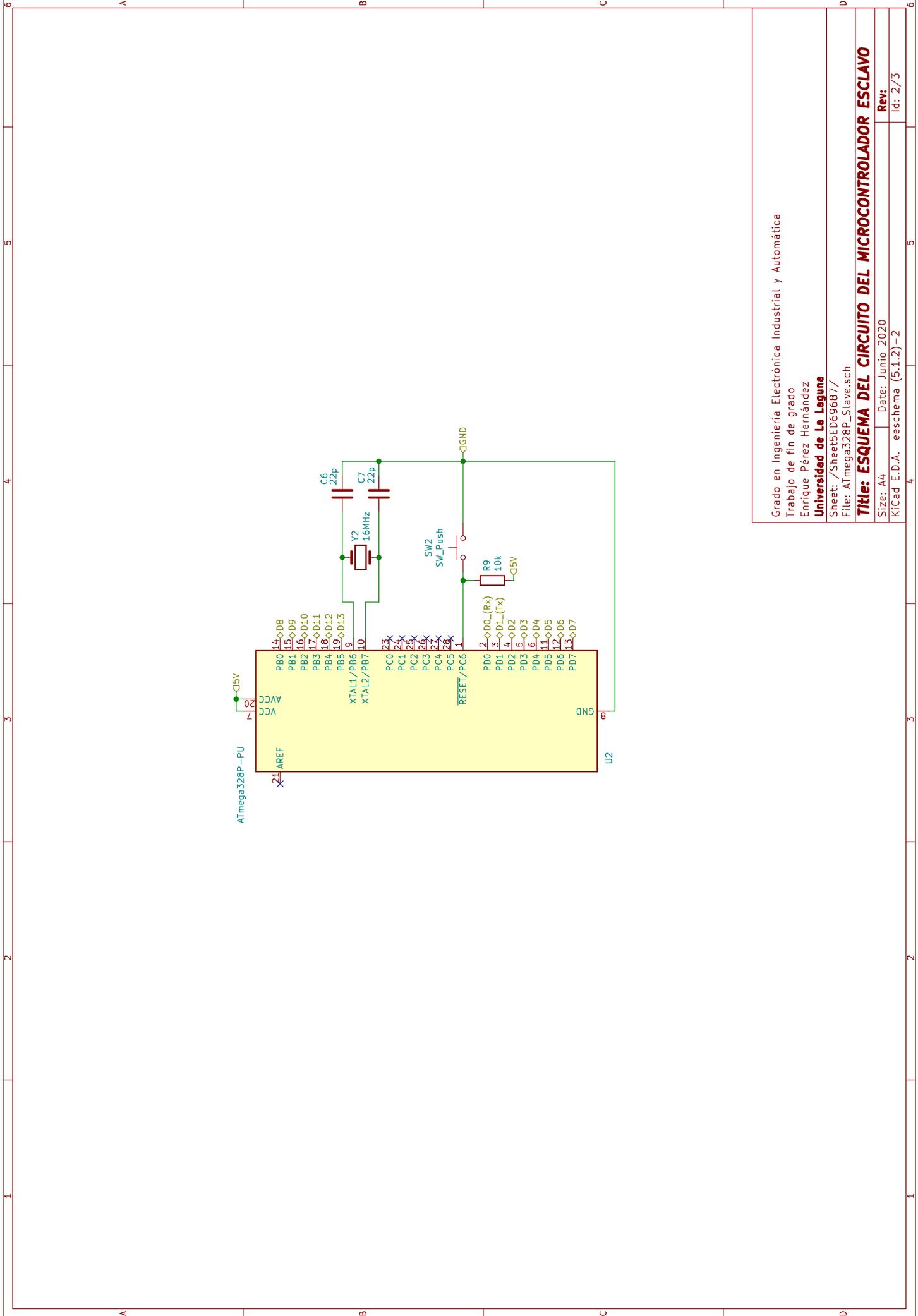
Grado en Ingeniería Electrónica Industrial y Automática  
 Trabajo de fin de grado  
 Enrique Pérez Hernández  
**Universidad de La Laguna**

Sheet: /sheet5EE29C1E/  
 File: ATmega328P\_Master.sch

**Title: ESQUEMA DEL CIRCUITO DEL MICROCONTROLADOR MAESTRO**

Size: A4 | Date: Junio 2020  
 KICad: E.D.A. eeschema (5.1.2)-2

Rev:  
 Id: 3/3



Grado en Ingeniería Electrónica Industrial y Automática  
 Trabajo de fin de grado  
 Enrique Pérez Hernández  
**Universidad de La Laguna**  
 Sheet: /Sheet5ED69687/  
 File: ATmega328P\_Slave.sch

**Title: ESQUEMA DEL CIRCUITO DEL MICROCONTROLADOR ESCLAVO**

Size: A4 | Date: Junio 2020 | Rev: |  
 KICad: E.D.A. | eeschema (5.1.2)-2 | Id: 2/3

## Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1KBytes EEPROM
  - 512/1K/1K/2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
    - Programming Lock for Software Security
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix® acquisition
  - Up to 64 sense channels
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package  
Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package  
Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 4MHz@1.8 - 5.5V, 0 - 10MHz@2.7 - 5.5V, 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
  - Active Mode: 0.2mA
  - Power-down Mode: 0.1µA
  - Power-save Mode: 0.75µA (Including 32kHz RTC)



## 8-bit Atmel Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

**ATmega48A**  
**ATmega48PA**  
**ATmega88A**  
**ATmega88PA**  
**ATmega168A**  
**ATmega168PA**  
**ATmega328**  
**ATmega328P**

Rev. 8271D-AVR-05/11



## 1. Pin Configurations

Figure 1-1. Pinout ATmega48A/PA/88A/PA/168A/PA/328/P

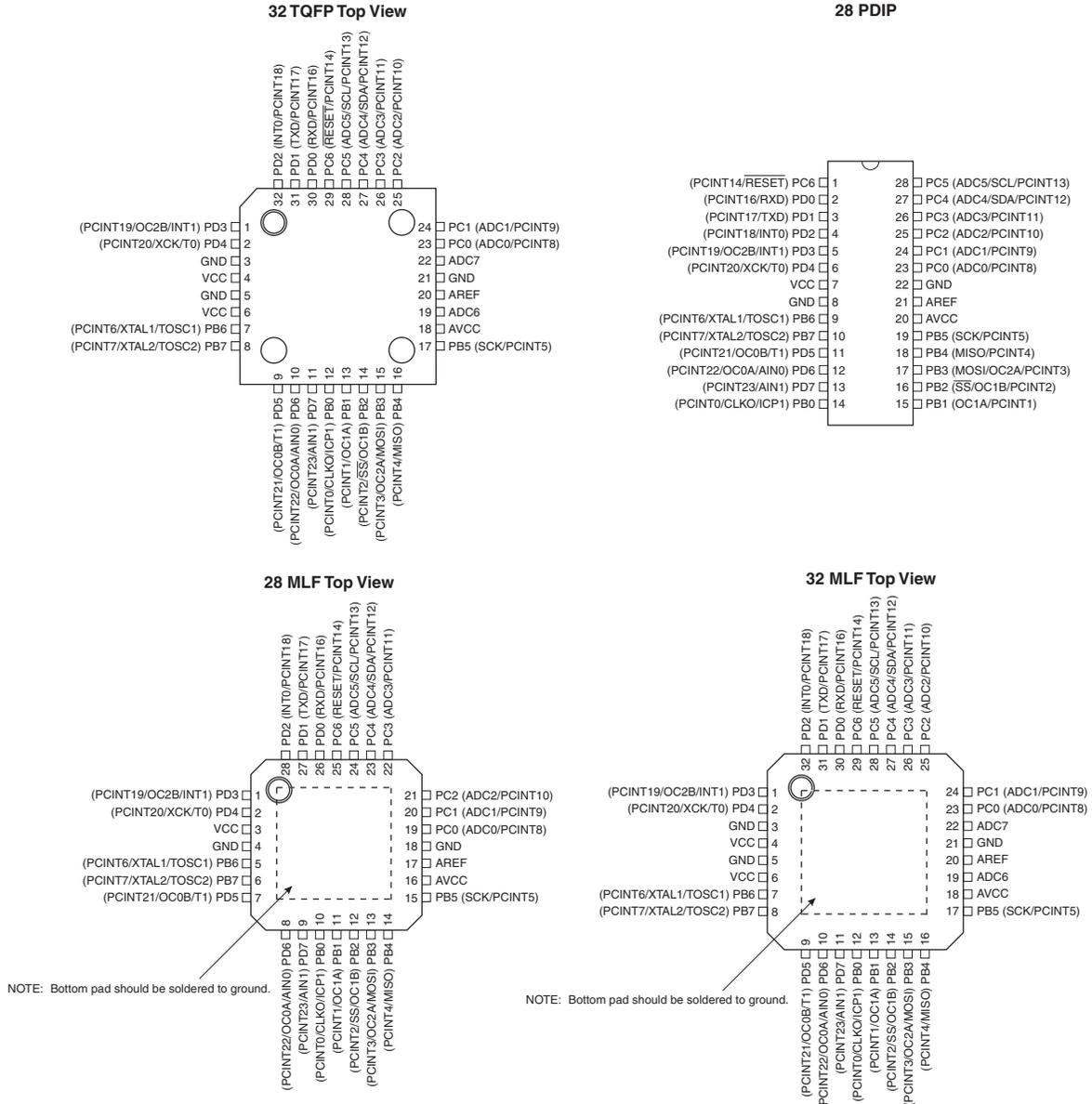


Table 1-1. 32UFBGA - Pinout ATmega48A/48PA/88A/88PA/168A/168PA

	1	2	3	4	5	6
A	PD2	PD1	PC6	PC4	PC2	PC1
B	PD3	PD4	PD0	PC5	PC3	PC0
C	GND	GND			ADC7	GND
D	VDD	VDD			AREF	ADC6
E	PB6	PD6	PB0	PB2	AVDD	PB5
F	PB7	PD5	PD7	PB1	PB3	PB4

## 2.7V 12-Bit A/D Converter with SPI Serial Interface

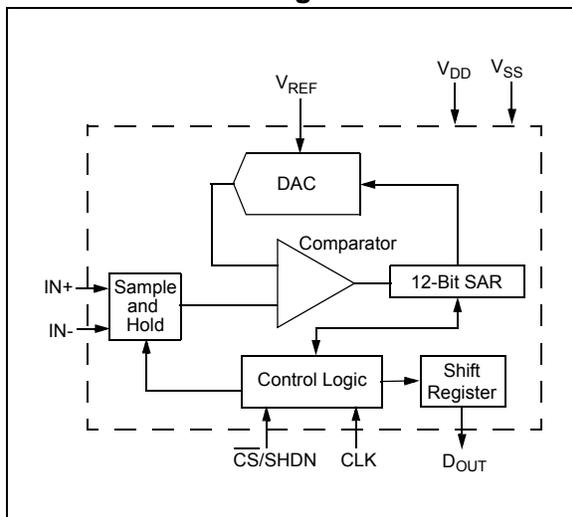
### Features

- 12-bit resolution
- $\pm 1$  LSB max DNL
- $\pm 1$  LSB max INL (MCP3201-B)
- $\pm 2$  LSB max INL (MCP3201-C)
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 100 ksp/s maximum sampling rate at  $V_{DD} = 5V$
- 50 ksp/s maximum sampling rate at  $V_{DD} = 2.7V$
- Low power CMOS technology
- 500 nA typical standby current, 2  $\mu A$  maximum
- 400  $\mu A$  maximum active current at 5V
- Industrial temp range:  $-40^{\circ}C$  to  $+85^{\circ}C$
- 8-pin MSOP, PDIP, SOIC and TSSOP packages

### Applications

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

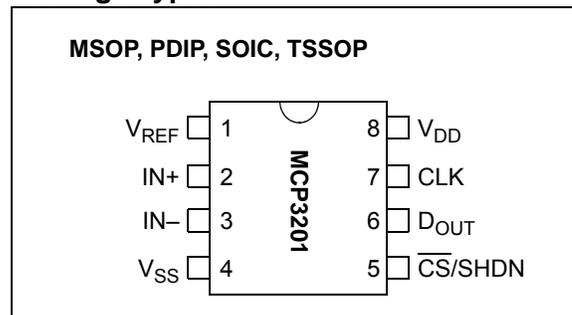
### Functional Block Diagram



### Description

The Microchip Technology Inc. MCP3201 device is a successive approximation 12-bit Analog-to-Digital (A/D) Converter with on-board sample and hold circuitry. The device provides a single pseudo-differential input. Differential Nonlinearity (DNL) is specified at  $\pm 1$  LSB, and Integral Nonlinearity (INL) is offered in  $\pm 1$  LSB (MCP3201-B) and  $\pm 2$  LSB (MCP3201-C) versions. Communication with the device is done using a simple serial interface compatible with the SPI protocol. The device is capable of sample rates of up to 100 ksp/s at a clock rate of 1.6 MHz. The MCP3201 device operates over a broad voltage range (2.7V - 5.5V). Low-current design permits operation with typical standby and active currents of only 500 nA and 300  $\mu A$ , respectively. The device is offered in 8-pin MSOP, PDIP, TSSOP and 150 mil SOIC packages.

### Package Types



## 1.0 ELECTRICAL CHARACTERISTICS

### 1.1 Maximum Ratings†

V <sub>DD</sub> .....	7.0V
All inputs and outputs w.r.t. V <sub>SS</sub> .....	-0.6V to V <sub>DD</sub> +0.6V
Storage temperature .....	-65°C to +150°C
Ambient temp. with power applied .....	-65°C to +125°C
ESD protection on all pins (HBM) .....	> 4 kV

†**Notice:** Stresses above those listed under “Maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

## ELECTRICAL CHARACTERISTICS

**Electrical Specifications:** All parameters apply at V<sub>DD</sub> = 5V, V<sub>SS</sub> = 0V, V<sub>REF</sub> = 5V, T<sub>A</sub> = -40°C to +85°C, f<sub>SAMPLE</sub> = 100 ksps, and f<sub>CLK</sub> = 16\*f<sub>SAMPLE</sub>, unless otherwise noted.

Parameter	Sym	Min	Typ	Max	Units	Conditions
<b>Conversion Rate:</b>						
Conversion Time	t <sub>CONV</sub>	—	—	12	clock cycles	
Analog Input Sample Time	t <sub>SAMPLE</sub>	1.5			clock cycles	
Throughput Rate	f <sub>SAMPLE</sub>	—	—	100 50	ksps ksps	V <sub>DD</sub> = V <sub>REF</sub> = 5V V <sub>DD</sub> = V <sub>REF</sub> = 2.7V
<b>DC Accuracy:</b>						
Resolution		12			bits	
Integral Nonlinearity	INL	—	±0.75 ±1	±1 ±2	LSB LSB	MCP3201-B MCP3201-C
Differential Nonlinearity	DNL	—	±0.5	±1	LSB	No missing codes over temperature
Offset Error		—	±1.25	±3	LSB	
Gain Error		—	±1.25	±5	LSB	
<b>Dynamic Performance:</b>						
Total Harmonic Distortion	THD	—	-82	—	dB	V <sub>IN</sub> = 0.1V to 4.9V@1 kHz
Signal to Noise and Distortion (SINAD)	SINAD	—	72	—	dB	V <sub>IN</sub> = 0.1V to 4.9V@1 kHz
Spurious Free Dynamic Range	SFDR	—	86	—	dB	V <sub>IN</sub> = 0.1V to 4.9V@1 kHz
<b>Reference Input:</b>						
Voltage Range		0.25	—	V <sub>DD</sub>	V	<b>Note 2</b>
Current Drain		—	100 .001	150 3	µA µA	CS = V <sub>DD</sub> = 5V
<b>Analog Inputs:</b>						
Input Voltage Range (IN+)	IN+	IN-	—	V <sub>REF</sub> +IN-	V	
Input Voltage Range (IN-)	IN-	V <sub>SS</sub> -100		V <sub>SS</sub> +100	mV	
Leakage Current		—	0.001	±1	µA	
Switch Resistance	R <sub>SS</sub>	—	1K	—	W	See <a href="#">Figure 4-1</a>
Sample Capacitor	C <sub>SAMPLE</sub>	—	20	—	pF	See <a href="#">Figure 4-1</a>
<b>Digital Input/Output:</b>						
Data Coding Format		Straight Binary				
High Level Input Voltage	V <sub>IH</sub>	0.7 V <sub>DD</sub>	—	—	V	
Low Level Input Voltage	V <sub>IL</sub>	—	—	0.3 V <sub>DD</sub>	V	

**Note 1:** This parameter is established by characterization and not 100% tested.

**2:** See graph that relates linearity performance to V<sub>REF</sub> level.

**3:** Because the sample cap will eventually lose charge, effective clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures. See **Section 6.2 “Maintaining Minimum Clock Speed”** for more information.

# MCP3201

## ELECTRICAL CHARACTERISTICS (CONTINUED)

**Electrical Specifications:** All parameters apply at  $V_{DD} = 5V$ ,  $V_{SS} = 0V$ ,  $V_{REF} = 5V$ ,  $T_A = -40^{\circ}C$  to  $+85^{\circ}C$ ,  $f_{SAMPLE} = 100$  kpsps, and  $f_{CLK} = 16 * f_{SAMPLE}$ , unless otherwise noted.

Parameter	Sym	Min	Typ	Max	Units	Conditions
High Level Output Voltage	$V_{OH}$	4.1	—	—	V	$I_{OH} = -1$ mA, $V_{DD} = 4.5V$
Low Level Output Voltage	$V_{OL}$	—	—	0.4	V	$I_{OL} = 1$ mA, $V_{DD} = 4.5V$
Input Leakage Current	$I_{LI}$	-10	—	10	$\mu A$	$V_{IN} = V_{SS}$ or $V_{DD}$
Output Leakage Current	$I_{LO}$	-10	—	10	$\mu A$	$V_{OUT} = V_{SS}$ or $V_{DD}$
Pin Capacitance (all inputs/outputs)	$C_{IN}$ , $C_{OUT}$	—	—	10	pF	$V_{DD} = 5.0V$ ( <b>Note 1</b> ) $T_A = +25^{\circ}C$ , $f = 1$ MHz
<b>Timing Parameters:</b>						
Clock Frequency	$f_{CLK}$	—	—	1.6 0.8	MHz MHz	$V_{DD} = 5V$ ( <b>Note 3</b> ) $V_{DD} = 2.7V$ ( <b>Note 3</b> )
Clock High Time	$t_{HI}$	312	—	—	ns	
Clock Low Time	$t_{LO}$	312	—	—	ns	
$\overline{CS}$ Fall To First Rising CLK Edge	$t_{SUCS}$	100	—	—	ns	
CLK Fall To Output Data Valid	$t_{DO}$	—	—	200	ns	See Test Circuits, <a href="#">Figure 1-2</a>
CLK Fall To Output Enable	$t_{EN}$	—	—	200	ns	See Test Circuits, <a href="#">Figure 1-2</a>
$\overline{CS}$ Rise To Output Disable	$t_{DIS}$	—	—	100	ns	See Test Circuits, <a href="#">Figure 1-2</a> ( <b>Note 1</b> )
$\overline{CS}$ Disable Time	$t_{CSH}$	625	—	—	ns	
$D_{OUT}$ Rise Time	$t_R$	—	—	100	ns	See Test Circuits, <a href="#">Figure 1-2</a> ( <b>Note 1</b> )
$D_{OUT}$ Fall Time	$t_F$	—	—	100	ns	See Test Circuits, <a href="#">Figure 1-2</a> ( <b>Note 1</b> )
<b>Power Requirements:</b>						
Operating Voltage	$V_{DD}$	2.7	—	5.5	V	
Operating Current	$I_{DD}$	—	300 210	400 —	$\mu A$ $\mu A$	$V_{DD} = 5.0V$ , $D_{OUT}$ unloaded $V_{DD} = 2.7V$ , $D_{OUT}$ unloaded
Standby Current	$I_{DDS}$	—	0.5	2	$\mu A$	$CS = V_{DD} = 5.0V$

**Note 1:** This parameter is established by characterization and not 100% tested.

**2:** See graph that relates linearity performance to  $V_{REF}$  level.

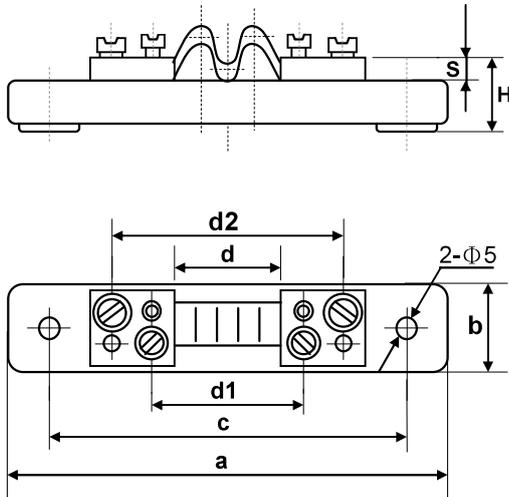
**3:** Because the sample cap will eventually lose charge, effective clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures. See [Section 6.2](#) “Maintaining Minimum Clock Speed” for more information.

## TEMPERATURE CHARACTERISTICS

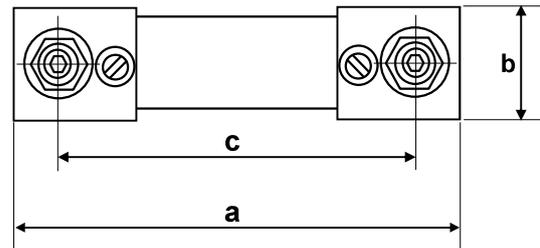
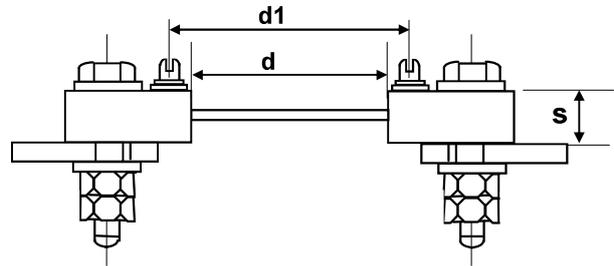
**Electrical Specifications:** Unless otherwise indicated,  $V_{DD} = +2.7V$  to  $+5.5V$ ,  $V_{SS} = GND$ .

Parameters	Sym	Min	Typ	Max	Units	Conditions
<b>Temperature Ranges</b>						
Specified Temperature Range	$T_A$	-40	—	+85	$^{\circ}C$	
Operating Temperature Range	$T_A$	-40	—	+85	$^{\circ}C$	
Storage Temperature Range	$T_A$	-65	—	+150	$^{\circ}C$	
<b>Thermal Package Resistances</b>						
Thermal Resistance, 8L-MSOP	$\theta_{JA}$	—	211	—	$^{\circ}C/W$	
Thermal Resistance, 8L-PDIP	$\theta_{JA}$	—	89.5	—	$^{\circ}C/W$	
Thermal Resistance, 8L-SOIC	$\theta_{JA}$	—	149.5	—	$^{\circ}C/W$	
Thermal Resistance, 8L-TSSOP	$\theta_{JA}$	—	139	—	$^{\circ}C/W$	

▶ FL-2 (5A-100A, Unit: mm)



**5A ~ 50A**



**75A ~ 100A**

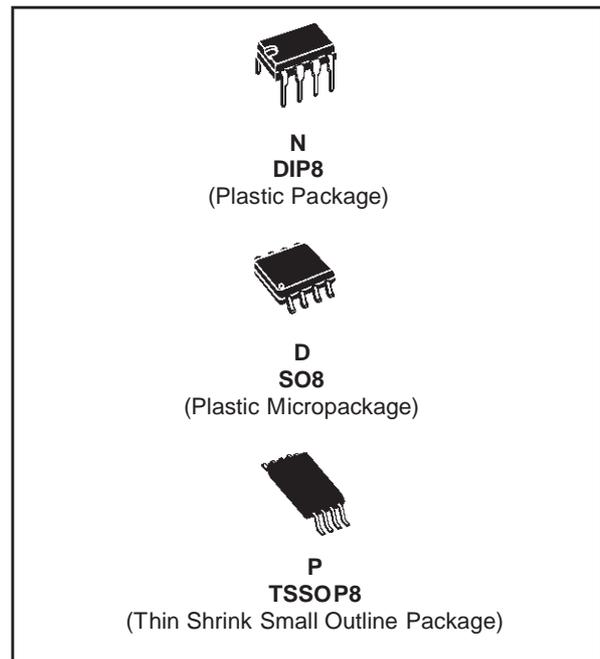
Rating Current (A)	Dimensions (Unit: mm)										High Current Bolt (mm)-PC (Option)	Shunt Voltage Bolt (mm)-PC (Option)
	a	c	b	c1	c2	H	S	d	d1	d2		
5	120	100	20			15	5	30	42	60	M5x6-2	M5x6-2
10	120	100	20			15	5	30	42	60	M5x6-2	
15	120	100	20			15	5	30	42	60	M5x6-2	
20	120	100	20			15	5	30	42	60	M5x6-2	
30	120	100	20			15	5	30	42	60	M5x6-2	
50	120	100	20			15	5	30	42	60	M5x6-2	
75	104	85	22			22	10	40	53		M8x35-2	
100	104	85	22			22	10	40	53		M8x35-2	



# TL082 TL082A - TL082B

## GENERAL PURPOSE J-FET DUAL OPERATIONAL AMPLIFIERS

- WIDE COMMON-MODE (UP TO  $V_{CC}^+$ ) AND DIFFERENTIAL VOLTAGE RANGE
- LOW INPUT BIAS AND OFFSET CURRENT
- OUTPUT SHORT-CIRCUIT PROTECTION
- HIGH INPUT IMPEDANCE J-FET INPUT STAGE
- INTERNAL FREQUENCY COMPENSATION
- LATCH UP FREE OPERATION
- HIGH SLEW RATE : 16V/ $\mu$ s (typ)

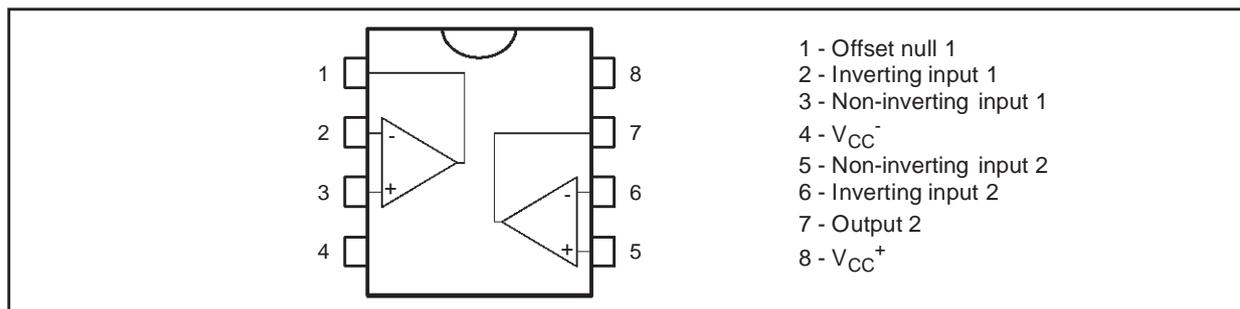


### DESCRIPTION

The TL082, TL082A and TL082B are high speed J-FET input dual operational amplifiers incorporating well matched, high voltage J-FET and bipolar transistors in a monolithic integrated circuit.

The devices feature high slew rates, low input bias and offset current, and low offset voltage temperature coefficient.

### PIN CONNECTIONS (top view)



### ORDER CODE

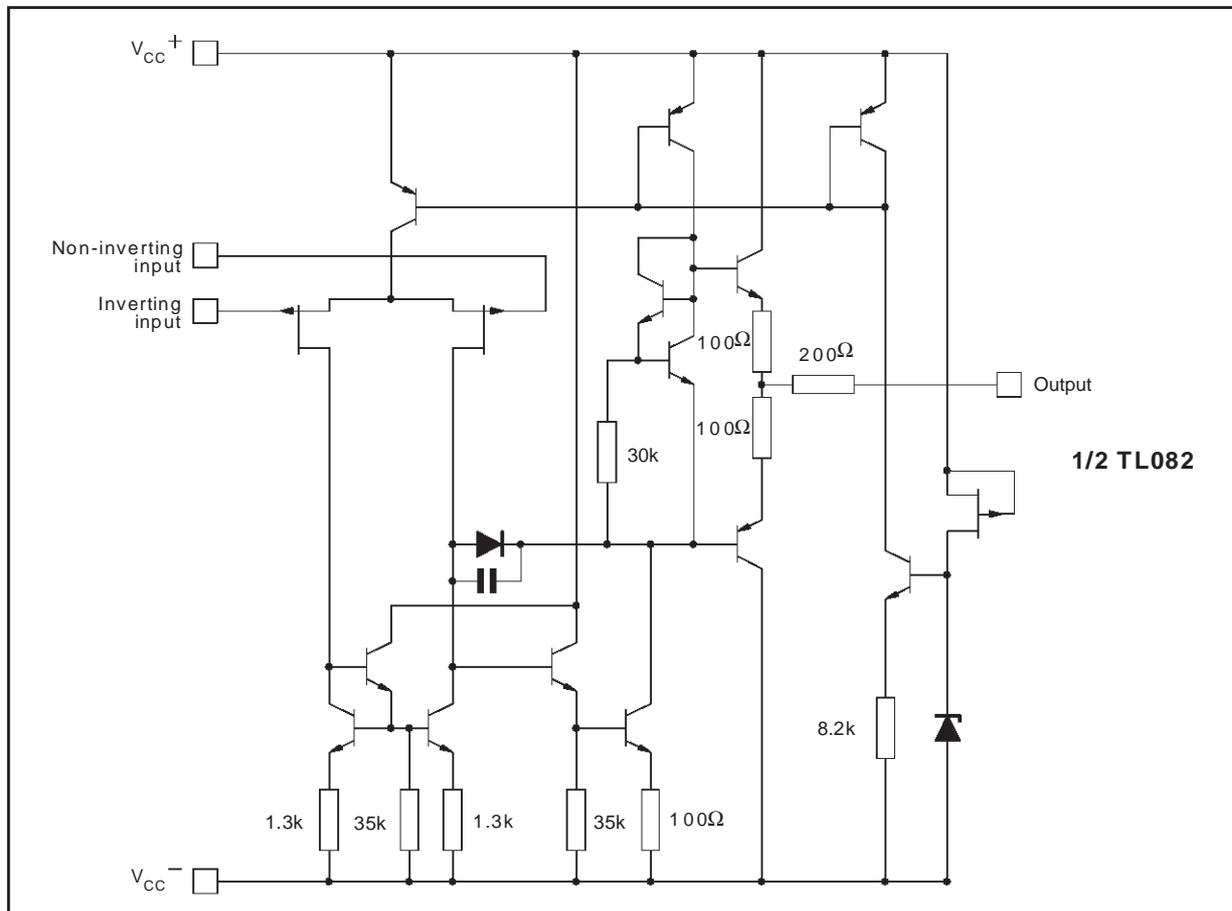
Part Number	Temperature Range	Package		
		N	D	P
TL082M/AM/BM	-55°C, +125°C	•	•	•
TL082I/AI/BI	-40°C, +105°C	•	•	•
TL082C/AC/BC	0°C, +70°C	•	•	•

**Example** : TL082CD, TL082IN

**N** = Dual in Line Package (DIP)  
**D** = Small Outline Package (SO) - also available in Tape & Reel (DT)  
**P** = Thin Shrink Small Outline Package (TSSOP) - only available in Tape & Reel (PT)

## TL082 - TL082A - TL082B

### SCHEMATIC DIAGRAM



### ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	TL082M, AM, BM	TL082I, AI, BI	TL082C, AC, BC	Unit
$V_{CC}$	Supply voltage - note <sup>1)</sup>	±18			V
$V_i$	Input Voltage - note <sup>2)</sup>	±15			V
$V_{id}$	Differential Input Voltage - note <sup>3)</sup>	±30			V
$P_{tot}$	Power Dissipation	680			mW
	Output Short-circuit Duration - note <sup>4)</sup>	Infinite			
$T_{oper}$	Operating Free-air Temperature Range	-55 to +125	-40 to +105	0 to +70	°C
$T_{stg}$	Storage Temperature Range	-65 to +150			°C

- All voltage values, except differential voltage, are with respect to the zero reference level (ground) of the supply voltages where the zero reference level is the midpoint between  $V_{CC}^+$  and  $V_{CC}^-$ .
- The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 volts, whichever is less.
- Differential voltages are the non-inverting input terminal with respect to the inverting input terminal.
- The output may be shorted to ground or to either supply. Temperature and/or supply voltages must be limited to ensure that the dissipation rating is not exceeded.

**ELECTRICAL CHARACTERISTICS**

$V_{CC} = \pm 15V$ ,  $T_{amb} = +25^{\circ}C$  (unless otherwise specified)

Symbol	Parameter	TL082I,M,AC,AI,AM, BC,BI,BM			TL082C			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
$V_{io}$	Input Offset Voltage ( $R_S = 50\Omega$ ) $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$ TL082 TL082A TL082B TL082 TL082A TL082B		3 3 1	10 6 3 13 7 5		3	10 13	mV
$DV_{io}$	Input Offset Voltage Drift		10			10		$\mu V/^{\circ}C$
$I_{io}$	Input Offset Current - note 1) $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$		5	100 4		5	100 10	pA nA
$I_{ib}$	Input Bias Current -note 1 $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$		20	200 20		20	400 20	pA nA
$A_{vd}$	Large Signal Voltage Gain ( $R_L = 2k\Omega$ , $V_o = \pm 10V$ ) $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$	50 25	200		25 15	200		V/mV
SVR	Supply Voltage Rejection Ratio ( $R_S = 50\Omega$ ) $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$	80 80	86		70 70	86		dB
$I_{CC}$	Supply Current, no load $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$		1.4	2.5 2.5		1.4	2.5 2.5	mA
$V_{icm}$	Input Common Mode Voltage Range	$\pm 11$	+15 -12		$\pm 11$	+15 -12		V
CMR	Common Mode Rejection Ratio ( $R_S = 50\Omega$ ) $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$	80 80	86		70 70	86		dB
$I_{os}$	Output Short-circuit Current $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$	10 10	40	60 60	10 10	40	60 60	mA
$\pm V_{opp}$	Output Voltage Swing $T_{amb} = +25^{\circ}C$ $T_{min} \leq T_{amb} \leq T_{max}$ RL = 2k $\Omega$ RL = 10k $\Omega$ RL = 2k $\Omega$ RL = 10k $\Omega$	10 12 10 12	12 13.5		10 12 10 12	12 13.5		V
SR	Slew Rate ( $T_{amb} = +25^{\circ}C$ ) $V_{in} = 10V$ , $R_L = 2k\Omega$ , $C_L = 100pF$ , unity gain	8	16		8	16		V/ $\mu s$
$t_r$	Rise Time ( $T_{amb} = +25^{\circ}C$ ) $V_{in} = 20mV$ , $R_L = 2k\Omega$ , $C_L = 100pF$ , unity gain		0.1			0.1		$\mu s$
$K_{ov}$	Overshoot ( $T_{amb} = +25^{\circ}C$ ) $V_{in} = 20mV$ , $R_L = 2k\Omega$ , $C_L = 100pF$ , unity gain		10			10		%
GBP	Gain Bandwidth Product ( $T_{amb} = +25^{\circ}C$ ) $V_{in} = 10mV$ , $R_L = 2k\Omega$ , $C_L = 100pF$ , $f = 100kHz$	2.5	4		2.5	4		MHz
$R_i$	Input Resistance		$10^{12}$			$10^{12}$		$\Omega$

# SONGLE RELAY

	<p>RELAY ISO9002</p>	<p><b>SRD</b></p>
---	----------------------	-------------------



## 1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for highdensity P.C. board mounting technique.
- UL,CUL,TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

## 2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.  
( Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

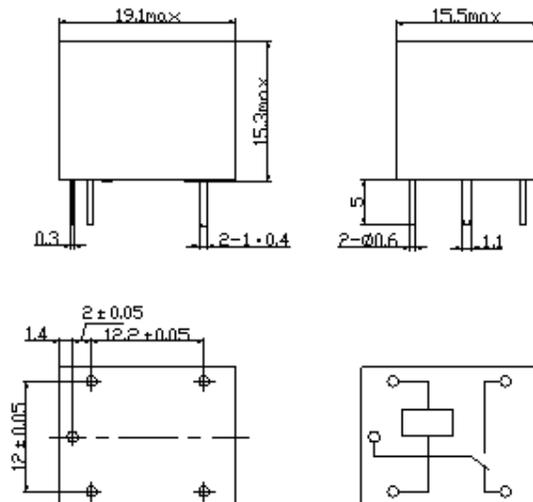
## 3. ORDERING INFORMATION

SRD	XX VDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil sensitivity	Contact form
SRD	03、05、06、09、12、24、48VDC	S:Sealed type	L:0.36W	A:1 form A
		F:Flux free type	D:0.45W	B:1 form B C:1 form C

## 4. RATING

CCC	FILE NUMBER:CH0052885-2000	7A/240VDC
CCC	FILE NUMBER:CH0036746-99	10A/250VDC
UL/CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R9933789	10A/240VAC 28VDC

## 5. DIMENSION (unit:mm) DRILLING (unit:mm) WIRING DIAGRAM



## 6. COIL DATA CHART (AT20°C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance ( $\Omega$ ) $\pm 10\%$	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75%Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
	48	48	7.5	6400				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500	abt. 0.51W			

## 7. CONTACT RATING

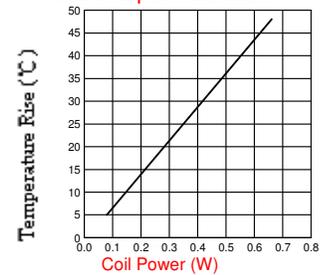
Item	Type	SRD	
		FORM C	FORM A
Contact Capacity Resistive Load ( $\cos\Phi=1$ )		7A 28VDC 10A 125VAC 7A 240VAC	10A 28VDC 10A 240VAC
Inductive Load ( $\cos\Phi=0.4$ L/R=7msec)		3A 120VAC 3A 28VDC	5A 120VAC 5A 28VDC
Max. Allowable Voltage		250VAC/110VDC	250VAC/110VDC
Max. Allowable Power Force		800VAC/240W	1200VA/300W
Contact Material		AgCdO	AgCdO

## 8. PERFORMANCE (at initial value)

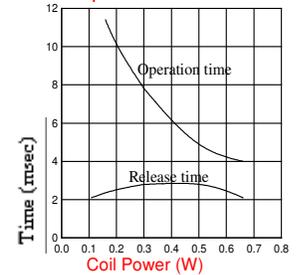
Item	Type	SRD
Contact Resistance		100m $\Omega$ Max.
Operation Time		10msec Max.
Release Time		5msec Max.
Dielectric Strength		
Between coil & contact		1500VAC 50/60HZ (1 minute)
Between contacts		1000VAC 50/60HZ (1 minute)
Insulation Resistance		100 M $\Omega$ Min. (500VDC)
Max. ON/OFF Switching		
Mechanically		300 operation/min
Electrically		30 operation/min
Ambient Temperature		-25°C to +70°C
Operating Humidity		45 to 85% RH
Vibration		
Endurance		10 to 55Hz Double Amplitude 1.5mm
Error Operation		10 to 55Hz Double Amplitude 1.5mm
Shock		
Endurance		100G Min.
Error Operation		10G Min.
Life Expectancy		
Mechanically		10 <sup>7</sup> operations. Min. (no load)
Electrically		10 <sup>5</sup> operations. Min. (at rated coil voltage)
Weight		abt. 10grs.

## 9. REFERENCE DATA

### Coil Temperature Rise

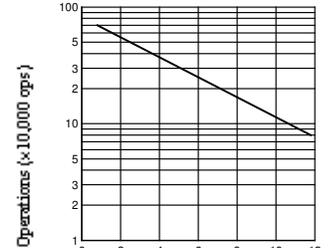


### Operation Time



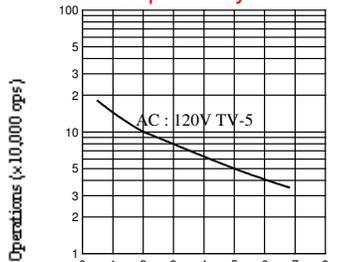
### Life Expectancy

AC120V/DC24V  $\cos\Phi=1$



### Current of Load (A)

### Life Expectancy



### Current of Load (A)

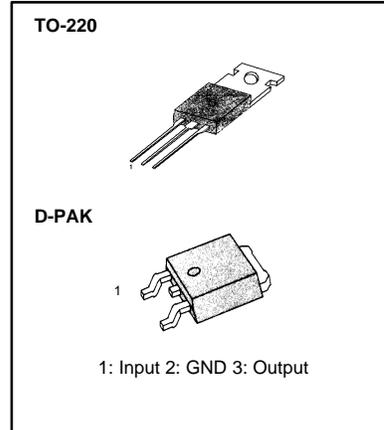
# LM78XX (KA78XX, MC78XX) FIXED VOLTAGE REGULATOR (POSITIVE)

## 3-TERMINAL 1A POSITIVE VOLTAGE REGULATORS

The LM78XX series of three-terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

## FEATURES

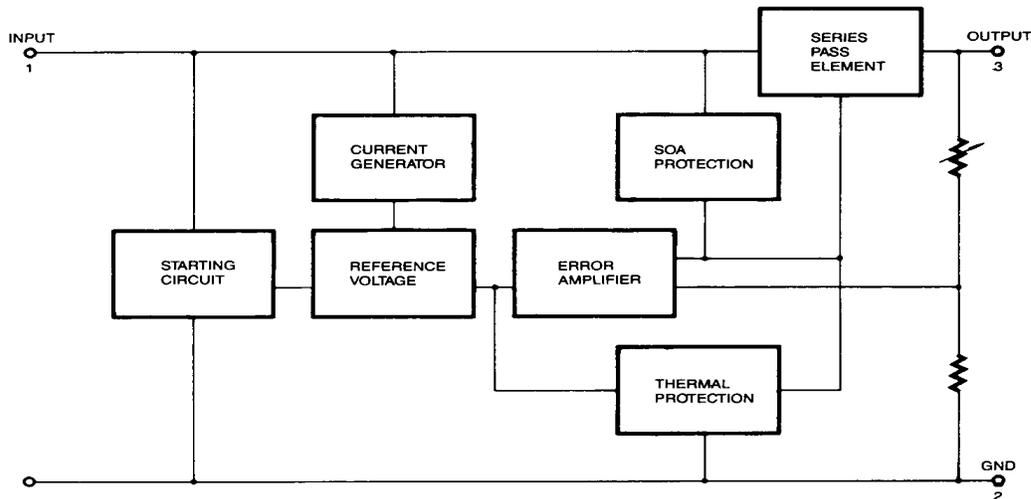
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 11, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor SOA Protection



## ORDERING INFORMATION

Device	Output Voltage Tolerance	Package	Operating Temperature
KA78XXCT	± 4%	TO-220	0 ~ +125 °C
KA78XXAT	± 2%		-40 ~ +125 °C
KA78XXIT	± 4%		
KA78XXR	± 2%	D-PAK	0 ~ +125 °C
KA78XXAR	± 2%		-40 ~ +125 °C
KA78XXIR	± 4%		

## BLOCK DIAGRAM



# LM78XX (KA78XX, MC78XX) FIXED VOLTAGE REGULATOR (POSITIVE)

## ABSOLUTE MAXIMUM RATINGS (T<sub>A</sub> = +25°C, unless otherwise specified)

Characteristic	Symbol	Value	Unit
Input Voltage (for V <sub>O</sub> = 5V to 18V) (for V <sub>O</sub> = 24V)	V <sub>I</sub>	35	V
	V <sub>I</sub>	40	V
Thermal Resistance Junction-Cases	R <sub>θJC</sub>	5	°C/W
Thermal Resistance Junction-Air	R <sub>θJA</sub>	65	°C/W
Operating Temperature Range KA78XX/A/R/RA KA78XXI/RI	T <sub>OPR</sub>	0 ~ +125	°C
		-40 ~ +125	°C
Storage Temperature Range	T <sub>STG</sub>	-65 ~ +150	°C

## LM7805/I/R/RI ELECTRICAL CHARACTERISTICS

(Refer to test circuit, T<sub>MIN</sub> < T<sub>J</sub> < T<sub>MAX</sub>, I<sub>O</sub> = 500mA, V<sub>I</sub> = 10V, C<sub>I</sub> = 0.33μF, C<sub>O</sub> = 0.1μF, unless otherwise specified)

Characteristic	Symbol	Test Conditions	LM7805I			LM7805			Unit
			Min	Typ	Max	Min	Typ	Max	
Output Voltage	V <sub>O</sub>	T <sub>J</sub> = +25°C	4.8	5.0	5.2	4.8	5.0	5.2	V
		5.0mA ≤ I <sub>O</sub> ≤ 1.0A, P <sub>O</sub> ≤ 15W V <sub>I</sub> = 7V to 20V V <sub>I</sub> = 8V to 20V	4.75	5.0	5.25	4.75	5.0	5.25	
Line Regulation	ΔV <sub>O</sub>	T <sub>J</sub> = +25°C	V <sub>O</sub> = 7V to 25V	4.0	100	4.0	100	mV	
			V <sub>I</sub> = 8V to 12V	1.6	50	1.6	50		
Load Regulation	ΔV <sub>O</sub>	T <sub>J</sub> = +25°C	I <sub>O</sub> = 5.0mA to 1.5A	9	100	9	100	mV	
			I <sub>O</sub> = 250mA to 750mA	4	50	4	50		
Quiescent Current	I <sub>Q</sub>	T <sub>J</sub> = +25°C	5.0	8	5.0	8	mA		
Quiescent Current Change	ΔI <sub>Q</sub>	I <sub>O</sub> = 5mA to 1.0A	0.03	0.5	0.03	0.5	mA		
		V <sub>I</sub> = 7V to 25V			0.3	1.3			
		V <sub>I</sub> = 8V to 25V	0.3	1.3					
Output Voltage Drift	ΔV <sub>O</sub> /ΔT	I <sub>O</sub> = 5mA	-0.8		-0.8		mV/°C		
Output Noise Voltage	V <sub>N</sub>	f = 10Hz to 100KHz, T <sub>A</sub> = +25°C	42		42		μV/V <sub>O</sub>		
Ripple Rejection	RR	f = 120Hz V <sub>O</sub> = 8 to 18V	62	73	62	73	dB		
Dropout Voltage	V <sub>O</sub>	I <sub>O</sub> = 1A, T <sub>J</sub> = +25°C	2		2		V		
Output Resistance	R <sub>O</sub>	f = 1KHz	15		15		mΩ		
Short Circuit Current	I <sub>SC</sub>	V <sub>I</sub> = 35V, T <sub>A</sub> = +25°C	230		230		mA		
Peak Current	I <sub>PK</sub>	T <sub>J</sub> = +25°C	2.2		2.2		A		

\* T<sub>MIN</sub> < T<sub>J</sub> < T<sub>MAX</sub>

LM78XXI/RI: T<sub>MIN</sub> = -40°C, T<sub>MAX</sub> = +125°C

LM78XX/R: T<sub>MIN</sub> = 0°C, T<sub>MAX</sub> = +125°C

\* Load and line regulation are specified at constant junction temperature. Changes in V<sub>O</sub> due to heating effects must be taken into account separately. Pulse testing with low duty is used.



# 15SQ045

## SCHOTTKY BARRIER RECTIFIER

Reverse Voltage - 45 Volts    Forward Current - 15.0 Ampere

### FEATURES

- The plastic package carries Underwriters Laboratory Flammability Classification 94V-0
- Construction utilizes void-free molded plastic technique
- Low reverse leakage
- High forward surge current capability
- High temperature soldering guaranteed:  
250°C/10 seconds, 0.375" (9.5mm) lead length, 5 lbs. (2.3kg) tension

### MECHANICAL DATA

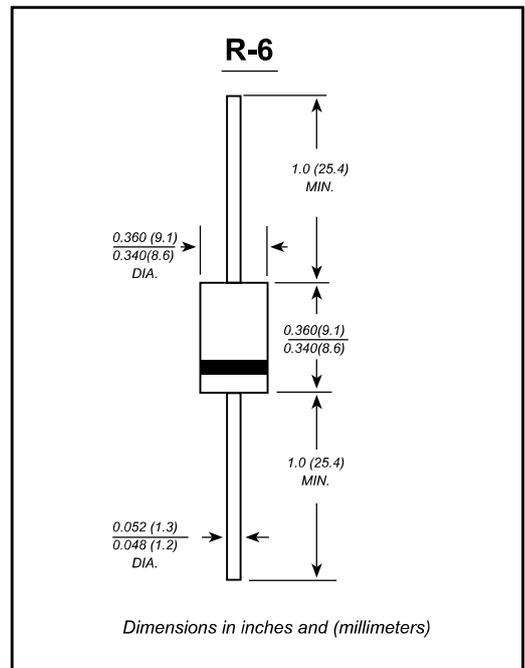
**Case:** R-6 molded plastic body

**Terminals:** Plated axial leads, solderable per MIL-STD-750, Method 2026

**Polarity:** Color band denotes cathode end

**Mounting Position:** Any

**Weight:** 0.072 ounce, 2.05 grams



### MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS

Ratings at 25°C ambient temperature unless otherwise specified.

Single phase half-wave 60Hz, resistive or inductive load, for capacitive load current derate by 20%.

	Symbols	15SQ045	Units
Maximum repetitive peak reverse voltage	V <sub>RRM</sub>	45	Volts
Maximum RMS voltage	V <sub>RMS</sub>	32	Volts
Maximum DC blocking voltage	V <sub>DC</sub>	45	Volts
Maximum average forward rectified current 0.375"(9.5mm) lead length(see fig. 1)	I <sub(av)< sub=""></sub(av)<>	15.0	Amps
Peak forward surge current 8.3ms single half sine-wave superimposed on rated load (JEDEC method at rated T <sub>J</sub> )	I <sub>FSM</sub>	150.0	Amps
Maximum instantaneous forward voltage at 15.0 A(Note 1 )	V <sub>F</sub>	0.55	Volts
Maximum instantaneous reverse current at rated DC blocking voltage(Note 1)	I <sub>R</sub>	T <sub>A</sub> = 25°C	0.5
		T <sub>A</sub> = 100°C	50
Typical junction capacitance(Note 3)	C <sub>J</sub>	400	pF
Typical thermal resistance (Note 2)	R <sub>θJC</sub>	3.0	°C/W
Operating junction temperature range	T <sub>J</sub>	-65 to +200	°C
Storage temperature range	T <sub>STG</sub>	-65 to +200	°C

Notes: 1. Pulse test: 300μs pulse width, 1% duty cycle

2. Thermal resistance from junction to case

3. Measured at 1MHz and reverse voltage of 4.0 volts

```
//CÓDIGO IMPLEMENTADO EN EL "MAESTRO"
```

```
//Pines de ENTRADA/SALIDA
```

```
//D13 - Clock (MCP3201 -> 7)
```

```
//D12 - Dout (MCP3201 -> 6)
```

```
//D10- CS enable (MCP3201 -> 5 - 5) compartido con esclavo
```

```
//D09 - Tx para modulo Bluetooth
```

```
//D08 - Rx para modulo Bluetooth
```

```
//D06 - Puerto Rx desde esclavo
```

```
//D05 - Señal de relé
```

```
//D04 - 2º Maestro puede recibir datos desde esclavo
```

```
//D03 - 1º Esclavo ha muestreado
```

```
//D02 - Activación muestreo
```

```
//D01 - Tx para modulo USB
```

```
//D00 - Rx para modulo USB
```

```
#include <SPI.h>
```

```
#include <SoftwareSerial.h>
```

```
const int SPin = 10; // Pin de habilitación
```

```
const int limitPtos = 250; // Límite de puntos a representar
```

```
uint16_t val = 0;
```

```
int nDatosLec = 0;
```

```
bool muestreo = 0;
```

```
bool finMuestr = 0; //indica si el muestreo ha finalizado para comenzar a enviar
```

```
bool leerSerial = 0;
```

```
bool printVec = 0;
```

```
byte readSerial = 0;
```

```
int tiempos[] = {2500, 1000, 500, 250, 170, 80, 40, 20, 10}; //vector de tiempos de espera para muestreo
```

```
uint16_t vecVoltaje[limitPtos]; // Vector que guarda los puntos de VOLATAJE
```

```
uint16_t vecCorriente[limitPtos]; // Vector que guarda los puntos de CORRIENTE
```

```
byte vecCorrByte[limitPtos*2];
```

```
String rec = " ";
```

```
//Definición de un nuevos puertos seriales
```

```
SoftwareSerial EslaveSerial(6, 7); // Rx, Tx. Comunicación con esclavo
```

```
SoftwareSerial BlueSerial(8, 9); // Rx, Tx. Puerto serial del Bluetooth
```

```

void setup() {
  Serial.begin(9600); //inicializa puerto estándar
  EslaveSerial.begin(9600);
  BlueSerial.begin(9600);
  SPI.begin();
  SPI.beginTransaction(SPISettings(500000, MSBFIRST, SPI_MODE0)); //125000
  pinMode(SPin, OUTPUT);
  digitalWrite(SPin, HIGH);
  digitalWrite(2, LOW);
  digitalWrite(5, LOW);

  pinMode(2, OUTPUT);
  pinMode(3, INPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT); //relé
}

```

```

void loop() {
  if ((Serial.available() > 0) && (muestreo == 0)){
    uint8_t recibe = (char)Serial.read();
    digitalWrite(2, HIGH);
    funInterrup(tiempos[recibe - 1]);
  }

  if((finMuestr == 1) && (digitalRead(3) == HIGH)){ //Final Muestreo Esclavo
    digitalWrite(4, HIGH);
    finMuestr = 0;
    leerSerial = 1;
  }
  if(leerSerial == 1){
    recibeVecEscl();
  }
  if(printVec == 1){
    representaVec();
  }
}

```

```
}
```

```
void funInterrup(int tiempo){  
    finMuestr = printVec = leerSerial = nDatosLec = 0;  
    muestreo = 1;  
    digitalWrite(5, HIGH); //activa relé  
    delay(4.5);  
    for(int i = 0; i < limitPtos; i++){  
        digitalWrite(SPin, LOW);  
        vecVoltaje[i] = ReadADC();  
        delayMicroseconds(20);  
        digitalWrite(SPin, HIGH);  
        delayMicroseconds(tiempo);  
    }  
    digitalWrite(5, LOW); //Desactiva relé  
    finMuestr = 1;  
}
```

```
//Función de lectura del ADC
```

```
uint16_t ReadADC() {  
    val = (SPI.transfer(0x00) << 8);  
    val |= SPI.transfer(0x00);  
    val = (val >> 1) & 0xFFF;  
    return val;  
}
```

```
//Función para recibir datos del Esclavo
```

```
void recibeVecEscl(){  
    while(EslaveSerial.available() > 0){  
        vecCorrByte[nDatosLec] = EsclaveSerial.read();  
        readSerial = 0;  
        nDatosLec++;  
    }  
    if(nDatosLec == (limitPtos*2)){  
        nDatosLec = leerSerial = 0;  
        digitalWrite(4, LOW);  
    }  
}
```

```
    preparaVec();  
}  
digitalWrite(2, LOW);  
}
```

```
void preparaVec(){  
    for(int i = 0; i < limitPtos; i++){  
        uint16_t byteH = vecCorrByte[i];  
        uint16_t byteL = vecCorrByte[i + limitPtos];  
        vecCorriente[i] = ((byteH << 8) | (byteL));  
        if((vecCorriente[i] < 0) || (vecCorriente[i] > 4095)){  
            vecCorriente[i] = 1;  
        }  
    }  
    printVec = 1;  
}
```

//Función para representar por pantalla los vectores

```
void representaVec(){  
    for(int i = 0; i < limitPtos; i++){  
        int valVolt = (vecVoltaje[i]);  
        Serial.println(valVolt);  
    }  
    for(int j = 0; j < limitPtos; j++){  
        int valCorr = (vecCorriente[j]);  
        Serial.println(valCorr);  
    }  
    printVec = 0;  
    muestreo = 0;  
}
```

```
//CÓDIGO IMPLEMENTADO EN EL "ESCLAVO"
```

```
//PINES DE ENTRADA/SALIDA
```

```
//D13 - Clock (7) desde maestro
```

```
//D12 - Dout (6)
```

```
//D05 - CS enable (5) desde D10_M (maestro)
```

```
//D04 - 2º Maestro puede recibir desde esclavo
```

```
//D03 - 1º Esclavo ha muestreado
```

```
//D02 - Activación del muestreo
```

```
//D01 - Tx
```

```
#include <SPI.h>
```

```
const int SPin = 5; // Pin de habilitación
```

```
const int limitPtos = 250; // Límite de puntos a representar
```

```
bool finMuestr = 0; // comienzo del envio
```

```
int i = 0;
```

```
int est = 0;
```

```
int mue = 0;
```

```
uint16_t val = 0;
```

```
uint16_t vecCorriente[limitPtos]; // Vector que guarda los puntos
```

```
byte vecCorrByteAlto[limitPtos];
```

```
byte vecCorrByteBajo[limitPtos];
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  SPI.begin();
```

```
  SPI.beginTransaction(SPISettings(500000, MSBFIRST, SPI_MODE0));
```

```
  pinMode(2, INPUT); //pin que habilita el muestreo
```

```
  pinMode(3, OUTPUT); //pin de habilitación cuando esclavo acabó muestreo
```

```
  pinMode(4, INPUT); //pin de habilitación para que esclavo mande los dato
```

```
  pinMode(5, INPUT); //Pin de entrada desde maestro
```

```
  digitalWrite(3, LOW);
```

```
}
```

```
void loop() {
```

```

if(digitalRead(2) == HIGH){
  if((est == 0) && (digitalRead(5) == LOW)){
    mue = 1;
    est = 1;
  }
  if((est == 1) && (digitalRead(5) == HIGH)){
    mue = 0;
    est = 0;
  }
  while(mue == 1){
    val = (SPI.transfer(0x00) << 8); // MSB
    val |= SPI.transfer(0x00); // LSB
    val = (val >> 1) & 0xFFF; // Remove null bit
    vecCorriente[i] = val;
    i++;
    mue = 0;
    if(i == limitPtos){
      lectura();
    }
  }
}
else{
  i = 0;
  mue = 0;
  digitalWrite(3, LOW);
  for(int i = 0; i < limitPtos; i++){
    vecCorriente[i] = 0;
  }
}
if((finMuestr == 1) && (digitalRead(4) == HIGH)){
  delay(1000);
  enviaVecMaestro();
  finMuestr = 0;
}
}

void lectura(){
  i = 0;

```

```
digitalWrite(3, HIGH);
for(int i = 0; i < limitPtos; i++){
    vecCorrByteAlto[i] = (vecCorriente[i] >> 8);
    vecCorrByteBajo[i] = (vecCorriente[i]);
}
finMuestr = 1;
}
```

//Función para enviar el vector al Maestro

```
void enviaVecMaestro(){
    for(int i = 0; i < limitPtos; i++){
        Serial.write(vecCorrByteAlto[i]);
    }
    for(int j = 0; j < limitPtos; j++){
        Serial.write(vecCorrByteBajo[j]);
    }
    digitalWrite(3, LOW);
}
```

## #CÓDIGO IMPLEMENTADO PARA LA CONFIGURACIÓN DE LA INTERFAZ

```
from PyQt5 import QtWidgets, uic
import pyqtgraph as pg
import sys
import muestreo_arduino as mArd
import pandas as pd
from datetime import datetime

vecVolt = []
vecCurr = []
vecPower = []

class MainWindow(QtWidgets.QMainWindow):
    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)

        uic.loadUi('mainwindow.ui', self)
        global curva_IV, curva_Power

        self.stackedWidget.setCurrentIndex(0) #mostrar primera página
        # botones de trazar
        self.button1_1.clicked.connect(self.muestreo)
        self.button1_2.clicked.connect(self.muestreo)

        # botones de mostrar/ocultar tabla
        self.button2_1.clicked.connect(lambda:self.mostra_tabla(vecVolt, vecCurr, vecPower))
        self.button2_2.clicked.connect(lambda:self.stackedWidget.setCurrentIndex(1))
        # botones de guardar
        self.button3_1.clicked.connect(lambda:self.save_data(vecVolt, vecCurr, vecPower, 'xlsx'))
        self.button3_2.clicked.connect(lambda:self.save_data(vecVolt, vecCurr, vecPower, 'xlsx'))
        self.button4.clicked.connect(lambda:self.save_data(vecVolt, vecCurr, vecPower, 'csv'))

        #graficas
        pg.mkQApp()
        self.grafica_iv.show()
        curva_IV = self.grafica_iv.plotItem
        curva_IV.setLabel('left', 'Current [mA]', color='#00d100')
```

```

curva_IV.setLabel('bottom', 'Voltage [V]', color='#000000')

#create new viewBox
curva_Power = pg.ViewBox()
curva_IV.showAxis('right')
curva_IV.scene().addItem(curva_Power)
curva_IV.getAxis('right').linkToView(curva_Power)
curva_Power.setXLink(curva_IV)
curva_IV.getAxis('right').setLabel('Power [W]', color='#0000ff')

#checkBox de Curva de Potencia
self.checkBox1.setChecked(True)
self.checkBox1.stateChanged.connect(lambda:self.checkBoxState('checkBox1', False, vecVolt,
vecPower))
self.checkBox1.setChecked(False)
self.checkBox1.stateChanged.connect(lambda:self.checkBoxState('checkBox1', True, vecVolt,
vecPower))

def muestreo(self):
    global curva_IV, curva_Power, Voc, Isc, Pmax, Vmp, Imp
    longiVec = 0
    self.label_Traz.setText('Sampling...')
    self.label_ArchG_1.setText('Sampling...')
    self.label_Isc.setText("")
    self.label_Voc.setText("")
    self.label_Pmax.setText("")
    self.label_Traz.repaint()
    self.label_ArchG_1.repaint()
    self.label_Isc.repaint()
    self.label_Voc.repaint()
    self.label_Pmax.repaint()
    self.grafica_iv.clear()
    curva_IV.clear()
    curva_Power.clear()
    self.grafica_iv.repaint()

try:
    vecDatos = mArd.muestreo_datos(1)

```

```

    longiVec = int(len(vecDatos)/2)
except:
    self.label_Traz.setText('Failed to connect with device')
    self.label_ArchG_1.setText('Failed to connect with device')
    return 0

self.stackedWidget.setCurrentIndex(1)

#Vectores de Voltaje y corriente
vecVolt.clear()
vecCurr.clear()
vecPower.clear()
for i in range(0, longiVec):
    vecVolt.append(vecDatos[i])
    vecCurr.append(vecDatos[i+longiVec])

# Vector de Potencia
for i in range (0, longiVec):
    vecPower.append(round((vecVolt[i]*vecCurr[i]/1000), 2))

self.label_Traz.setText(' ')
self.label_ArchG_1.setText(' ')
self.label_ArchG_2.setText(' ')

self.grafica_iv.setBackground('w') #w: white
self.grafica_iv.setBackground('w')
self.grafica_iv.showButtons()
self.grafica_iv.showGrid(x=True, y=True) #cuadrícula
self.checkBox1.setChecked(False)

# Valores de Isc, Voc y Pmax
Isc = vecCurr[0]
Voc = vecVolt[longiVec - 1]
Pmax = max(vecPower)
posPmax = vecPower.index(max(vecPower))
Vmp = vecVolt[posPmax]
Imp = vecCurr[posPmax]

```

```

self.label_Isc.setText(str(Isc))
self.label_Voc.setText(str(Voc))
self.label_Pmax.setText(str(Pmax))
self.label_Imp.setText(str(Imp))
self.label_Vmp.setText(str(Vmp))

#PLOT CURVA I-V
curva_Power.setGeometry(curva_IV.vb.sceneBoundingRect())
curva_Power.linkedViewChanged(curva_IV.vb, curva_Power.XAxis)
curva_IV.plot(vecVolt, vecCurr, pen = pg.mkPen('g'), symbol='o', symbolSize=5, symbolBrush='g')

def mostrarGraf_Power(self, check, vecVolt, vecPower):
    global curva_IV, curva_Power
    if(check == True):
        curva_Power.addItem(pg.PlotCurveItem(vecVolt, vecPower, pen = pg.mkPen('b')))
        self.grafica_iv.showAxis('right', True)

    else:
        curva_Power.clear()
        self.grafica_iv.showAxis('right', False)

def checkBoxState(self, checkBox, state, vecVolt, vecPower):
    if (checkBox == 'checkBox1'):
        if (self.checkBox1.isChecked() == state == True):
            self.mostrarGraf_Power(True, vecVolt, vecPower)
        if (self.checkBox1.isChecked() == state == False):
            self.mostrarGraf_Power(False, vecVolt, vecPower)
            self.checkBox1.setChecked(False)

def muestra_tabla(self, vecVolt, vecCurr, vecPower):
    self.stackedWidget.setCurrentIndex(2)
    n_row = len(vecVolt) #numero de filas
    self.tabla_datos.setRowCount(n_row) #numero de filas
    #rellenar tabla
    row = 0

```

```

for row in range(0, n_row):
    self.tabla_datos.setItem(row, 0, QtWidgets.QTableWidgetItem(str(vecVolt[row])))
    self.tabla_datos.setItem(row, 1, QtWidgets.QTableWidgetItem(str(vecCurr[row])))
    self.tabla_datos.setItem(row, 2, QtWidgets.QTableWidgetItem(str(vecPower[row])))

def save_data(self, vecVolt, vecCurr, vecPower, fileType):
    global Voc, Isc, Vmp, Imp, Pmax
    self.label_ArchG_1.setText("")
    self.label_ArchG_2.setText("")

    data = {'Voltage[V]': vecVolt, 'Current[mA]': vecCurr, 'Power[W]': vecPower}
    df = pd.DataFrame(data, columns = ['Voltage[V]', 'Current[mA]', 'Power[W]', 'Voc[V]', 'Isc[mA]',
    'Pmax[W]', 'Vmp[V]', 'Imp[mA]'])
    df.loc[0, 'Voc[V]'] = Voc
    df.loc[0, 'Isc[mA]'] = Isc
    df.loc[0, 'Pmax[W]'] = Pmax
    df.loc[0, 'Vmp[V]'] = Vmp
    df.loc[0, 'Imp[mA]'] = Imp

    dt = datetime.now() # Fecha y hora actual
    timeStr = '_' + str(dt.day) + '_' + str(dt.month) + '_' + str(dt.year) + '_' + str(dt.hour) + '_' +
    str(dt.minute) + '_' + str(dt.second) + '_'

    if (fileType == 'xlsx'):
        nameXlsx = 'PVcurve' + timeStr + '.xlsx'
        df.to_excel(nameXlsx)
        self.label_ArchG_1.setText('Saved Excel file')
        self.label_ArchG_2.setText('Saved Excel file')
    elif (fileType == 'csv'):
        self.label_ArchG_2.setText('Saved .csv file')
        nameCsv = 'PVcurve' + timeStr + '.csv'
        df.to_csv(nameCsv, index=False)

def main():
    app = QtWidgets.QApplication(sys.argv)
    main = MainWindow()

```

```
main.show()  
sys.exit(app.exec_())
```

```
if __name__ == '__main__':  
    main()
```

# CÓDIGO PARA CONEXIÓN Y ALMACENAJE DE DATOS DESDE EL MICROCONTROLADOR

```
import serial, time
```

```
limitPtos = 250
```

```
Vref = 4.99 # voltaje de ref ADC
```

```
refBytes = 4095 # Bytes
```

```
Vparti = 23.6 #  $(R1+R2)/R2 = (150.9k + 6.71k)/6.71k$ 
```

```
Gain = 68 #  $1+R5/R6 = 1 + 67k/0.994k$ 
```

```
Rshunt = 10/0.075 # 10A/75mV
```

```
def muestreo_datos(loop):
```

```
    arduino = serial.Serial('COM5', 9600)
```

```
    while(loop > 0):
```

```
        vecVolt = []
```

```
        vecCurr = []
```

```
        vecVoltFiltr = []
```

```
        vecCurrFiltr = []
```

```
        vecDatos = []
```

```
        val = 0
```

```
        numDatosFiltr = 0
```

```
        ceroInt = 0
```

```
        time.sleep(2)
```

```
        arduino.write(str.encode(chr(loop)))
```

```
    # VOLTAJE: Bucle para rellenar
```

```
    for i in range(0, (limitPtos)):
```

```
        val = arduino.readline() #lectura desde serial
```

```
        valStr = str(val) #paso a string de val
```

```
        lenVal = len(valStr) #longitud del string
```

```
        startStr = 2
```

```
        endStr = (lenVal - 5)
```

```
        men = valStr[startStr: endStr] #corte del string para quedarse solo con n°
```

```
        valInt = float(men) #paso a float del string
```

```
        valInt = valInt*(Vref/refBytes)*Vparti
```

```
        vecVolt.append(round(valInt, 2)) #construccion del vector de voltaje
```

```
    # CORRIENTE: Bucle para rellenar el vector
```

```

for i in range(0, (limitPtos)):
    val = arduino.readline() #lectura desde serial
    valStr = str(val) #paso a string de val
    lenVal = len(valStr) #longitud del string
    startStr = 2
    endStr = (lenVal - 5)
    men = valStr[startStr: endStr] #corte del string para quedarse solo con n°
    valInt = float(men) #paso a float del string
    valInt = valInt*(Vref/refBytes)*(1/Gain)*(Rshunt)*1000 #valor en mA
    vecCurr.append(round(valInt, 0)) #construccion del vector de voltaje

# Primer FILTRO DE DATOS
for i in range(0, limitPtos):
    if((i <= 15) & (vecVolt[i] < 10) & (vecCurr[i] > 10)):
        vecVoltFiltr.append(vecVolt[i])
        vecCurrFiltr.append(vecCurr[i])

    if((i > 16) & (vecCurr[i] >= 0) & (ceroInt < 5)):
        vecVoltFiltr.append(vecVolt[i])
        vecCurrFiltr.append(vecCurr[i])
        if vecCurr[i] == 0:
            ceroInt = ceroInt + 1

#Segundo filtro de datos
numDatosFiltr = len(vecVoltFiltr)
if (numDatosFiltr > 120):
    #ENVIAR DATOS HACIA INTERFAZ
    for i in range(0, numDatosFiltr):
        vecDatos.append(vecVoltFiltr[i])

    for i in range(0, numDatosFiltr):
        vecDatos.append(vecCurrFiltr[i])

loop = 0
return vecDatos
arduino.close()

```

```
else:  
    #VOLVER A MUESTREAR  
    if (loop < 10):  
        loop = loop + 1  
    else:  
        loop = 0  
        return 0  
        arduino.close()
```