Universidad de La Laguna

TRABAJO FIN DE GRADO

Escuela Superior de Ingeniería y Tecnología

Grado en Ingeniería Electrónica Industrial y Automática

DESARROLLO DE PROTOCOLO INTERFAZ CEREBRO-ORDENADOR (BRAIN-COMPUTER INTERFACE) PARA EL CONTROL DE DISPOSITIVOS

Curso académico 2020/2021

Autor: Sergio Pérez Widera

Tutores académicos: Kevin Martín Chinea y José Francisco Gómez González

Desarrollo de protocolo interfaz cerebro-ordenador (Brain-Computer Interface) para el control de dispositivos

Sergio Pérez Widera

Resumen

El objetivo principal del presente TFG consiste en desarrollar un protocolo de comunicación entre una interfaz gráfica de usuario (GUI) y una interfaz cerebro-ordenador (BCI).

El sistema GUI tiene que ser diseñado de tal forma que debe ser capaz de recibir e interpretar la información recibida del BCI, y posteriormente reaccionar de una forma u otra en función del comando recibido.

Este TFG está dividido en tres partes. La primera de ellas consiste en la implementación de la GUI, en la cual se debe diseñar y programar dicha GUI. La segunda parte consiste en el estudio de los datos extraídos de las señales obtenidas mediante la realización de un electroencefalograma (EEG). Por último, la tercera parte consiste en implementar las comunicaciones entre la GUI y el sistema BCI.

El resultado final de este TFG es el diseño e implementación del sistema de interacción de la GUI en función de los comandos recibidos por parte del BCI, en función de la clasificación de los datos que se extraen del EEG.

Palabras clave: electroencefalografía, interfaz gráfica de usuario, interfaz cerebro-ordenador.

Abstract

The main objective of this TFG is to develop a communication protocol between a graphical user interface (GUI) and a brain-computer interface (BCI).

The GUI system has to be designed in such a way that it must be able to receive and interpret the information received from the BCI, and subsequently react in one way or another depending on the command received.

This project is divided in three parts. The first one consists in the implementation of the GUI, in which it is necessary to design and develop this GUI. Then, the second part consists in the study of the extracted data of some electroencephalography (EEG) records. And for the last, the third part consists in the implementation of the communications between the GUI designed in the first part and the BCI.

As the final result of this project, is expected to, by the classification of the extracted data of the EEG with a streaming code, the interaction of the GUI according to the received commands by the BCI.

Keywords: electroencephalography, graphical user interface, brain-computer interface.

Índice general

Índice de figuras

Índice de tablas

Introducción

El primer indicio de lo que ahora se conoce como BCI data de la década de 1920, en la que el neurólogo alemán Hans Berger logró registrar actividad eléctrica al colocar electrodos en una trepanación craneana y conectarlos a un galvanómetro. A principios de la década de 1960, la científica Thelma Estrin sugirió el sistema de computación digital electroencefalográfico y dio órdenes para llevar a cabo el sistema BCI que permitiría trasmitir las señales cerebrales a una computadora. En la década de 1970, Jacques Vidal publicó un artículo en el cual sugiere diferentes técnicas para la creación de una comunicación directa cerebro-ordenador (*El mayor logro de la historia de la neurociencia: las interfaces cerebro ordenador*, 2020).

Las interfaces cerebro-ordenador (BCI) tienen como finalidad establecer comunicación de manera directa entre el cerebro y un ordenador. Esto es gracias a la actividad eléctrica que se produce cuando las neuronas cerebrales se comunican entre sí, que se detecta y se decodifica en comandos que se envían a un dispositivo externo el cual realiza las acciones pertinentes.

Algunas de las diversas aplicaciones que tienen estos BCI son:

- Sillas de ruedas o prótesis, en las cuales se facilita tanto el control como la comunicación con estas.
- Videojuegos de realidad virtual basados en estos BCI.
- El control de un ordenador y la navegación por internet.
- Neuromarketing, estudiando como determinada publicidad afecta a sus consumidores.

El funcionamiento del BCI consiste, tal y como se puede observar en la figura 1, en adquirir las señales cerebrales mediante determinadas metodologías, tales como la realización de un electroencefalograma (EEG) o una electrocorticografía (ECoG) entre otros métodos, procesar las señales adquiridas para obtener sus características principales y transformarlas en comandos. Posteriormente, se envían al dispositivo conectado al BCI mediante los cuales el dispositivo reaccionará de una forma u otra.

Figura 1. Esquema de funcionamiento de un BCI (*Arquitectura BCI*, 2021).

En la actualidad, para poder adquirir las señales cerebrales se hace uso de un electroencefalograma (EEG), una prueba médica consistente en registrar la actividad eléctrica del cerebro.

Para ello, se coloca sobre la cabeza del usuario sobre el cual se realiza el EEG un dispositivo que contiene un número determinado de sensores metálicos, denominados electrodos, y sobre los cuales se aplica un gel capilar que permite que los impulsos eléctricos conduzcan mejor.

Cada electrodo envía una señal a una máquina llamada electroencefalógrafo, que muestra la fluctuación rítmica de la actividad eléctrica del cerebro (ondas cerebrales) visualmente como una línea ondulante (*Electroencefalograma*, s.f.).

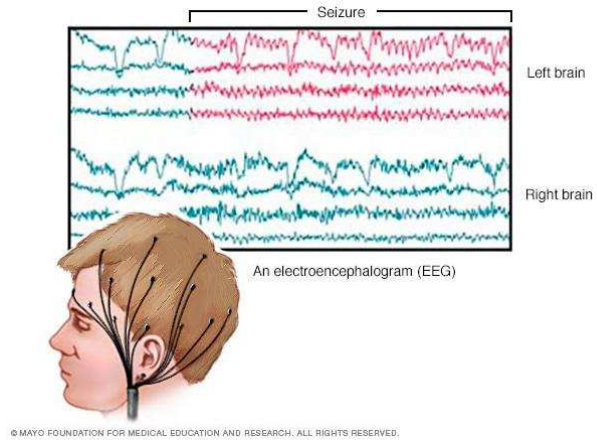


Figura 2. Ejemplo de un EEG (*EEG (electroencefalograma)*, s.f.).

El objetivo de este TFG es desarrollar un sistema BCI que decodifique los datos recibidos por las señales de un EEG en forma de comandos y envíe estos a una interfaz gráfica de usuario (GUI), que reaccionará de una forma u otra en función del comando enviado por el BCI (*EEG (electroencefalograma)*, s.f.).

Es necesario también definir previamente 2 conceptos que serán de gran ayuda para comprender el trabajo que se está realizando: la interfaz gráfica de usuario (GUI) y las ondas cerebrales.

Una interfaz gráfica de usuario (GUI) es una herramienta informática que permite la comunicación entre una persona y el sistema operativo de una maquina u ordenador. Dicha GUI utiliza un conjunto de imágenes y objetos gráficos para así poder facilitar la comunicación entre las personas y el sistema operativo (Martínez, 2015).

Figura 3. Ejemplo de GUI perteneciente al sistema operativo iOS (*Graphical User Interface (GUI),* s.f.*).*

Las ondas cerebrales recogidas con un electroencefalograma muestran la actividad eléctrica producida por el cerebro, principalmente en su corteza cerebral. Tienen una amplitud muy baja y no siguen una forma sinusoidal regular. Dichas ondas se clasifican de la siguiente manera según sus compontes en frecuencia (*Onda cerebral*, 2019).

Tabla 1. Clasificación de los distintos tipos de ondas cerebrales.

Tipos de ondas	Rango de frecuencia
Ondas delta	1 a 3 Hz
Ondas theta	3,1 a 7,9 Hz
Ondas alpha	8 a 13 Hz
Ondas beta	14 a 29 Hz
Ondas gamma	30 a 100 Hz

Cada una de las distintas ondas cerebrales tiene asociado un significado, el cual se detalla a continuación *(¿Qué son las ondas cerebrales?,* 2019):

- Las ondas deltas predominan durante el sueño.
- Las ondas theta se producen cuando se está procesando información interna y el individuo se encuentra aislado del mundo exterior.
- Las ondas alpha aparecen cuando el sistema nervioso central de cada individuo se encuentra en calma, pero despierto y prestando atención.
- Las ondas beta predominan cuando el individuo se encuentra en periodo de vigilia.
- Las ondas gamma se relacionan con el proceso de información simultánea en distintos lugares del sistema nervioso central.

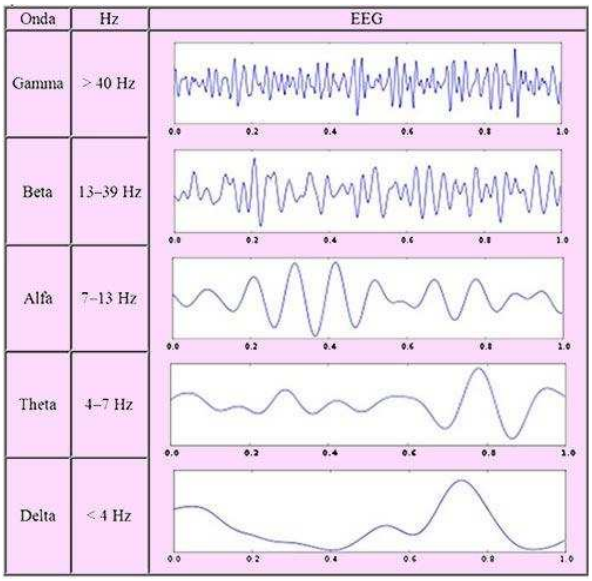


Figura 4. Imagen aproximada de la forma de los diferentes tipos de ondas cerebrales (*Tipos de ondas cerebrales*, s.f.).

Objetivos

A continuación, se enumeran y describen todas las tareas llevadas a cabo para la realización de este trabajo:

- Diseño de una GUI que realice unas determinadas acciones, y que permite interactuar con un teclado.
- Diseño de un BCI que clasifique la información extraída de diferentes registros de EEG.
- Desarrollo de mapas topográficos con la información de los registros de EEG para estudiar los diferentes estados de estimulación (ojos abiertos, ojos cerrados, ojo derecho cerrado y ojo izquierdo cerrado).
- Diseño e implementación del protocolo de comunicaciones entre la GUI y el BCI.

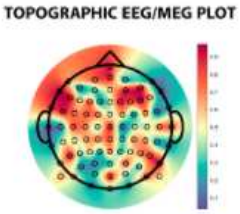
Materiales y métodos

Software utilizado

Para el desarrollo de este TFG se han empleado los diferentes software y herramientas que se citan a continuación junto con las tareas realizadas en cada uno:

Tabla 2. Softwares y herramientas utilizadas parala realización de este TFG.

Logotipo	Nombre	Tareas realizadas
	Unity 3D	Diseño y configuración de la GUI.
	Microsoft Visual Studio	Programación de la GUI en lenguaje C#.

	Matlab	Realización de los códigos correspondientes al estudio de las señales del EEG.
	Brainflow	Realización del preprocesamiento de los datos extraídos del EEG y obtención de las características de estos.
	Classification learner (herramientas de Matlab)	Utilizado para la obtención del modelo que clasifique las señales del EEG.
	Topographic EEG/MEG Plot	Utilizada para realizar los 4 mapas topográficos necesarios para observar si existe diferencia visual entre los protocolos estudiados.

Materiales

Para la realización del EEG y la posterior adquisición de señales cerebrales se ha hecho uso de un gorro de tela, con 16 electrodos, y una placa que es capaz de adquirir las señales en 16 canales distintos, correspondientes con los 16 electrodos del gorro de

tela. Tanto estos materiales como los cables que permiten la comunicación entre los electrodos y la placa son del fabricante OpenBCI.

Figura 5. Gorro de tela utilizado para realizar el EEG.

La placa utilizada para la adquisición de señales es el modelo Cyton + Daisy, que como se menciona anteriormente es capaz de registrar datos en un total de 16 canales distintos. La placa consta de dos módulos, el módulo Cyton, capaz de adquirir datos en un total de 8 canales y el módulo Daisy, que permite adquirir datos en otros 8 canales. Las características de cada uno de los módulos, junto con las del dongle, dispositivo que permite la comunicación entre la placa y el ordenador de manera directa, se detallan a continuación:

OpenBCI Cyton Module:

- 8 differential, high gain, low noise input channels
- Compatible with active and passive electrodes
- Texas Instruments ADS1299 ADC

- PIC32MX250F128B microcontroller w/chipKI bootloader (50MHz)
- RFduino™ Low Power Bluetooth™ radio
- 24-bit channel data resolution
- Programmable gain: 1, 2, 4, 6, 8, 12, 24
- 3.3V digital operating voltage
- +/-2.5V analog operating voltage
- ~3.3-12V input voltage
- LIS3DH accelerometer
- Micro SD card slot
- 5 GPIO pins, 3 of which can be Analog

OpenBCI Daisy Module:

- 8 differential, high gain, low noise input channels
- Compatible with active and passive electrodes
- Texas Instruments ADS1299 ADC
- 24-bit channel data resolution
- Programmable gain: 1, 2, 4, 6, 8, 12, 24
- 3.3V digital operating voltage
- +/-2.5V analog operating voltage
- Powered by OpenBCI board

OpenBCI Dongle:

- RFD22301 radio module from RFdigita
- FT231X USB-to-serial converter from FTDI

- Can upload code to the OpenBCI board or the dongle
- Fully broken out and pin-compatible w/ RFduino form factor

Figura 6. Placa utilizada para la adquisición de señales (*Cyton + Daisy*, s.f.).

Las hojas de datos (datasheet) necesarias de los componentes de la placa se sitúan en los anexos de esta memoria.

El ordenador utilizado tanto para la realización de todos los códigos que componen este TFG como para la realización de las pruebas de funcionamiento de estos posee las siguientes características:

- Procesador Intel Core i5 de 2 núcleos con una frecuencia básica de 2,5 GHz y una frecuencia turbo máxima de 3,10 GHz.
- Memoria RAM de 8 GB.
- Disco duro de 1 TB HDD.

El ordenador utilizado para realizar los registros de los distintos usuarios en un EEG y probar el sistema BCI entero son:

- Procesador Intel Core i5 de 2 núcleos con una frecuencia básica de 2,5 GHz y una frecuencia turbo máxima de 3,10 GHz.

- Memoria RAM de 8 GB.
- Disco duro de 256 GB SSD.

Metodología

Para obtener una visión general de cómo será el funcionamiento del sistema BCI desarrollado en este TFG, se adjunta en la figura 7 un diagrama de bloques en el que se expone como se relacionan cada una de las partes del sistema.

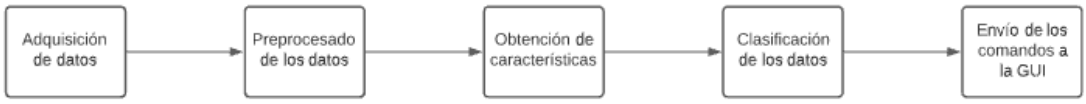


Figura 7. Diagrama de bloques en el que se expone el funcionamiento del sistema BCI.

Concretamente, el funcionamiento del sistema consiste en adquirir los datos procedentes de las señales del EEG, realizar el preprocesamiento de estos, obtener las características que proporcionan una mayor información sobre los datos obtenidos y clasificar estos en comandos para posteriormente enviar a la GUI la tarea a realizar, donde la GUI reaccionará de una forma u otra en función del comando recibido.

Diseño y programación de la GUI

Antes de llevar a cabo el estudio de las señales adquiridas, se realizará tanto el diseño como la programación de la GUI.

El objetivo de la realización de esta parte del TFG es, tal y como se menciona en el apartado de objetivos de esta memoria, el diseño de una GUI que realice unas determinadas acciones según una entrada proporcionada. En esta GUI se incluyen un

total de cuatro botones, todos ellos de dirección, de manera muy similar al panel de botones de dirección de un mando a distancia. La GUI está conformada por un botón que permite desplazarse hacia adelante, otro para retroceder y por último dos para poder desplazarse lateralmente, uno para desplazarse hacia la izquierda y otro para hacerlo hacia la derecha. Esta GUI está controlada mediante las señales cerebrales producidas al abrir o cerrar uno o ambos ojos, ya que al no mover ninguna parte más del cuerpo, los resultados obtenidos son más fiables.

Para ir observando el movimiento dentro de la GUI, un cursor recorre todos los botones que posee la interfaz y cambia de color estos estos cuando están seleccionados. El funcionamiento de la GUI viene definido por las acciones que se describen a continuación, todas ellas correspondientes a cerrar uno o ambos ojos.

Tabla 3. Controles de funcionamiento de la GUI.

Acción	Resultado
Cerrar el ojo derecho	El cursor se desplaza en sentido horario dentro de la GUI
Cerrar el ojo izquierdo	El cursor se desplaza en sentido antihorario dentro de la GUI
Cerrar ambos ojos	Se selecciona el botón que esté activo en ese momento

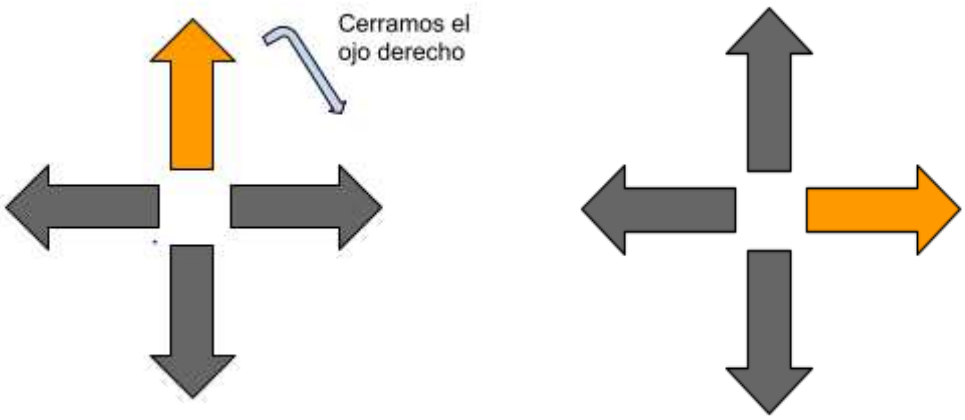


Figura 8. Ejemplo de funcionamiento de la GUI.

Para desarrollar la GUI se ha dividido en la tarea en dos partes:

- Diseño de la GUI.
- Programación de la GUI y configuración de los botones que la componen.

Tanto para el diseño de la GUI como para la configuración de los botones que la componen se ha hecho uso del software Unity, cuyo principal uso es el de desarrollo de videojuegos para diferentes plataformas, tales como PC o smartphone entre otros. El diseño de la interfaz, tal y como se describe con anterioridad, es bastante sencillo y consta de cuatro botones de dirección. Para la realización de la interfaz se creó primero un Canvas en Unity, es decir, la ventana gráfica en la cual posteriormente se añadieron los cuatro botones de dirección de la interfaz. Una vez creado tanto el Canvas como los cuatro botones, el resultado quedaría como en la figura 9.

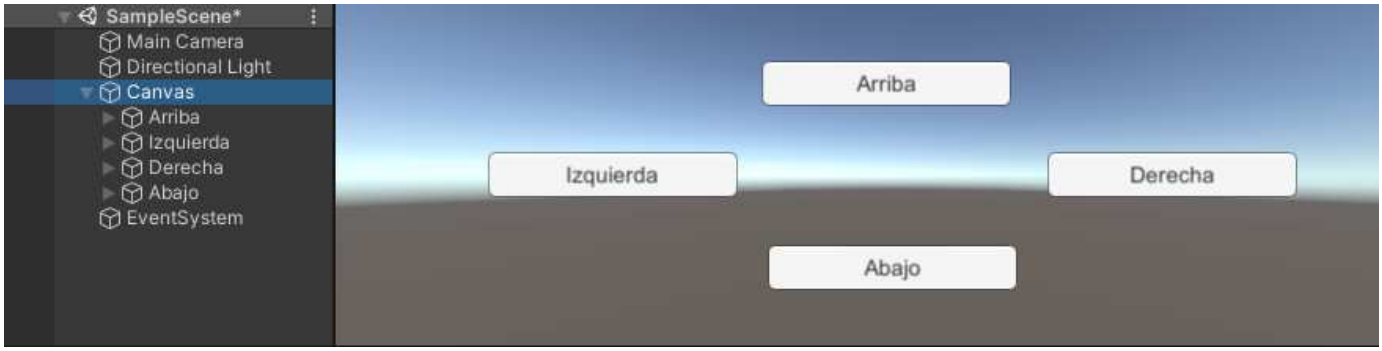


Figura 9. Diseño final de la GUI que se utilizará en este TFG.

Para llevar a cabo la programación de la interfaz se ha hecho uso del software Microsoft Visual Studio, en el cual los códigos desarrollados han sido realizados en lenguaje C#. Para poder comprobar su funcionamiento se realizaron dos pruebas, una de ellas clicando sobre cada uno de los botones y otra de ellas mediante la utilización del teclado.

La primera prueba tiene como objetivo conseguir que al clicar sobre cada uno de los botones que componen la interfaz se muestre por la consola un mensaje con el nombre del botón que se está pulsando en ese momento. El código utilizado para realizar esta prueba se denomina NewBehaviourScript.cs. Si se indaga en el código, se puede encontrar que se utiliza para poder mostrar por consola cada botón la misma función, que consiste, básicamente, en vincular el nombre de cada botón con su homónimo dentro de la GUI y mostrar por consola con la sentencia print más el nombre del botón a mostrar, obteniendo un total de 4 funciones, una por cada botón. Pueden consultarse los detalles en la figura 10.

```
public void Arriba()
{
    print("Arriba");
}

public void Abajo()
{
    print("Abajo");
}

public void Izquierda()
{
    print("Izquierda");
}

public void Derecha()
{
    print("Derecha");
}
```

Figura 10. Funciones utilizadas para mostrar por consola un mensaje con el nombre del botón seleccionado.

Estas funciones permiten que al clicar sobre cualquiera de los 4 botones que componen la GUI, el nombre de estos pueda ser mostrado por la consola.

Para configurar los botones de la interfaz en Unity únicamente es necesario añadir al Canvas el script NewBehaviourScript.cs. Añadido el script, en la figura 11 se observa el resultado obtenido.

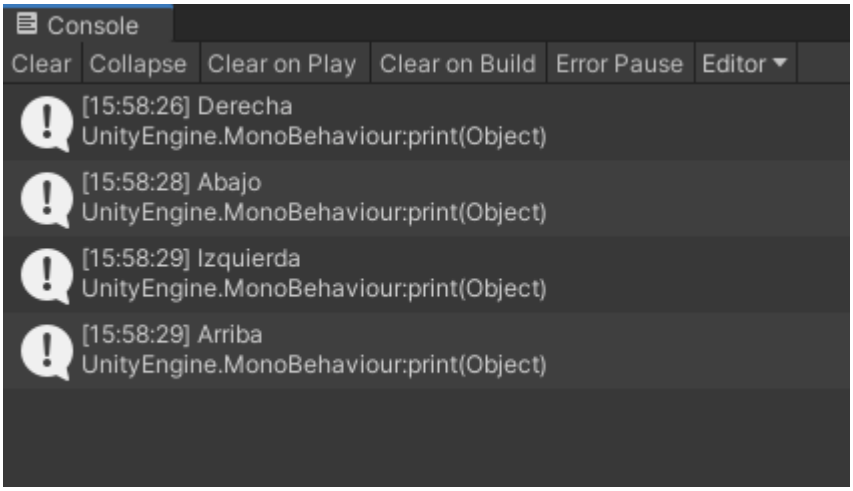


Figura 11. Resultado final en la consola de la prueba realizada.

Para la segunda prueba realizada, el objetivo es poder sustituir las acciones realizadas con los ojos, indicadas en la tabla 3, por acciones realizadas mediante el teclado. Las acciones y el resultado que se obtendría sería el siguiente.

Tabla 4. Controles de funcionamiento de la GUI para la segunda prueba realizada.

Acción	Resultado
Pulsar la tecla A	El cursor se desplaza en sentido horario dentro de la GUI
Pulsar la tecla D	El cursor se desplaza en sentido antihorario dentro de la GUI
Pulsar la tecla Espacio	Se selecciona el botón que esté activo en ese momento

Cuando el cursor se va moviendo alrededor de los cuatro botones de la interfaz, estos botones se iluminan si se encuentran seleccionados. Si al pulsar la tecla Espacio se encuentra un botón seleccionado, se muestra por un mensaje en la consola el nombre

del botón que ha sido seleccionado. Esto se consigue mediante la realización de 2 scripts: Manager.cs y Boton.cs.

Por un lado, el script Manager.cs se encarga de gestionar el movimiento dentro de la GUI y posee tres funciones fundamentales que permiten que el cursor de la GUI se mueva en sentido horario, presionando la tecla A, o antihorario, presionando la tecla D, y una última que permite mostrar por pantalla el nombre del botón seleccionado pulsando la tecla espacio. Estas funciones se pueden encontrar en las figuras 12, 13 y 14.

```
if (input.GetKeyDown("A"))
{
    botones[Posicion].Seleccionado = false;
    Posicion++;

    if (Posicion < 0)
    {
        Posicion = botones.Length - 1;
        botones[Posicion].Seleccionado = true;

        return;
    }
    if (Posicion > botones.Length - 1)
    {
        Posicion = 0;
        botones[Posicion].Seleccionado = true;

        return;
    }

    botones[Posicion].Seleccionado = true;


}
```

Figura 12. Función que permite poder mover el cursor de la interfaz en sentido horario.

```
if (input.GetKeyDown("D"))
{
    botones[Posicion].Seleccionado = false;
    Posicion--;

    if (Posicion < 0)
    {
        Posicion = botones.Length - 1;
        botones[Posicion].Seleccionado = true;

        return;
    }
    if (Posicion > botones.Length - 1)
    {
        Posicion = 0;
        botones[Posicion].Seleccionado = true;

        return;
    }

    botones[Posicion].Seleccionado = true;

}
```

Figura 13. Función que permite poder mover el cursor de la interfaz en sentido antihorario.

```
if (input.GetKeyDown("space"))
{
    if (botones[0].Seleccionado)
    {
        print("Arriba");

    }

    if (botones[1].Seleccionado)
    {
        print("Izquierda");

    }

    if (botones[2].Seleccionado)
    {
        print("Abajo");

    }

    if (botones[3].Seleccionado)
    {
        print("Derecha");

    }
}
```

Figura 14. Función que permite poder mostrar el nombre de cualquiera de los botones que componen la interfaz.

Por otro lado, se ha realizado también el script Boton.cs, que se encarga de cambiar el color de los botones en caso de estar seleccionados. La función que se

adjunta en la figura 15 permite cambiar a color azul los botones seleccionados y mantenerlos en color blanco en caso de no estar seleccionados.

```
if (Seleccionado)
{
    GetComponent<Image>().color = Color.blue;
}

else
{

    GetComponent<Image>().color = Color.white;
}
```

Figura 15. Función que permite cambiar de color a los botones si se encuentran seleccionados.

Para realizar la configuración de los botones de la interfaz, en primer lugar, se configura el Canvas, en el cual se añade el script Manager.cs, que gestiona el movimiento dentro de la GUI y una vez hecho, se agregan un total de cuatro elementos, correspondientes a los cuatro botones de la GUI y se vinculan a dichos cuatro botones.

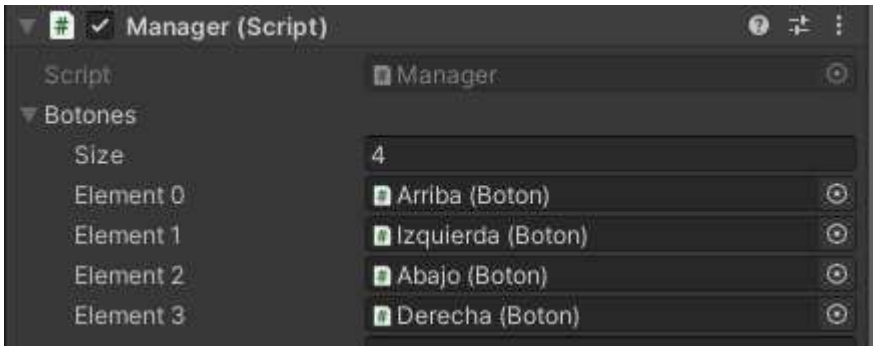


Figura 16. Configuración del Canvas para la segunda prueba.

En segundo lugar, para terminar de configurar la GUI, a cada botón situado dentro del Canvas se le añade el script Boton.cs. Una vez agregado el script, se puede observar una casilla que se marca cuando el botón en cuestión esté seleccionado.

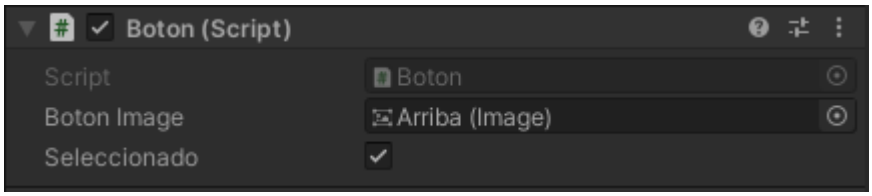


Figura 17. Configuración de uno de los botones de la GUI.

Cuando el cursor se va moviendo alrededor de los cuatro botones de la interfaz, estos botones se iluminan si se encuentran seleccionados. Si al pulsar la tecla Espacio se encuentra un botón seleccionado, se muestra por un mensaje en la consola el nombre del botón que ha sido seleccionado.

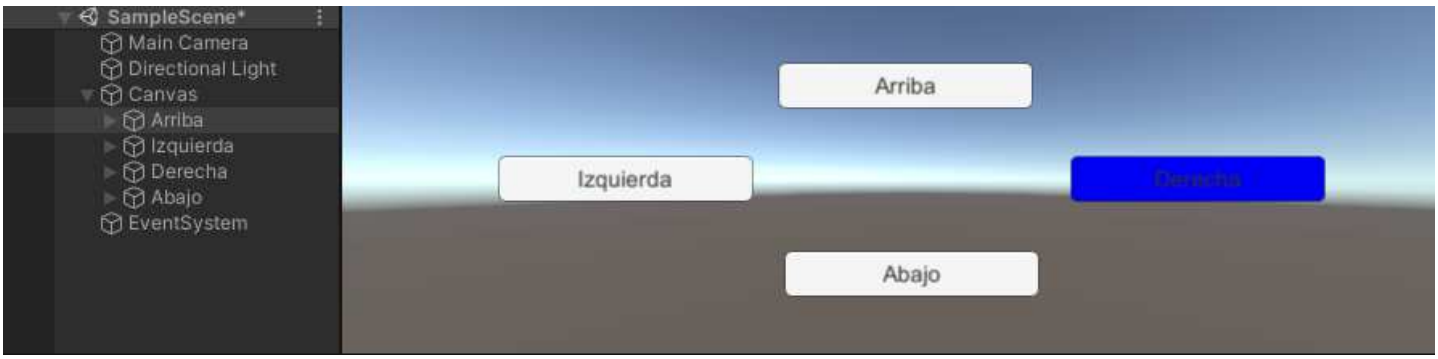


Figura 18. Imagen de la GUI cuando el botón derecho está seleccionado.

Cuando se pulsa la tecla espacio el resultado obtenido es igual que el que se muestra en la figura 11, en la cual se muestra por consola el nombre del botón seleccionado.

Estudio de las señales cerebrales

Adquisición de datos

Una vez desarrollada la GUI, se comienza con el estudio de las señales cerebrales, y para ello se comienza realizando registros de datos sobre 3 usuarios distintos mediante un EEG.

Para poder realizar el EEG, se coloca sobre la cabeza del usuario el gorro de tela descrito en el apartado de materiales y que posee los electrodos necesarios para realizar el EEG. Una vez colocado el gorro y comprobado que está bien ajustado a la cabeza del usuario, se coloca un gel capilar sobre cada uno de estos electrodos, que facilitarán la conducción de los impulsos eléctricos en el cerebro. Los electrodos irán conectados a cada uno de los 16 canales de los que puede registrar datos la placa Cyton + Daisy.

Tras colocar el gorro de tela y aplicar el gel capilar sobre cada uno de los electrodos, se inicializan los códigos necesarios para llevar a cabo la adquisición de señales. Estos códigos están hechos en lenguaje C++ y Python, y están proporcionados por los tutores de este TFG. Por un lado, los códigos en C++ permiten generar y mostrar por pantalla los protocolos con los que se va a trabajar en este TFG: ojos abiertos, ojos cerrados, ojo derecho cerrado y ojo izquierdo cerrado. Por otro lado, los códigos en Python son los encargados de generar el software para la adquisición y el guardar las señales cerebrales con la placa de OpenBCI. Una vez inicializados estos códigos, el protocolo experimental se trata de un protocolo de dos partes, la primera corresponde al estado basal en el que se muestra una cruz, la segunda a una acción, ambos con una duración 20 segundos y 10 segundos de descanso entre ellos. Dentro de las acciones

estudiadas existen tres opciones de estímulo: sonoro, que corresponde con la acción de cerrar ambos ojos en el cual un pitido indica cuando cerrarlos y otro cuando abrirlos, o una flecha, con dirección izquierda o derecha correspondiendo con el ojo que se tiene que cerrar. Una vez terminado el proceso de registro de datos, estos se almacenan en ficheros con formato CSV (valores separados por coma). En la figura 19 se puede observar un ejemplo de lo que se vería por pantalla a la hora de realizar uno de los registros de datos.

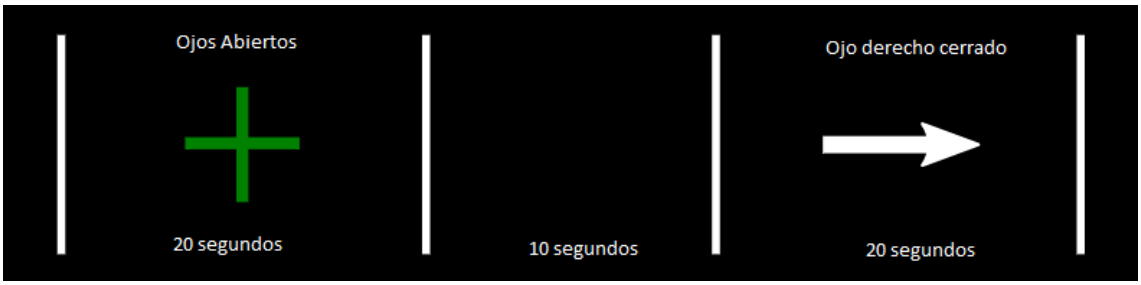


Figura 19 Ejemplo de lo que se vería por pantalla a la hora de hacer uno de los registros.

Estos registros se realizan 4 veces para los protocolos de ojos cerrados, ojo derecho cerrado y ojo izquierdo cerrado junto con el protocolo de ojos abiertos, obteniendo así un total de 12 ficheros CSV por cada usuario.

Por un lado, los archivos CSV obtenidos poseen un número no uniforme de columnas en las que se colocan los datos extraídos del EEG. Por otro lado, los ficheros CSV poseen también 17 filas, de las cuales las 16 primeras corresponden a los 16 canales procedentes de los electrodos del casco y la última de ellas correspondiente a un vector fila relleno principalmente por ceros salvo por cuatro espacios, correspondientes a una marca que indica el inicio y el final de las acciones. Cada marca

tiene un valor de 1, 2, 3 o 4 en función de la acción a la que corresponda: ojos abiertos, ojos cerrados, ojo derecho cerrado y ojo izquierdo cerrado.

BNQ	BNR	BNS	BNT	BNU	BNV	BNW	BNX	BNY
								closee
VarName1...	VarName1...	VarName1...	VarName1...	VarName1...	VarName1...	VarName1...	VarName1...	VarName1...
Number	Number	Number	Number	Number	Number	Number	Number	Number
19577.0883...	19695.9101...	19590.6334...	19652.1454...	19575.2554...	19688.8917...	19584.1514...	19643.5847...	19641.5730...
21203.1330...	21431.5007...	21243.0755...	21365.5407...	21203.2671...	21416.1451...	21238.9852...	21330.0015...	21354.8343...
1825.77989...	2011.94757...	1858.21227...	1953.76598...	1819.78962...	2010.09237...	1842.00726...	1923.99345...	1953.22954...
7018.44775...	7231.07990...	7056.95981...	7175.80403...	7017.17370...	7230.16348...	7050.61191...	7137.56020...	7181.12375...
11666.9624...	11899.0182...	11702.1440...	11827.2914...	11654.0654...	11884.7801...	11681.3569...	11773.0214...	11817.1884...
14208.8474...	14415.8693...	14245.6384...	14360.6158...	14218.1458...	14432.0296...	14255.6743...	14333.5478...	14360.2582...
4615.09878...	4817.71735...	4647.59822...	4761.63682...	4618.96563...	4809.80483...	4627.54870...	4706.94210...	4739.97798...
6460.72703...	6647.27469...	6494.36640...	6596.02214...	6477.62495...	6659.76931...	6492.84648...	6570.07176...	6583.23694...
31220.5143...	31333.2565...	31379.5916...	31433.4817...	31231.1314...	31329.2555...	31373.0202...	31202.4094...	31434.8675...
12115.6736...	12224.0125...	12268.2243...	12326.8306...	12124.5473...	12223.0738...	12259.2612...	12092.2490...	12323.2766...
28583.6343...	28691.3697...	28728.3842...	28780.4190...	28594.8549...	28677.1316...	28738.7330...	28555.3146...	28773.2665...
14578.1877...	14679.2623...	14693.9026...	14747.3233...	14600.4500...	14674.7248...	14686.1242...	14576.5560...	14743.1212...
15087.4722...	15202.9189...	15218.5875...	15284.4804...	15098.1563...	15194.8052...	15216.2629...	15078.7773...	15267.6049...
-2038.5461...	-1936.8233...	-1861.6544...	-1831.5913...	-2026.2973...	-1929.1343...	-1886.7778...	-2058.0815...	-1823.2094...
7132.01697...	7233.89622...	7267.55795...	7320.68804...	7144.98098...	7240.15471...	7271.35774...	7120.68463...	7326.45479...
5417.72757...	5521.21615...	5556.39780...	5610.39961...	5424.14253...	5513.86243...	5540.12573...	5390.70432...	5597.70382...
1	0	0	0	0	0	0	0	0

Figura 20. Extracto de uno de los ficheros CSV.

Una vez obtenidos todos los ficheros CSV, los siguientes pasos corresponden a trabajar con dichos datos, dentro de esto se pueden destacar cuatro partes: el preprocesamiento de los datos obtenidos mediante la adquisición de señales, la obtención de distintas características de los mismos, la obtención de diferentes mapas topográficos realizados con las características obtenidas en cada canal en los distintos protocolos y, por último, la obtención del modelo que clasifique las señales obtenidas por el EEG.

Preprocesamiento de los datos

Para la realización del preprocesamiento de los datos adquiridos, se codificará en Matlab una clase que permita almacenar dentro de una estructura los conjuntos de

datos denominados trials, previamente preprocesados con un filtro del cual se darán detalles más adelante junto con el marcador correspondiente a cada protocolo. Estos trials son los conjuntos de datos correspondientes a cada protocolo de 20 segundos.

Para que estos conjuntos de datos con los que se va a trabajar tengan todos un tamaño uniforme, se dividen los trials obtenidos por cada protocolo de 20 segundos en trials de 250 muestras por cada 2 segundos, obteniendo así un total de 125 muestras por segundo. Utilizando los ficheros CSV obtenidos, se realiza este mismo proceso para los 4 ficheros correspondientes a cada protocolo, aunque en el caso del protocolo de ojos abiertos, se obtienen un total de 40 trials de 120 posibles, para así poder obtener un número uniforme de trials para cada protocolo. Una vez realizado este proceso, se obtienen un total de 160 trials, 40 por cada protocolo, de 125 muestras por segundo.

Para averiguar que tipo de filtro se debe usar para la realización del preprocesamiento de los datos, se ha investigado en diferentes artículos sobre ello. En el primero de ellos, titulado *Eyes-closed resting EEG predicts the learning of alpha down-regulation in neurofeedback training* (Nan, 2018), se dice que a la hora de preprocesar los datos se aplica un filtro paso banda cuya banda de frecuencias oscila entre los 0.5 y 30 Hz.

Por otro lado, en otro de los artículos consultados, titulado *Classification of Neuroticism using Psychophysiological Signals During Speaking Task based on Two Different Baseline Measurements* (Izhar et al., 2019), aplican un filtro paso banda cuya banda de frecuencias oscila entre los 0.53 y 48 Hz.

Por tanto, basándonos en lo expuesto en los artículos mencionados, el filtro que se ha optado a utilizar para el preprocesamiento de los datos será un filtro paso banda con rango de frecuencias entre 0.5 y 30 Hz, al igual que en el primer artículo citado.

Tanto para la realización del preprocesado de los datos como para la obtención de características de estos, se ha hecho uso de la herramienta Brainflow, disponible para varias plataformas, entre ellas Matlab, y que permite realizar diferentes operaciones con los datos extraídos de las señales del EEG.

A la primera función proporcionada por Brainflow, la cual simula un filtro Butterworth paso banda, se le pasan los 160 trials y se colocan en una estructura de datos junto con el marcador correspondiente a cada protocolo utilizado. Esta función, que puede observarse en las últimas dos líneas del código que se expone en la figura 21 se añade al código utilizado para obtener los 160 trials, denominado Dataset.m, al que se puede acceder mediante este enlace: https://github.com/Eyes-EEG/signal_study/blob/main/Dataset.m. Como resultado final se obtiene una estructura en la cual se tiene en una columna todos los trials ya filtrados en el paso banda junto con su marcador.

```
BoardShim.set_log_file('brainflow.log');
BoardShim.enable_dev_board_logger();

params = BrainFlowInputParams();
board_shim = BoardShim(int32(BoardIDs.SYNTHETIC_BOARD), params);
board_shim.prepare_session();
board_shim.start_stream(45000, '');
pause(5);
board_shim.stop_stream();
data = board_shim.get_current_board_data(64);
board_shim.release_session();

eeg_channels =
BoardShim.get_eeg_channels(int32(BoardIDs.SYNTHETIC_BOARD));
% apply iir filter to the first eeg channel %
first_eeg_channel = eeg_channels(1);
original_data = data(first_eeg_channel, :);
sampling_rate =
BoardShim.get_sampling_rate(int32(BoardIDs.SYNTHETIC_BOARD));
filtered_data = DataFilter.perform_lowpass(original_data,
sampling_rate, 10.0, 3, int32(FilterTypes.BUTTERWORTH), 0.0);
```

Figura 21. Ejemplo de código proporcionado por Brainflow para preprocesamiento de datos.

Obtención de características

Ya realizado el preprocesamiento de las señales, el siguiente objetivo es el de obtener las características que nos aporten una mayor información. Consultando de nuevo el primer artículo citado y un nuevo artículo titulado *EEG differences between eyes-closed and eyes-open resting conditions* (Barry et al., 2007) se llega a la conclusión de que la banda de frecuencias en la que se sitúan las ondas alpha es de la que más información se obtiene.

Una vez que se sabe que del rango de frecuencias de las ondas alpha es de donde se puede sacar la mayor información, el siguiente paso es codificar en Matlab una clase que nos permita obtener una tabla con los valores de alpha en los 16 canales de todos los trials. Esta tabla estará compuesta por 17 columnas, en las cuales estarán los valores en alfa de los 16 canales además de los marcadores correspondientes a la acción utilizada, y de 160 filas, correspondientes a los 160 trials que se usan para obtener los

valores en el rango de frecuencias de las ondas alpha. Esta clase se denomina Features.m, en el que se encuentra tanto la segunda función utilizada proporcionada por Brainflow, y que puede observarse en las últimas siete líneas del código que se expone en la figura 22, para calcular los valores en el rango de frecuencias de las ondas alpha de todos los canales, como las funciones necesarias para obtener la tabla mencionada con anterioridad a partir de dichos valores se encuentra disponible en el siguiente enlace: https://github.com/Eyes-EEG/signal_study/blob/main/Features.m.

Junto con la realización de las clases Dataset.m y Features.m se debe redactar también un código que sirva para inicializar estas clases. Dicho fichero se denomina main.m y es posible acceder a él desde este enlace: https://github.com/Eyes-EEG/signal_study/blob/main/main.m.

```
BoardShim.set_log_file('brainflow.log');
BoardShim.enable_dev_board_logger();

params = BrainFlowInputParams();
board_shim = BoardShim(int32(BoardIDs.SYNTHETIC_BOARD), params);
board_id = int32(BoardIDs.SYNTHETIC_BOARD);
board_descr = BoardShim.get_board_descr(board_id);
sampling_rate = int32(board_descr.sampling_rate);
board_shim.prepare_session();
board_shim.start_stream(45000, '');
pause(10);
board_shim.stop_stream();
nfft = DataFilter.get_nearest_power_of_two(sampling_rate);
data = board_shim.get_board_data();
board_shim.release_session();

eeg_channels = board_descr.eeg_channels;
eeg_channel = eeg_channels(3);
original_data = data(eeg_channel, :);
detrended = DataFilter.detrend(original_data,
int32(DetrendOperations.LINEAR));
[ampls, freqs] = DataFilter.get_psd_welch(detrended, nfft, nfft / 2,
sampling_rate, int32(WindowFunctions.HANNING));
band_power_alpha = DataFilter.get_band_power(ampls, freqs, 7.0, 13.0);
band_power_beta = DataFilter.get_band_power(ampls, freqs, 14.0, 30.0);
ratio = band_power_alpha / band_power_beta;
```

Figura 22. Ejemplo de código proporcionado por Brainflow para estudio de frecuencias.

Fields	canal1	canal2	canal3	canal4	canal5	canal6	canal7
1	1.7000e+04	8.6785e+03	737.4320	1.0692e+03	9.1758e+03	9.5141e+03	250.0927
2	1.6961e+04	8.6604e+03	753.2940	1.0710e+03	9.0587e+03	9.3433e+03	232.6642
3	1.6896e+04	8.6213e+03	754.9019	1.0660e+03	9.0749e+03	9.3195e+03	227.3433
4	1.7094e+04	8.7324e+03	761.5521	1.0792e+03	9.1663e+03	9.5053e+03	261.3479
5	1.7102e+04	8.7505e+03	768.2216	1.0740e+03	9.1982e+03	9.4459e+03	251.2675
6	1.7009e+04	8.6969e+03	763.8680	1.0696e+03	9.1685e+03	9.3223e+03	232.3478
7	1.7011e+04	8.7763e+03	743.2789	1.0685e+03	9.2837e+03	9.3871e+03	260.7763
8	1.6899e+04	8.6534e+03	744.1595	1.0642e+03	9.2260e+03	9.2763e+03	250.5808
9	1.6861e+04	8.6973e+03	746.4000	1.0587e+03	9.3124e+03	9.3154e+03	265.5329
10	1.6745e+04	8.6801e+03	740.0981	1.0620e+03	9.2492e+03	9.2238e+03	242.7811
11	1.6912e+04	8.4333e+03	824.4916	1.0661e+03	1.0171e+04	8.8338e+03	235.1443
12	1.6842e+04	8.2825e+03	825.1734	1.0265e+03	1.0175e+04	8.6626e+03	225.5873

Figura 23. Extracto de la tabla obtenida con los valores en los rangos de frecuencia de las ondas alpha.

Realización de mapas topográficos

El siguiente objetivo dentro de la realización del estudio de señales es el de graficar distintos mapas topográficos con la finalidad de observar si existe diferencia a nivel visual entre los registros de datos adquiridos con anterioridad.

Para ello se utiliza una herramienta de Matlab denominada Topographic EEG/MEG Plot, utilizada para realizar tanto mapas topográficos sobre EEG como magnetoencefalografía (MEG). La función que nos proporciona esta herramienta se encuentra en el siguiente enlace: https://es.mathworks.com/matlabcentral/fileexchange/72729-topographic-eeg-meg-plot

El procedimiento a seguir para obtener los mapas topográficos es, en primer lugar, pasarle a dicha función proporcionada tanto la posición de los electrodos situados en el casco para realizar la EEG como los valores en alpha de cada uno de los

canales. Para obtener las posiciones de los electrodos, se ha hecho uso de la siguiente imagen, que igual que el casco sigue el sistema 10/20.

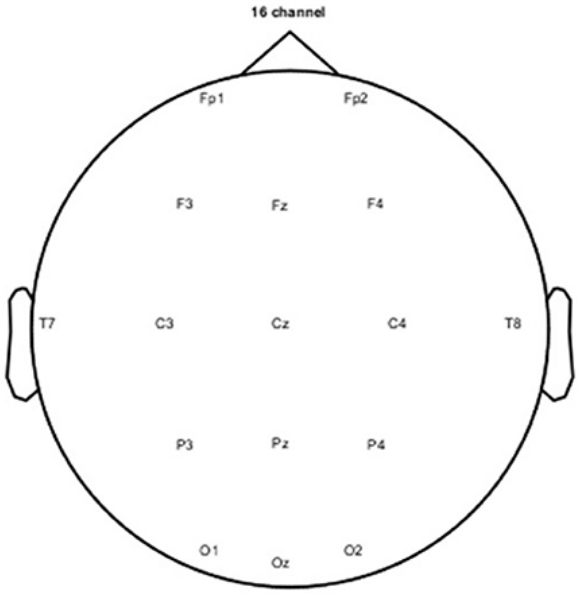


Figura 24. Posiciones de los electrodos para la realización del EEG.

El segundo parámetro que se le pasa a esta función son los valores en alpha de cada canal, obtenidos mediante el uso de la clase Features.m, para cada uno de estos electrodos. Este proceso se lleva a cabo una vez para cada acción, pudiendo ver así si existen diferencias visuales entre los mapas topográficos correspondientes a cada acción. En las figuras 25, 26, 27 y 28 se muestran los cuatro mapas correspondientes las cuatro accines. Se adjunta en el siguiente uno de los códigos utilizados para graficar los mapas topográficos: https://github.com/Eyes-EEG/signalstudy/tree/main/Topographic-eeg-meg-plot

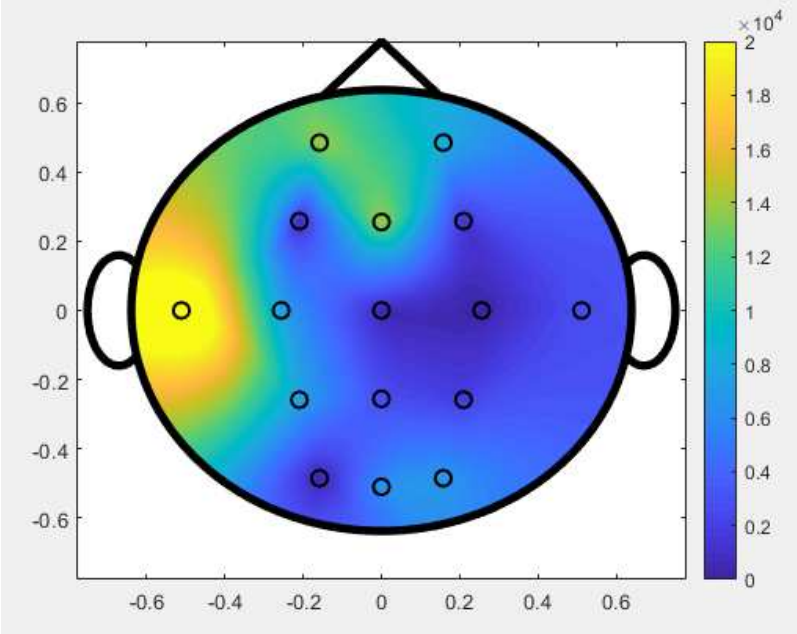


Figura 25. Mapa topográfico correspondiente al protocolo de ojos abiertos.

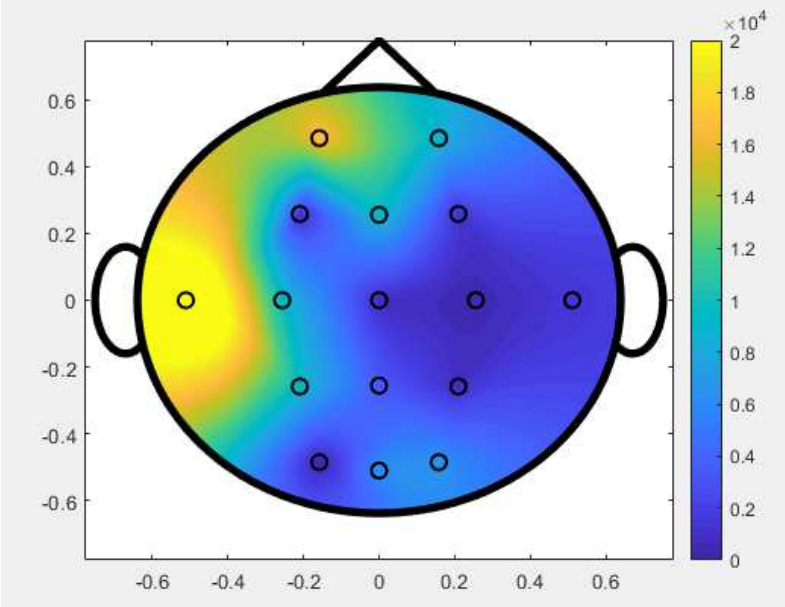


Figura 26. Mapa topográfico correspondiente al protocolo de ojos cerrados.

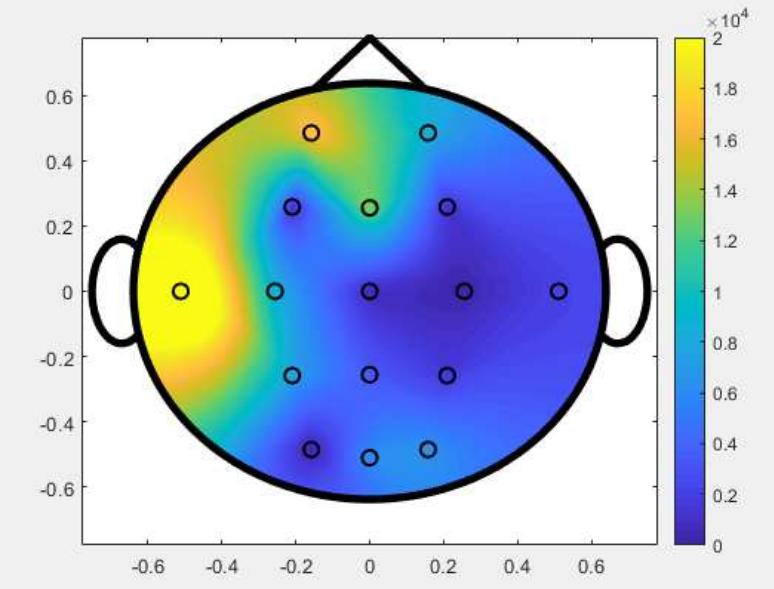


Figura 27. Mapa topográfico correspondiente al protocolo de ojo derecho cerrado.

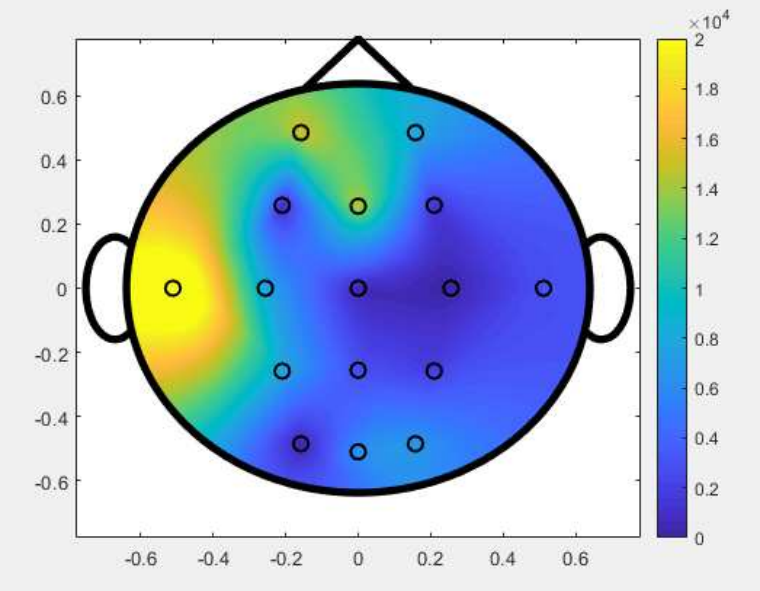


Figura 28. Mapa topográfico correspondiente al protocolo de ojo izquierdo cerrado.

Tal y como se puede observar en los 4 mapas topográficos representados la diferencia entre ellos en ínfima. Dentro del mapa topográfico correspondiente al protocolo de ojos abiertos se ve cierta diferencia con el resto en la parte delantera del

mapa, pero salvando dicha diferencia, los mapas aparentan ser prácticamente iguales entre ellos.

Obtención del modelo que clasifique las señales

Para la obtención del modelo que clasifique los datos procedentes de las señales del EEG se hará uso de una aplicación de Matlab denominada Classification Learner, que permite entrenar modelos para clasificar datos. Gracias a esta aplicación, es posible explorar el aprendizaje automático supervisado utilizando varios clasificadores (*Aprendiz de clasificación*, s.f.).

Partiendo de la tabla obtenida con la clase Features.m y gracias a esta aplicación, se podrá obtener el modelo que pueda clasificar las señales que se obtienen del EEG.

La principal función del modelo obtenido es la de clasificar, partiendo de los datos que vaya recibiendo en streaming del EEG, la acción que debe enviarse a Unity, haciendo que la GUI reaccione de una forma o de otra en función de la acción que recibe Unity.

Para comenzar, se inicializa el Classification Learner (ver Figura 26) y se le añade la tabla de características obtenida con anterioridad, para posteriormente comenzar con la configuración del programa. Una vez agregada la tabla, se debe seleccionar la variable que va a clasificar el modelo, que en este caso se trata del "label" (etiqueta), el marcador asignado a cada acción estudiada. Además, se debe seleccionar el tipo de validación de los datos. En este caso se ha hecho uso de una técnica denominada validación cruzada, en la cual se establece un número de campos entre los

cuales se dividen los datos de la tabla que se le pasa al classification learner. Una vez establecido el número de campos, se ha de dividir el número total de muestras entre dicho número de campos establecido y ese será el número de muestras que se utilizarán para la validación del modelo. Los restantes son los que se utilizarán para entrenar dicho modelo. Este proceso se repite un número de veces igual al del número de campos establecido. Por ejemplo, si se dispone de un total de 4000 muestras y se establece un total de cinco campos, el número de muestras que se utilizaría para el entrenamiento del modelo sería de 3200 y de 800 para la validación del mismo. El proceso se llevaría a cabo un total de cinco veces. Para el entrenamiento del modelo se ha establecido un total de cinco campos, valor por defecto que proporciona la aplicación, de los cuales cuatro de ellos se utilizan para entrenar el modelo y el restante para la validación del mismo.

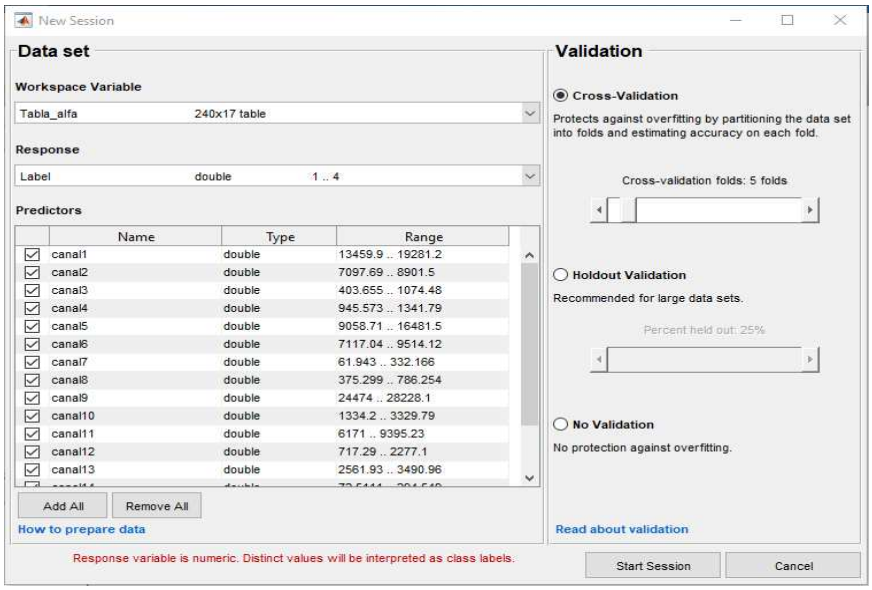


Figura 29. Configuración inicial del Classification Learner.

Una vez realizada esta primera configuración, el siguiente paso es entrenar todos los posibles clasificadores y seleccionar el que pueda proporcionar una mayor

exactitud. Este proceso se lleva a cabo, tal y como se menciona con anterioridad, con tres usuarios, aunque para evitar que este apartado sea muy tedioso, se adjuntan únicamente los resultados obtenidos con uno de ellos, ya que la diferencia entre los resultados obtenidos respecto a los restantes usuarios es ínfima.

Usuario 1

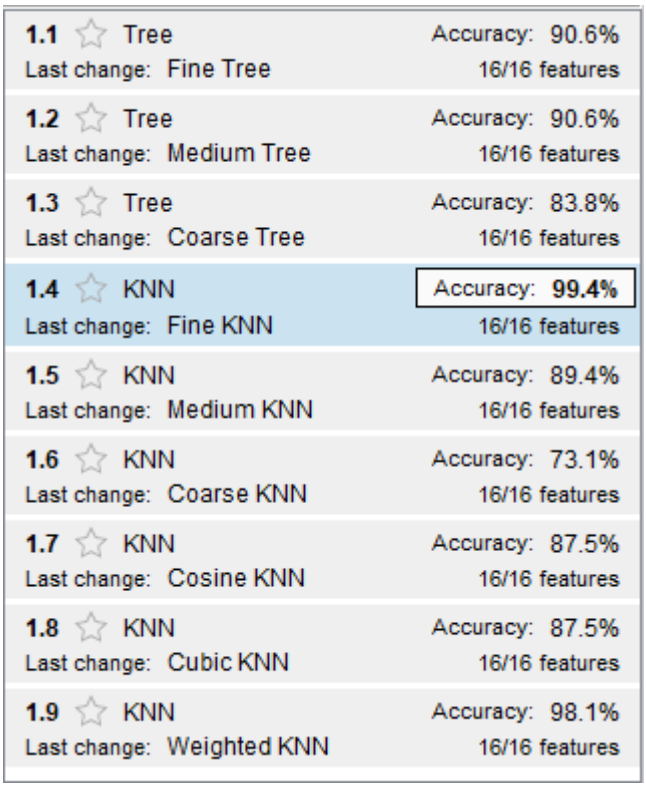


Figura 30. Lista de todos los clasificadores entrenados junto a su respectiva exactitud.

Tal y como se observa en la figura 30, El clasificador que proporciona una mayor exactitud y por tanto una mayor fiabilidad es el clasificador con vecinos más cercanos (KNN), más concretamente el Fine KNN, con una exactitud del 99,4%.

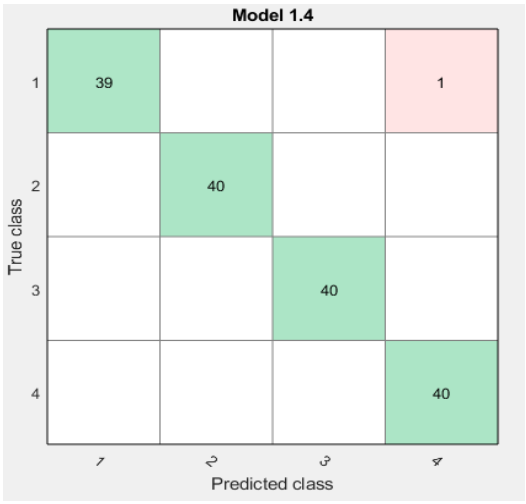


Figura 31. Representación de la matriz de confusión para el usuario 1

Si se observa la figura 31, correspondiente a la matriz de confusión para el caso del usuario 1, se puede deducir que para el caso de los protocolos 2 (ojos cerrados), 3 (ojo derecho cerrado) y 4 (ojo izquierdo cerrado), la relación entre el número de clases verdaderas y las clases predichas es del 100%, en cambio, para los protocolos 1 (ojos abiertos), la relación es casi perfecta, salvo por una clase.

Por último, en las figuras 32, 33, 34 y 35 se puede observar la curva de características de funcionamiento del receptor (ROC), utilizada para averiguar cómo enfocar la exactitud del modelo, correspondiente a cada una de las 4 clases con las que se ha trabajado.

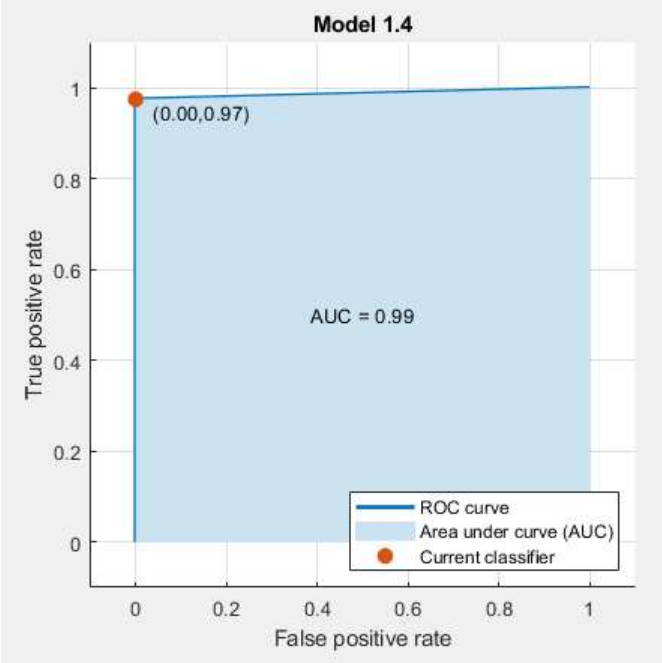


Figura 32. Curva ROC correspondiente al protocolo de ojos abiertos.

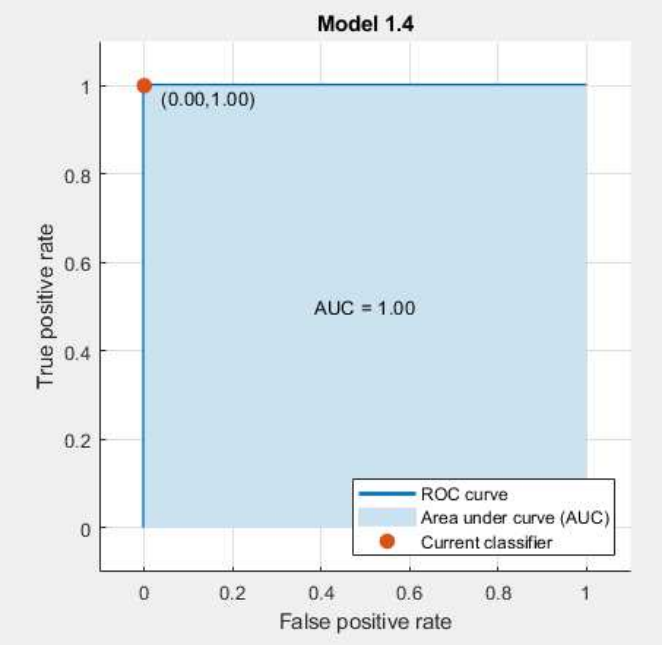


Figura 33. Curva ROC correspondiente al protocolo de ojos cerrados.

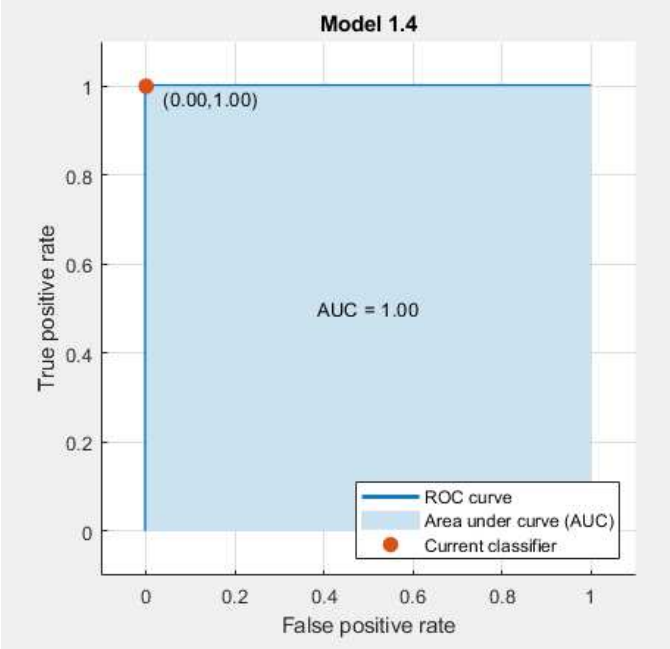


Figura 34. Curva ROC correspondiente al protocolo de ojo derecho cerrado.

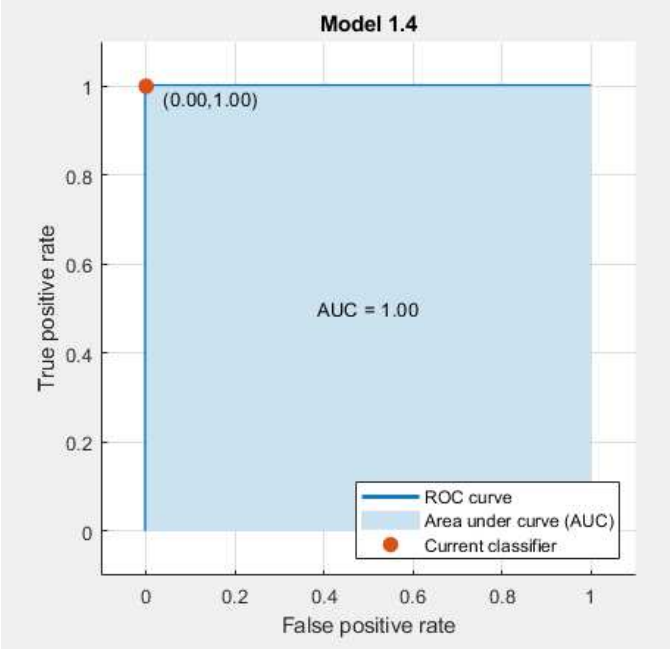


Figura 35. Curva ROC correspondiente al protocolo de ojo izquierdo cerrado.

Implementación de las comunicaciones entre la GUI y el BCI

En esta última parte del funcionamiento del sistema BCI el objetivo es, tal y como se expone en el apartado de objetivos, realizar el diseño y la implementación del protocolo de comunicaciones entre la GUI y el BCI. Para ello, se añaden las clases realizadas en Matlab al código en streaming proporcionado por los tutores de este TFG que permite clasificar en streaming los datos procedentes de las señales del EEG obteniendo los comandos que se le pasan a la GUI y así interactuar con la GUI desarrollada.

Para establecer la comunicación entre Matlab y Unity se hará uso de una conexión TCP-IP entre Matlab y Unity3D, que nos permite la conexión entre estos dos softwares Para la realización de esta parte del TFG, se tomará a Matlab como cliente, que se encarga de enviar los comandos a Unity, que actúa como servidor recibiendo dichos comandos desde Matlab. A continuación, se adjuntan los códigos proporcionados por esta herramienta que nos permiten configurar a Matlab como cliente y a Unity como servidor.

Por un lado, el código de Matlab, que se puede observar en la figura 36, se trata de una función que se encarga de iniciar la conexión con el servidor, mandar los comandos y por último cerrar dicha conexión.

Por otro lado, el código de Unity, que se puede observar en la figura 37, se trata de una clase que se encarga de inicializar el protocolo de comunicaciones entre el cliente y el servidor y posteriormente se puede observar, entre las líneas 24 y 35 del código expuesto en la figura 37 una función que permite recibir los comandos desde Matlab.

```
clc
clear all
tcpipClient = tcpip('127.0.0.1',55001,'NetworkRole','Client');
set(tcpipClient,'Timeout',30);
fopen(tcpipClient);
Action = '3';
fwrite(tcpipClient,Action);
fclose(tcpipClient);
```

Figura 36. Código proporcionado para configurar Matlab como cliente.

```
using UnityEngine;
using System.Collections;
using System.Net;
using System.Net.Sockets;
using System.Linq;
using System;
using System.IO;
using System.Text;
public class readSocket : MonoBehaviour
{
    // Use this for initialization
    TcpListener listener;
    String msg;
    void Start()
    {
        //listener=new TcpListener (55001);
        listener = new TcpListener(IPAddress.Parse("127.0.0.1"), 55001);
        listener.Start();
        print("is listening");
    }
    // Update is called once per frame
    void Update()
    {
        if (!listener.Pending())
        {
        }
        else
        {
            print("socket comes");
            TcpClient client = listener.AcceptTcpClient();
            NetworkStream ns = client.GetStream();
            StreamReader reader = new StreamReader(ns);
            msg = reader.ReadToEnd();
            print(msg);
        }
    }
}
```

Figura 37. Código proporcionado para configurar Unity como servidor.

Con estas funciones, antes de configurar el código en streaming, el primer paso es comprobar la comunicación entre Matlab y Unity. Para ello se escribe en Matlab el

código de la figura 36, para posteriormente modificar el código Manager.cs, que define el funcionamiento de la GUI, añadiendo a este el código en C# que se observa en la figura 37 y también sustituyendo la parte en la cual se utiliza el teclado para comprobar el funcionamiento de la GUI por los comandos que se van recibiendo desde Matlab. Una vez escritos estos códigos, se inicializa en primer lugar Unity, ya que al actuar de servidor debe de estar atento a lo que le llegue del cliente, y en segundo lugar se inicializa Matlab, que cada vez que se inicialice, manda un comando a Unity, provocando a su vez que la GUI reaccione.

El resultado obtenido a partir de este proceso es el que se describe en la tabla 5, en el que Matlab envía los comandos y la GUI reacciona a estos. En el caso de que se seleccione uno de los botones, se muestra un mensaje por consola con el nombre del botón seleccionado.

Tabla 5. Acciones de la GUI en función del comando recibido

Comando	Resultado
2	El cursor se desplaza en sentido horario dentro de la GUI
3	El cursor se desplaza en sentido antihorario dentro de la GUI
4	Se selecciona el botón que esté activo en ese momento

Resultados

Una vez realizada esta prueba para comprobar las comunicaciones entre Matlab y Unity, el siguiente paso es colocar sobre el usuario sobre el que se va a realizar el

EEG el gorro con los 16 electrodos necesarios para obtener los datos correspondientes a los 16 canales de los cuales permite adquirir señales la placa utilizada para ello. Tras la colocación del gorro y la posterior comprobación de está bien colocado el dispositivo, se procede a aplicar el gel capilar sobre cada uno de los electrodos para permitir una mejor conducción eléctrica de los impulsos y se comienza con la realización de las pruebas. Se debe comenzar con el entrenamiento de los datos registrados con anterioridad en tiempo real para proseguir con la clasificación de los datos mediante el uso del código en streaming.

El funcionamiento de los códigos en streaming es el siguiente:

- Adquisición de datos y colocación de estos en ficheros CSV.
- Preprocesamiento de los datos mediante filtro paso banda (Butterworth, rango de frecuencias de 0.5 a 30 Hz).
- Obtención de la tabla de valores en el rango de frecuencias de ondas alpha.
- Clasificación de los datos de la tabla y posterior decodificación de los datos en forma de comandos.

Una vez realizado todo este proceso, Matlab envía a Unity los comandos que decodifica y actúa de una forma u otra en función de los resultados obtenidos en la clasificación de los datos. Las pruebas con el sistema se realizarán con dos usuarios diferentes

Usuario 1

El resultado obtenido en las pruebas realizadas con este usuario no es el esperado, ya que se detecta en todo momento que se está cerrando el ojo derecho sin

parar. A pesar de realizar numerosos intentos abriendo y cerrando uno o ambos ojos, el sistema sigue sin responder a ninguna de las órdenes que se le envían. Dado que el resultado de hacer las pruebas ha sido fallido, se ha procedido a realizar las pruebas con un nuevo usuario.

Usuario 2

En el caso de las pruebas realizadas sobre este usuario, el resultado obtenido al principio de la prueba seguía sin ser el esperado, ya que el sistema detectaba que el usuario tenía los ojos continuamente cerrados. Aunque, posteriormente, realizando diferentes acciones con los ojos, sí que el sistema llegó a responder bien en algún momento, aunque no con la fiabilidad que cabría esperar antes de realizar estas pruebas con el sistema.

Tras investigar cual podría ser la causa de que el sistema esté fallando tanto, una de las conclusiones a las que se llegó es que existe una gran probabilidad de que haya un sobreajuste en el algoritmo de clasificación de los datos. Se le denomina sobreajuste al hecho de llevar a cabo un modelo tan ajustado a los datos de entrenamiento, en el caso de este TFG, los datos extraídos por el EEG, que provoca que el algoritmo de clasificación no interprete bien los datos de test, en el caso de este TFG, los datos recibidos mediante el código en streaming. Para comprobar si realmente existe sobreajuste o no dentro del algoritmo de clasificación desarrollado, se ha repetido el entrenamiento de los datos adquiridos para los registros, con la salvedad de que en esta ocasión no se ha hecho con el 100% de los datos registrados, si no que se ha hecho con una cantidad muy pequeña de estos. El resultado obtenido en cuanto a la exactitud de los clasificadores ronda un 45 y un 60%, alrededor de la mitad de lo que

se obtiene al realizar el entrenamiento con todos los datos registrados, en los que se obtiene unos resultados de entre un 97 y 99%. tal y como se expone en el este artículo titulado *Overfitting in Machine Learning: what it is and how to prevent it*, obtener unos resultados como los mencionados anteriormente hacen que la posibilidad de que exista sobreajuste en el algoritmo de clasificación sea muy elevada, por lo que se concluye que esta es, muy probablemente, la causa de que el sistema haya fallado tan estrepitosamente.

Dado que el sistema no termina de ser lo suficientemente fiable, se realizará una última prueba con la GUI y un script similar al empleado para probar las comunicaciones entre la GUI y el BCI. El script utilizado se muestra en la figura 50.

La única diferencia entre estos scripts es que en el utilizado para la realización de esta prueba se utiliza un bucle for para ir enviar una secuencia de comandos a la GUI y así observar la reacción de la GUI y el tiempo empleado por esta para realizar todas las acciones. El script realizado en Matlab para llevar a cabo esta prueba se adjunta en la figura 38.

```
clc
clear all
tcpipClient = tcpip('127.0.0.1',55001,'NetworkRole','Client');
set(tcpipClient,'Timeout',30);
a = [2 2 4 3 2 4];
for i = 1:6
fopen(tcpipClient);
b=a(i);
c=int2str(b);
fwrite(tcpipClient,c);
fclose(tcpipClient);
end
```

Figura 38. Código realizado en Matlab para probar la GUI.

Tras haber realizado esta última prueba con la GUI el resultado obtenido es que la GUI funciona de manera correcta y los tiempo de ejecución de las acciones realizadas por esta son instantáneos.

Conclusiones

El objetivo principal de este TFG es el de desarrollar un protocolo de comunicación entre una interfaz gráfica de usuario (GUI) y una interfaz cerebro-ordenador (BCI). El primer paso llevado a cabo ha sido el diseño y la programación de la interfaz, en la cual se realizaron diferentes pruebas con ellas mediante el uso del teclado y del ratón. Tras haber logrado desarrollar la GUI, se comenzó con el estudio de diferentes registros de datos adquiridos mediante la realización de un EEG. Dentro de este apartado se preprocesaron dichos datos, se obtuvieron sus características más relevantes y se clasificaron mediante el uso de un algoritmo de Machine Learning. Por último, se estableció el protocolo de comunicaciones entre la GUI y el BCI. Dado que el sistema no fue fiable a la hora de realizar las pruebas con el código en streaming, se optó por realzar una última prueba en la que se le envía a la GUI una secuencia de comandos para corroborar su buen funcionamiento y el tiempo de ejecución de las acciones realizadas. Una vez realizada esta última prueba, se concluye que el funcionamiento de la GUI es correcto y que los tiempos de ejecución de las acciones son instantáneos.

Con los resultados obtenidos mediante la realización de todas estas pruebas, de cara a las líneas futuras de este proyecto se puede encontrar la mejora de la selección de características, pudiendo elegir, por ejemplo, las ondas beta en lugar de las alpha, la mejora del preprocesado de los datos adquiridos, haciendo uso de filtros con diferentes características al usado en este TFG, la mejora del algoritmo de clasificación, ya que, observando las exactitudes obtenidas mediante las diferentes pruebas, es muy probable que haya sido la principal causa del fallo en el sistema. Una

vez lograda una mayor optimización del sistema, este sistema BCI podría implementarse en un futuro, por ejemplo, en un videojuego de realidad virtual basado en un sistema BCI y valorar entonces la experiencia de usuario con dicho videojuego de realidad virtual.

Conclusions

The main objective of this TFG is to develop a communication protocol between a graphical user interface (GUI) and a brain-computer interface (BCI). The first step carried out was the design and programming of the interface, in which different tests were carried out using the keyboard and the mouse. After having successfully developed the GUI, we began with the study of different data records acquired by means of an EEG. In this section, the data were preprocessed, their most relevant characteristics were obtained and classified using a Machine Learning algorithm. Finally, the communication protocol between the GUI and the BCI was established. Given that the system was not reliable when testing the streaming code, it was decided to carry out a final test in which a sequence of commands is sent to the GUI to corroborate its proper functioning and the execution time of the actions carried out. Once this last test has been carried out, it is concluded that the GUI works correctly and that the execution times of the actions are instantaneous.

With the results obtained by carrying out all these tests, with a view to the future lines of this project, we can find the improvement of the selection of features, being able to choose, for example, beta waves instead of alpha waves, the improvement of the preprocessing of the acquired data, making use of filters with different

characteristics to the one used in this TFG, the improvement of the classification algorithm, since, observing the accuracies obtained through the different tests, it is very likely that it has been the main cause of the failure in the system. Once a greater optimisation of the system has been achieved, this BCI system could be implemented in the future, for example, in a virtual reality video game based on a BCI system and the user experience with this virtual reality video game could then be evaluated.

Bibliografía

El mayor logro de la neurociencia: las interfaces cerebro-ordenador (s.f.). Cerebrum. Recuperado de https://cerebrum.la/2020/07/26/el-mayor-logro-de-la-neurociencia-la-interfaz-cerebro-computadora/

Cyton + Daisy Biosensong Boards (16-Channels) (s.f.). OpenBCI. [Figura] Recuperado de https://shop.openbci.com/products/cyton-daisy-biosensing-boards-16-channel?variant=38959256526

Electroencefalograma (s.f.) TopDoctors. Recuperado de https://www.topdoctors.es/diccionario-medico/electroencefalograma

Selecting buttons through the keyboard (s.f.). Packt. Recuperado de https://subscription.packtpub.com/book/game_development/9781785885822/1/ch011v11sec16/selecting-buttons-trough-the-keyboard

Canvas (2016). Unity. Recuperado de https://docs.unity3d.com/e/UICanvas.html

Neelarghya (15 de noviembre de 2015). *Multiple Cameras and When to use them.[Unity3D].* Recuperado de https://medium.com/xrpractices/multiple-cameras-and-when -to-use-them-unity3d-c271cb9a1f02

Unity. (1 de julio de 2020). *How to work with multiple scenes in Unity.* [Archivo de vídeo]. https://www.youtube.com/watch?v=zObWVOv1GlE

ButterGames – Desarrollo de videojuegos. (23 de enero de 2018). *Como hacer un menú con teclado unity 5 | como hacer un menú y un juego en unity 5.* [Archivo de vídeo]. https://www.youtube.com/watch?v=5Wf-fNopxHE

Nan, W.; Wan, F.; Tang, Q.; Wong, C. M.; Wang, B.; Rosa, A. (2018) Eyes-closed resting EEG predicts the learning of alpha down-regulation in neurofeedback training. *Front. Psychol*, ***9***, 1–7, doi:10.3389/fpsyg.2018.01607

Gopika Gopan, K.; Sinha, N.; Dinesh Babu, J. Statistical feature analysis for EEG baseline classification: Eyes Open vs Eyes Closed. *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON* 2017, doi:10.1109/TENCON.2016.7848476

Barry, R. J.; Clarke, A. R.; Johnstone, S. J.; Magee, C. A.; Rushby, J. A. EEG differences between eyes-closed and eyes-open resting conditions. 2007, ***118***, 2765–2773, doi:10.1016/j.clinph.2007.07.028

Martínez, J.A. (2015). *Definición de GUI.* dis.um.es. Recuperado de https://dis.um.es/~lopezquesada/documentos/IES_1415/LMSGI/curso/xhtml/html12/html/definicion.html

Graphical User Interface (GUI) (s.f.). citamayob3. [Figura]. Recuperado de https://citamayob3.wixsite.com/interfaz-de-usuario/single-post/2017/09/03/graphical-user-interface-gui

EEG (electroencefalograma) (29 de julio de 2021). Mayo clinic. Recuperado de https://www.mayoclinic.org/es-es/tests-procedures/eeg/about/pac-20393875

Actividad registrada por encefalograma (29 de julio de 2021). Mayo clinic. [Figura]. Recuperado de https://www.mayoclinic.org/es-es/tests-procedures/eeg/about/pac-20393875

Onda cerebral (20 de septiembre de 2019). Wikipedia. Recuperado de https://es.wikipedia.org/wiki/Onda_cerebral

¿Qué son las ondas cerebrales? (24 de julio de 2019). Neurofeedback. Recuperado de https://www.neurofeedback.cat/que-son-las-ondas-cerebrales/

Tipos de ondas cerebrales (s.f.). Esto va de salud [Figura]. Recuperado de https://www.estovadesalud.com/mente-y-espiritu/tipos-de-ondas-cerebrales/

Arquitectura BCI (3 de agosto de 2021). Wikipedia. [Figura]. Recuperado de https://es.wikipedia.org/wiki/Interfaz_cerebro-computadora

Cyton + Daisy (s.f.). tdrobotica. [Figura]. Recuperado de https://tienda.tdrobotica.co/home/1621-cyton-daisy-biosensing-boards-16-canales.html

Matlab examples (2018). Brainflow. Recuperado de https://brainflow.readthedocs.io/en/stable/Examples.html#matlab

Mit License (2018). Brainflow. Recuperado de https://brainflow.readthedocs.io/en/stable/License.html

Aprendiz de clasificación (2020). Mathworks. Recuperado de https://es.mathworks.com/help/stats/classificationlearner-app.html

Validación cruzada (s.f.). Wikipedia. Recuperado de https://es.wikipedia.org/wiki/Validaci%C3%B3n_cruzada

Martínez-Cagigal, V. (2020). *Topographic EEG/MEG plot*. Mathworks Recuperado de https://es.mathworks.com/matlabcentral/fileexchange/72729-topographic-eeg-meg-plot

Nugroho, L. (2020). *TCP-IP Connection between Matlab and Unity3D*. Mathworks. Recuperado de https://es.mathworks.com/matlabcentral/fileexchange/75728-tcp-ip-connection-between-matlab-and-unity3d

Anexos

Anexo 1: códigos utilizados para la implementación de la GUI

https://github.com/Eyes-EEG/Interface

Anexo 2: códigos utilizados para el estudio de las señales recibidas por el EEG

https://github.com/Eyes-EEG/signal_study

Anexo 3: códigos utilizados para la comunicación entre Matlab y Unity

https://github.com/Eyes-EEG/streaming_codes_and_Unity_project

Anexo 4: datasheets de los componentes Texas Instruments ADS1299 ADC y LIS3DH accelerometer

ADS1299-x Low-Noise, 4-, 6-, 8-Channel, 24-Bit, Analog-to-Digital Converter for EEG and Biopotential Measurements

1 Features

- Up to Eight Low-Noise PGAs and Eight High-Resolution Simultaneous-Sampling ADCs
- Input-Referred Noise: 1 μV_{PP} (70-Hz BW)
- Input Bias Current: 300 pA
- Data Rate: 250 SPS to 16 kSPS
- CMRR: –110 dB
- Programmable Gain: 1, 2, 4, 6, 8, 12, or 24
- Unipolar or Bipolar Supplies:
  - Analog: 4.75 V to 5.25 V
  - Digital: 1.8 V to 3.6 V
- Built-In Bias Drive Amplifier, Lead-Off Detection, Test Signals
- Built-In Oscillator
- Internal or External Reference
- Flexible Power-Down, Standby Mode
- Pin-Compatible with the ADS129x
- SPI-Compatible Serial Interface
- Operating Temperature Range: –40°C to +85°C

2 Applications

- Medical Instrumentation Including:
  - Electroencephalogram (EEG) Study
  - Fetal Electrocardiography (ECG)
  - Sleep Study Monitor
  - Bispectral Index (BIS)
  - Evoked Audio Potential (EAP)

3 Description

The ADS1299-4, ADS1299-6, and ADS1299 devices are a family of four-, six-, and eight-channel, low-noise, 24-bit, simultaneous-sampling delta-sigma ($\Delta\Sigma$) analog-to-digital converters (ADCs) with a built-in programmable gain amplifier (PGA), internal reference, and an onboard oscillator. The ADS1299-x incorporates all commonly-required features for extracranial electroencephalogram (EEG) and electrocardiography (ECG) applications. With its high levels of integration and exceptional performance, the ADS1299-x enables the creation of scalable medical instrumentation systems at significantly reduced size, power, and overall cost.

The ADS1299-x has a flexible input multiplexer per channel that can be independently connected to the internally-generated signals for test, temperature, and lead-off detection. Additionally, any configuration of input channels can be selected for derivation of the patient bias output signal. Optional SRB pins are available to route a common signal to multiple inputs for a referential montage configuration. The ADS1299-x operates at data rates from 250 SPS to 16 kSPS. Lead-off detection can be implemented internal to the device using an excitation current sink or source.

Multiple ADS1299-4, ADS1299-6, or ADS1299 devices can be cascaded in high channel count systems in a daisy-chain configuration. The ADS1299-x is offered in a TQFP-64 package specified from –40°C to +85°C.

Device Information[1]

PART NUMBER	PACKAGE	BODY SIZE (NOM)
ADS1299-x	TQFP (64)	10.00 mm × 10.00 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Block Diagram

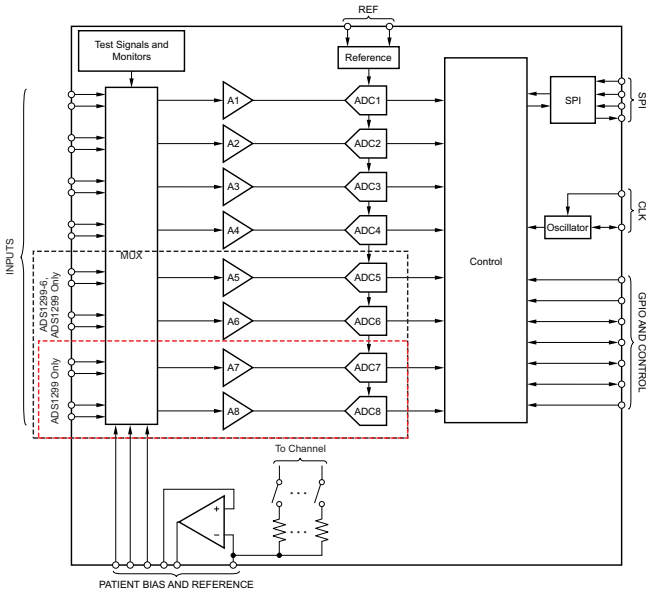


Table of Contents

4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision B (October 2016) to Revision C **Page**

- Changed *Maximum Junction* parameter name to *Junction* in *Absolute Maximum Ratings* table ... 7
- Changed *Recommended Operating Conditions* table: changed *free-air* to *ambient* in conditions statement, changed specifications of *Input voltage* parameter, and added V_{CM} and f_{CLK} symbols ... 8
- Changed conditions statement of *Electrical Characteristics* table: added T_A to temperature conditions, moved DVDD condition to after AVDD – AVSS condition ... 9
- Changed *Input bias current* parameter test conditions from *input* to *InxP and INxN* ... 9
- Changed *Drift* parameter unit from *ppm* to *ppm/°C* and changed *Internal clock accuracy* parameter test conditions from *–40°C ≤ T_A ≤ +85°C* to *T_A = –40°C to +85°C* in *Electrical Characteristics* table ... 10
- Changed I_{AVDD} and I_{DVDD} parameters [deleted *(normal mode)* from parameter names and added *Normal mode* to test conditions], and deleted *Quiescent* from *Power dissipation* parameter name in *Electrical Characteristics* table ... 11
- Changed *free-air* to *ambient* in conditions statement of *Timing Requirements: Serial Interface* table ... 12
- Changed *Analog Input* section ... 22
- Changed Table 9 cross-reference to Table 7 in *Settling Time* section ... 34
- Changed *Ideal Output Code versus Input Signal* table: changed all V_{REF} in first column to *FS* in and deleted footnote 1 ... 38
- Changed reset settings of bits 4 and 3 in bit register of CONFIG1 register ... 46
- Changed reset value settings of bits 7 to 5 in CONFIG2 register: split cells apart ... 47
- Changed reset value settings of bits 6 to 5 in CONFIG3 register: split cells apart ... 48
- Changed *AVDD – AVSS* to *AVDD + AVSS* in description of bit 3 in *Configuration Register 3 Field Descriptions* ... 48
- Changed *Lead-Off Control Register Field Descriptions* table: changed 01 bit setting of bits 3:2 to *24 nA* from *12 nA* changed description of bits 1:0 ... 49
- Changed *Unused Inputs and Outputs* section: added $\overline{\text{DRDY}}$ description, deleted statement of not floating unused

Revision History (continued)

Changes from Revision A (August 2012) to Revision B **Page**

Changes from Original (July 2012) to Revision A **Page**

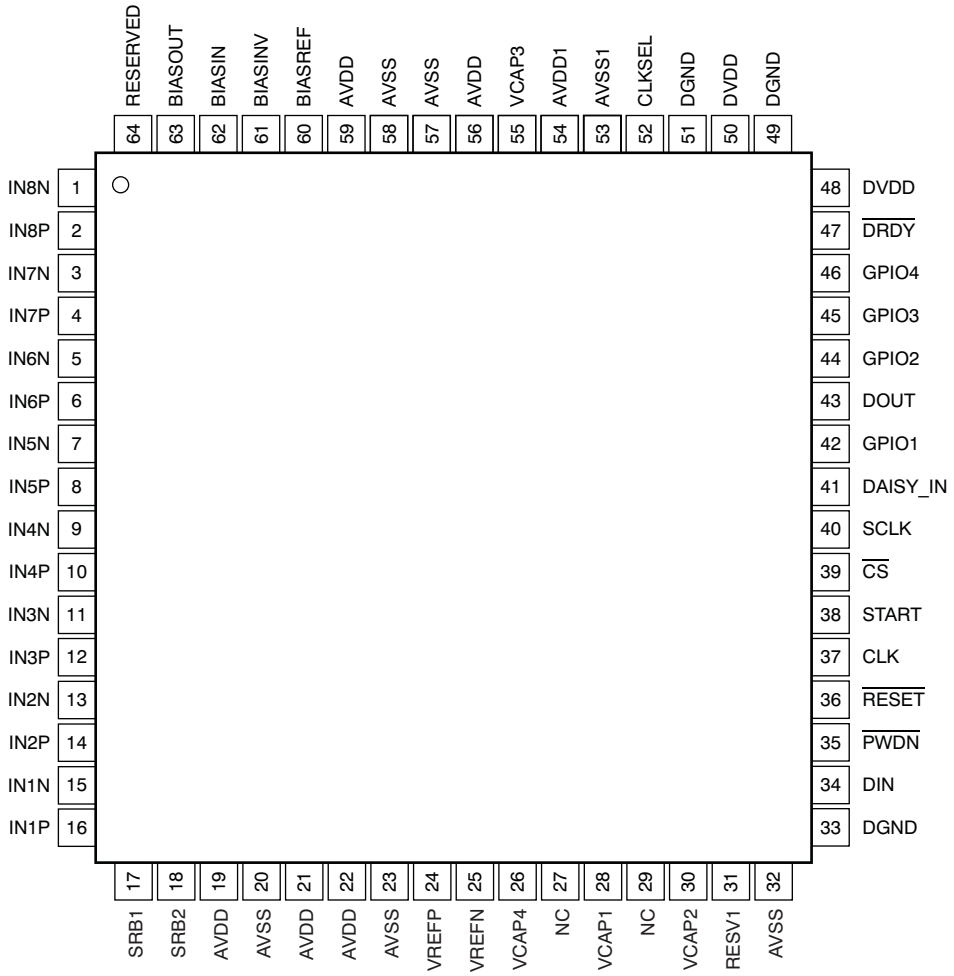
- Changed product column of Family and Ordering Information table 1

5 Device Comparison

PRODUCT	PACKAGE OPTIONS	OPERATING TEMPERATURE RANGE	CHANNELS	ADC RESOLUTION	MAXIMUM SAMPLING RATE
ADS1299-4	TQFP-64	–40°C to +85°C	4	24	16 kSPS
ADS1299-6	TQFP-64	–40°C to +85°C	6	24	16 kSPS
ADS1299	TQFP-64	–40°C to +85°C	8	24	16 kSPS

6 Pin Configuration and Functions

PAG Package
64-Pin TQFP
Top View



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
AVDD	19, 21, 22, 56, 59	Supply	Analog supply. Connect a 1-µF capacitor to AVSS.
	59	Supply	Charge pump analog supply. Connect a 1-µF capacitor to AVSS, pin 58.
AVDD1	54	Supply	Analog supply. Connect a 1-µF capacitor to AVSS1.
AVSS	20, 23, 32, 57	Supply	Analog ground
	58	Supply	Analog ground for charge pump
AVSS1	53	Supply	Analog ground
BIASIN	62	Analog input	Bias drive input to MUX
BIASINV	61	Analog input/output	Bias drive inverting input
BIASOUT	63	Analog output	Bias drive output
BIASREF	60	Analog input	Bias drive noninverting input
$\overline{CS}$	39	Digital input	Chip select, active low
CLK	37	Digital input	Master clock input
CLKSEL	52	Digital input	Master clock select(1)
DAISY_IN	41	Digital input	Daisy-chain input
DGND	33, 49, 51	Supply	Digital ground
DIN	34	Digital input	Serial data input
DOUT	43	Digital output	Serial data output
$\overline{DRDY}$	47	Digital output	Data ready, active low
DVDD	48, 50	Supply	Digital power supply. Connect a 1-µF capacitor to DGND.
GPIO1	42	Digital input/output	General-purpose input/output pin 1. Connect to DGND with a ≥10-kΩ resistor if unused.
GPIO2	44	Digital input/output	General-purpose input/output pin 2. Connect to DGND with a ≥10-kΩ resistor if unused.
GPIO3	45	Digital input/output	General-purpose input/output pin 3. Connect to DGND with a ≥10-kΩ resistor if unused.
GPIO4	46	Digital input/output	General-purpose input/output pin 4. Connect to DGND with a ≥10-kΩ resistor if unused.
IN1N	15	Analog input	Differential analog negative input 1(2)
IN1P	16	Analog input	Differential analog positive input 1(2)
IN2N	13	Analog input	Differential analog negative input 2(2)
IN2P	14	Analog input	Differential analog positive input 2(2)
IN3N	11	Analog input	Differential analog negative input 3(2)
IN3P	12	Analog input	Differential analog positive input 3(2)
IN4N	9	Analog input	Differential analog negative input 4(2)
IN4P	10	Analog input	Differential analog positive input 4(2)
IN5N	7	Analog input	Differential analog negative input 5(2) (ADS1299-6 and ADS1299 only)
IN5P	8	Analog input	Differential analog positive input 5(2) (ADS1299-6 and ADS1299 only)
IN6N	5	Analog input	Differential analog negative input 6(2) (ADS1299-6 and ADS1299 only)
IN6P	6	Analog input	Differential analog positive input 6(2) (ADS1299-6 and ADS1299 only)
IN7N	3	Analog input	Differential analog negative input 7(2) (ADS1299 only)
IN7P	4	Analog input	Differential analog positive input 7(2) (ADS1299 only)
IN8N	1	Analog input	Differential analog negative input 8(2) (ADS1299 only)
IN8P	2	Analog input	Differential analog positive input 8(2) (ADS1299 only)
NC	27, 29	—	No connection, leave as open circuit
Reserved	64	Analog output	Reserved for future use, leave as open circuit
$\overline{RESET}$	36	Digital input	System reset, active low
RESV1	31	Digital input	Reserved for future use, connect directly to DGND
SCLK	40	Digital input	Serial clock input
SRB1	17	Analog input/output	Patient stimulus, reference, and bias signal 1
SRB2	18	Analog input/output	Patient stimulus, reference, and bias signal 2

(1) Set the two-state mode setting pins high to DVDD or low to DGND through ≥10-kΩ resistors.
(2) Connect unused analog inputs directly to AVDD.

Pin Functions (continued)

PIN		TYPE	DESCRIPTION
NAME	NO.		
START	38	Digital input	Synchronization signal to start or restart a conversion
$\overline{\text{PWDN}}$	35	Digital input	Power-down, active low
VCAP1	28	Analog output	Analog bypass capacitor pin. Connect a 100-µF capacitor to AVSS.
VCAP2	30	Analog output	Analog bypass capacitor pin. Connect a 1-µF capacitor to AVSS.
VCAP3	55	Analog output	Analog bypass capacitor pin. Connect a parallel combination of 1-µF and 0.1-µF capacitors to AVSS.
VCAP4	26	Analog output	Analog bypass capacitor pin. Connect a 1-µF capacitor to AVSS.
VREFN	25	Analog input	Negative analog reference voltage.
VREFP	24	Analog input/output	Positive analog reference voltage. Connect a minimum 10-µF capacitor to VREFN.

7 Specifications

7.1 Absolute Maximum Ratings(1)

		MIN	MAX	UNIT
Voltage	AVDD to AVSS	–0.3	5.5	V
	DVDD to DGND	–0.3	3.9	
	AVSS to DGND	–3	0.2	
	VREFP to AVSS	–0.3	AVDD + 0.3	
	VREFN to AVSS	–0.3	AVDD + 0.3	
	Analog input	AVSS – 0.3	AVDD + 0.3	
	Digital input	DGND – 0.3	DVDD + 0.3	
Current	Input, continuous, any pin except power supply pins(2)	–10	10	mA
Temperature	Junction, T_J		150	°C
	Storage, T_{stg}	–60	150	

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) Input pins are diode-clamped to the power-supply rails. Limit the input current to 10 mA or less if the analog input voltage exceeds AVDD + 0.3 V or is less than AVSS – 0.3 V, or if the digital input voltage exceeds DVDD + 0.3 V or is less than DGND – 0.3 V.

7.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001(1)	±1000	V
		Charged-device model (CDM), per JEDEC specification JESD22-C101(2)	±500	

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

7.3 Recommended Operating Conditions

over operating ambient temperature range (unless otherwise noted)

			MIN	NOM	MAX	UNIT
POWER SUPPLY						
	Analog power supply	AVDD to AVSS	4.75	5	5.25	V
	Digital power supply	DVDD to DGND	1.8	1.8	3.6	V
	Analog to Digital supply	AVDD – DVDD	–2.1		3.6	V
ANALOG INPUTS						
	Full-scale differential input voltage	$V_{INxP} – V_{INxN}$		$\pm V_{REF}$ / gain		V
V_{CM}	Input common-mode range	$(V_{INxP} + V_{INxN}) / 2$		See the *Input Common-Mode Range* subsection of the *PGA Settings and Input Range* section		
VOLTAGE REFERENCE INPUTS						
V_{REF}	Reference input voltage	$V_{REF} = (V_{VREFP} – V_{VREFN})$		4.5		V
VREFN	Negative input			AVSS		V
VREFP	Positive input			AVSS + 4.5		V
CLOCK INPUT						
f_{CLK}	External clock input frequency	CLKSEL pin = 0	1.5	2.048	2.25	MHz
DIGITAL INPUTS						
	Input voltage		DGND – 0.1		DVDD + 0.1	V
TEMPERATURE RANGE						
T_A	Operating temperature range		–40		85	°C

7.4 Thermal Information

THERMAL METRIC(1)		ADS1299-4, ADS1299-6, ADS1299 PAG (TQFP) 64 PINS	UNIT
$R_{\theta JA}$	Junction-to-ambient thermal resistance	46.2	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	5.8	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	19.6	°C/W
ψ_{JT}	Junction-to-top characterization parameter	0.2	°C/W
ψ_{JB}	Junction-to-board characterization parameter	19.2	°C/W
$R_{\theta JC(bot)}$	Junction-to-case (bottom) thermal resistance	n/a	°C/W

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report.

7.5 Electrical Characteristics

Minimum and maximum specifications apply from T_A = –40°C to 85°C. Typical specifications are at T_A = +25°C. All specifications are at AVDD – AVSS = 5 V, DVDD = 3.3 V, V_{REF} = 4.5 V, external f_{CLK} = 2.048 MHz, data rate = 250 SPS, and gain = 12 (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
ANALOG INPUTS						
	Input capacitance			20		pF
	Input bias current	T_A = +25°C, InxP and INxN = 2.5 V			±300	pA
		T_A = –40°C to +85°C, InxP and INxN = 2.5 V		±300		
	DC input impedance	No lead-off	1000			MΩ
		Current source lead-off detection ($I_{LEADOFF}$ = 6 nA)		500		
PGA PERFORMANCE						
	Gain settings			1, 2, 4, 6, 8, 12, 24		
BW	Bandwidth			See Table 5		
ADC PERFORMANCE						
	Resolution		24			Bits
DR	Data rate	f_{CLK} = 2.048 MHz	250		16000	SPS
DC CHANNEL PERFORMANCE						
	Input-referred noise (0.01 Hz to 70 Hz)	10 seconds of data, gain = 24 (1)		1		μV_{PP}
		250 points, 1 second of data, gain = 24, T_A = +25°C		1	1.35	
		250 points, 1 second of data, gain = 24, T_A = –40°C to +85°C		1	1.6	
		All other sample rates and gain settings		See *Noise Measurements*		
INL	Integral nonlinearity	Full-scale with gain = 12, best fit		8		ppm
	Offset error			60		µV
	Offset error drift			80		nV/°C
	Gain error	Excluding voltage reference error		0.1	±0.5	% of FS
	Gain drift	Excluding voltage reference drift		3		ppm/°C
	Gain match between channels			0.2		% of FS
AC CHANNEL PERFORMANCE						
CMRR	Common-mode rejection ratio	f_{CM} = 50 Hz and 60 Hz (2)	–110	–120		dB
PSRR	Power-supply rejection ratio	f_{PS} = 50 Hz and 60 Hz		96		dB
	Crosstalk	f_{IN} = 50 Hz and 60 Hz		–110		dB
SNR	Signal-to-noise ratio	V_{IN} = –2 dBFs, f_{IN} = 10-Hz input, gain = 12		121		dB
THD	Total harmonic distortion	V_{IN} = –0.5 dBFs, f_{IN} = 10 Hz		–99		dB
PATIENT BIAS AMPLIFIER						
	Integrated noise	BW = 150 Hz		2		μV_{RMS}
	Gain bandwidth product	50-kΩ || 10-pF load, gain = 1		100		kHz
	Slew rate	50-kΩ || 10-pF load, gain = 1		0.07		V/µs
THD	Total harmonic distortion	f_{IN} = 10 Hz, gain = 1		–80		dB
	Common-mode input range		AVSS + 0.3		AVDD – 0.3	V
	Short-circuit current			1.1		mA
	Quiescent power consumption			20		µA

(1) Noise data measured in a 10-second interval. Test not performed in production. Input-referred noise is calculated with the input shorted (without electrode resistance) over a 10-second interval.
(2) CMRR is measured with a common-mode signal of AVSS + 0.3 V to AVDD – 0.3 V. The values indicated are the minimum of the eight channels.

Electrical Characteristics (continued)

Minimum and maximum specifications apply from T_A = –40°C to 85°C. Typical specifications are at T_A = +25°C. All specifications are at AVDD – AVSS = 5 V, DVDD = 3.3 V, V_{REF} = 4.5 V, external f_{CLK} = 2.048 MHz, data rate = 250 SPS, and gain = 12 (unless otherwise noted)

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNIT
LEAD-OFF DETECT							
	Frequency		Continuous		At dc, f_{DR} / 4, see *Register Maps* for settings		Hz
			One time or periodic		7.8, 31.2		
	Current		ILEAD_OFF[1:0] = 00		6		nA
			ILEAD_OFF[1:0] = 01		24		
			ILEAD_OFF[1:0] = 10		6		µA
			ILEAD_OFF[1:0] = 11		24		
	Current accuracy				±20%		
	Comparator threshold accuracy				±30		mV
EXTERNAL REFERENCE							
	Input impedance				5.6		kΩ
INTERNAL REFERENCE							
V_{REF}	Internal reference voltage				4.5		V
	V_{REF} accuracy				±0.2%		
	Drift		T_A = –40°C to +85°C		35		ppm/°C
	Start-up time				150		ms
SYSTEM MONITORS							
	Reading error	Analog supply			2%		
		Digital supply			2%		
	Device wake up		From power-up to $\overline{\text{DRDY}}$ low		150		ms
			STANDBY mode		31.25		µs
	Temperature sensor reading	Voltage	T_A = +25°C		145		mV
		Coefficient			490		µV/°C
	Test signal	Signal frequency	See *Register Maps* section for settings		f_{CLK} / 2^{21}, f_{CLK} / 2^{20}		Hz
		Signal voltage	See *Register Maps* section for settings		±1, ±2		mV
		Accuracy			±2%		
CLOCK							
	Internal oscillator clock frequency		Nominal frequency		2.048		MHz
	Internal clock accuracy		T_A = +25°C			±0.5%	
			T_A = –40°C to +85°C			±2.5%	
	Internal oscillator start-up time				20		µs
	Internal oscillator power consumption				120		µW
DIGITAL INPUT/OUTPUT (DVDD = 1.8 V to 3.6 V)							
V_{IH}	High-level input voltage			0.8 DVDD		DVDD + 0.1	V
V_{IL}	Low-level input voltage			–0.1		0.2 DVDD	V
V_{OH}	High-level output voltage		I_{OH} = –500 µA	0.9 DVDD			V
V_{OL}	Low-level output voltage		I_{OL} = +500 µA			0.1 DVDD	V
	Input current		0 V < $V_{DigitalInput}$ < DVDD	–10		10	µA

Electrical Characteristics (continued)

Minimum and maximum specifications apply from T_A = –40°C to 85°C. Typical specifications are at T_A = +25°C. All specifications are at AVDD – AVSS = 5 V, DVDD = 3.3 V, V_{REF} = 4.5 V, external f_{CLK} = 2.048 MHz, data rate = 250 SPS, and gain = 12 (unless otherwise noted)

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNIT
SUPPLY CURRENT (Bias Turned Off)							
I_{AVDD}	AVDD current	ADS1299-4	Normal mode, AVDD – AVSS = 5 V		4.06		mA
		ADS1299-6			5.57		
		ADS1299			7.14		
I_{DVDD}	DVDD current	ADS1299-4	Normal mode, DVDD = 3.3 V		0.54		mA
		ADS1299-6			0.66		
		ADS1299			1		
		ADS1299-4	Normal mode, DVDD = 1.8 V		0.27		
		ADS1299-6			0.34		
		ADS1299			0.5		
POWER DISSIPATION (Analog Supply = 5 V, Bias Amplifiers Turned Off)							
	Power dissipation	ADS1299-4	Normal mode		22	24	mW
			Power-down		10		µW
			Standby mode, internal reference		5.1		mW
		ADS1299-6	Normal mode		30	33	mW
			Power-down		10		µW
			Standby mode, internal reference		5.1		mW
		ADS1299	Normal mode		39	42	mW
			Power-down		10		µW
			Standby mode, internal reference		5.1		mW

7.6 Timing Requirements: Serial Interface

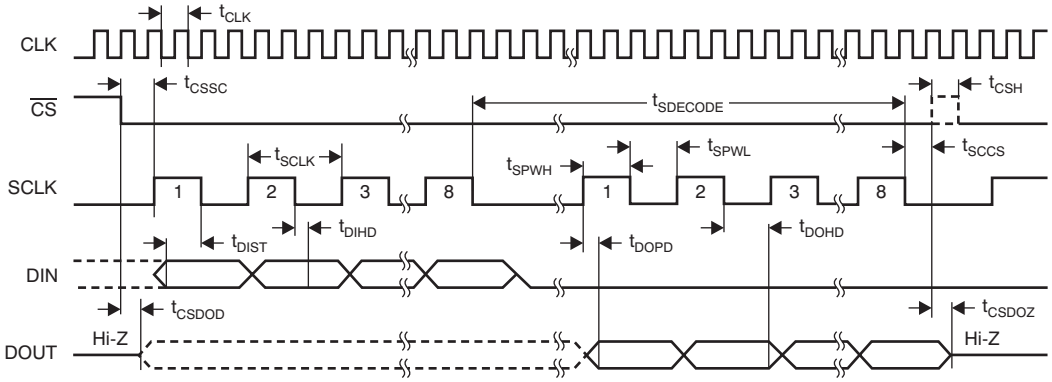
over operating ambient temperature range (unless otherwise noted)

		2.7 V ≤ DVDD ≤ 3.6 V		1.8 V ≤ DVDD ≤ 2.0 V		
		MIN	MAX	MIN	MAX	UNIT
t_{CLK}	Master clock period	414	666	414	666	ns
t_{CSSC}	Delay time, $\overline{CS}$ low to first SCLK	6		17		ns
t_{SCLK}	SCLK period	50		66.6		ns
$t_{SPWH, L}$	Pulse duration, SCLK pulse duration, high or low	15		25		ns
t_{DIST}	Setup time, DIN valid to SCLK falling edge	10		10		ns
t_{DIHD}	Hold time, valid DIN after SCLK falling edge	10		11		ns
t_{CSH}	Pulse duration, $\overline{CS}$ high	2		2		t_{CLK}
t_{SCCS}	Delay time, final SCLK falling edge to $\overline{CS}$ high	4		4		t_{CLK}
$t_{SDECODE}$	Command decode time	4		4		t_{CLK}
$t_{DISCK2ST}$	Setup time, DAISY_IN valid to SCLK rising edge	10		10		ns
$t_{DISCK2HT}$	Hold time, DAISY_IN valid after SCLK rising edge	10		10		ns

7.7 Switching Characteristics: Serial Interface

over operating ambient temperature range (unless otherwise noted)

PARAMETER		2.7 V ≤ DVDD ≤ 3.6 V		1.8 V ≤ DVDD ≤ 2.0 V		
		MIN	MAX	MIN	MAX	UNIT
t_{DOHD}	Hold time, SCLK falling edge to invalid DOUT	10		10		ns
t_{DOPD}	Propagation delay time, SCLK rising edge to DOUT valid		17		32	ns
t_{CSDOD}	Propagation delay time, $\overline{CS}$ low to DOUT driven	10		20		ns
t_{CSDOZ}	Propagation delay time, $\overline{CS}$ high to DOUT Hi-Z		10		20	ns



NOTE: SPI settings are CPOL = 0 and CPHA = 1.

Figure 1. Serial Interface Timing

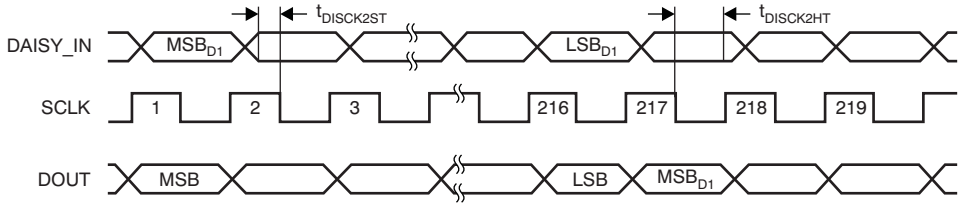


Figure 2. Daisy-Chain Interface Timing

7.8 Typical Characteristics

At T_A = 25°C, AVDD = 5 V, AVSS = 0 V, DVDD = 3.3 V, internal VREFP = 4.5 V, VREFN = AVSS, external clock = 2.048 MHz, data rate = 250 SPS, and gain = 12 (unless otherwise noted)

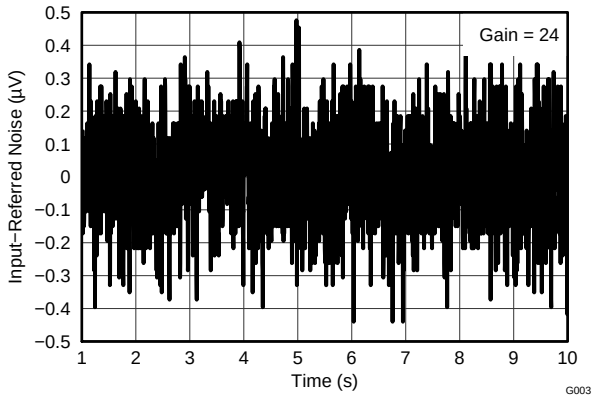


Figure 3. Input-Referred Noise

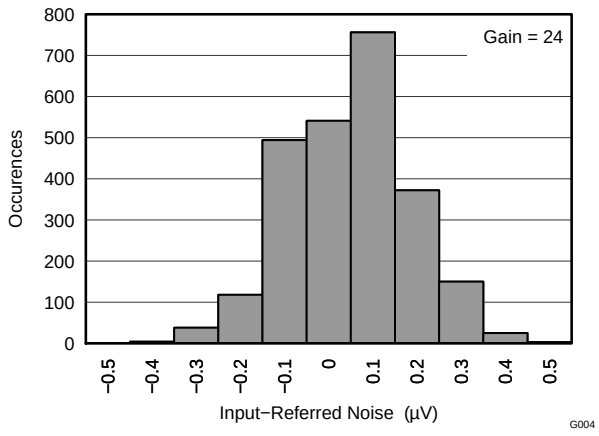


Figure 4. Noise Histogram

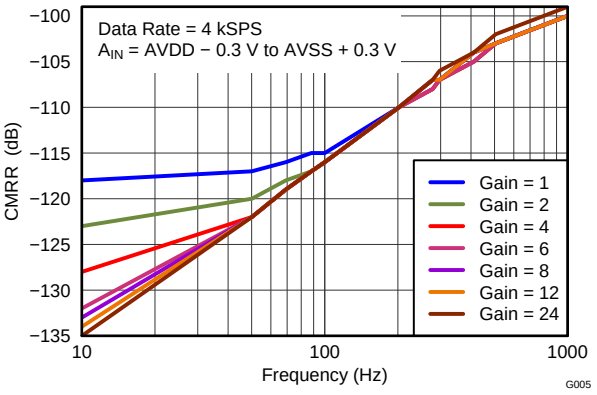


Figure 5. Common-Mode Rejection Ratio vs Frequency

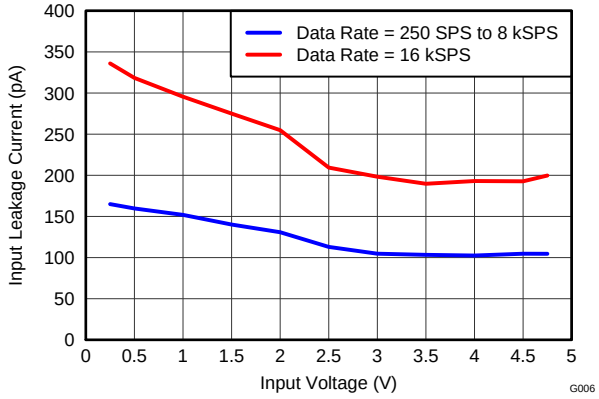


Figure 6. Leakage Current vs Input Voltage

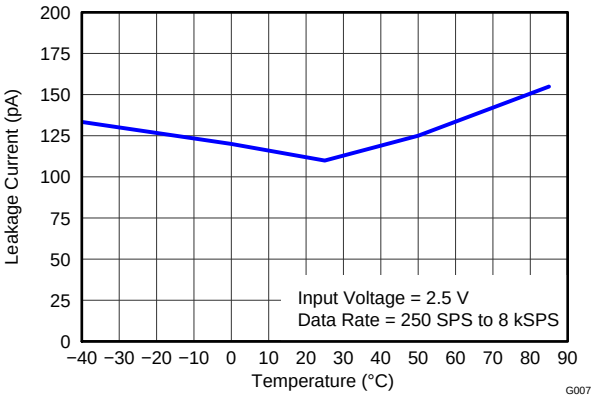


Figure 7. Leakage Current vs Temperature

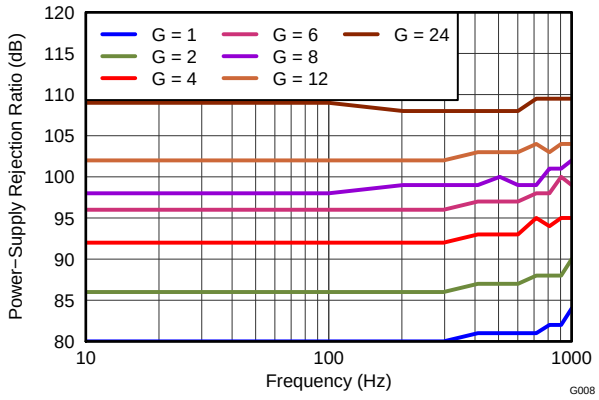


Figure 8. PSRR vs Frequency

Typical Characteristics (continued)

At T_A = 25°C, AVDD = 5 V, AVSS = 0 V, DVDD = 3.3 V, internal VREFP = 4.5 V, VREFN = AVSS, external clock = 2.048 MHz, data rate = 250 SPS, and gain = 12 (unless otherwise noted)

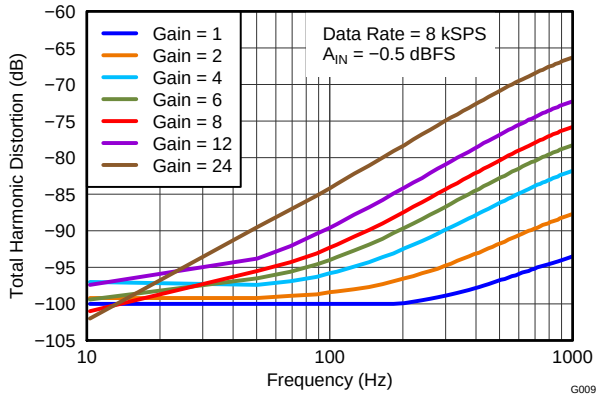


Figure 9. THD vs Frequency

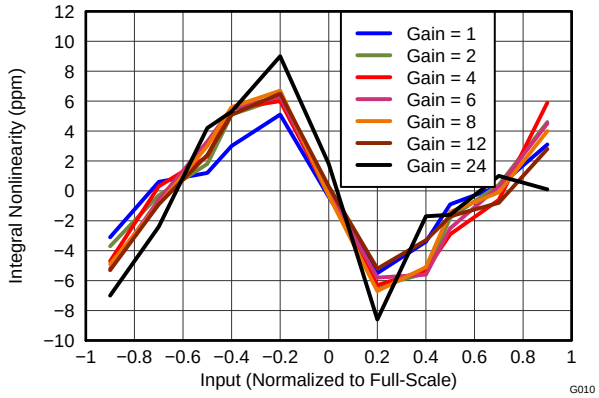


Figure 10. INL vs PGA Gain

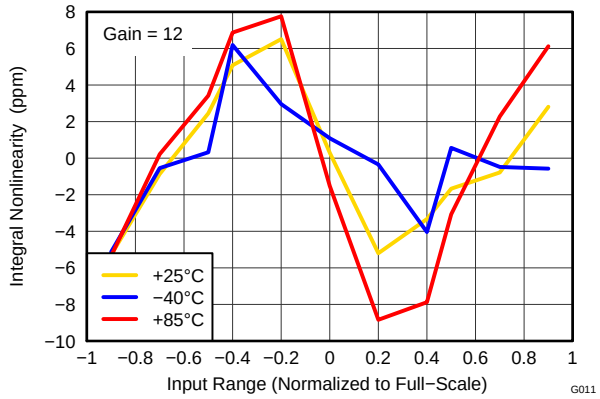


Figure 11. INL vs Temperature

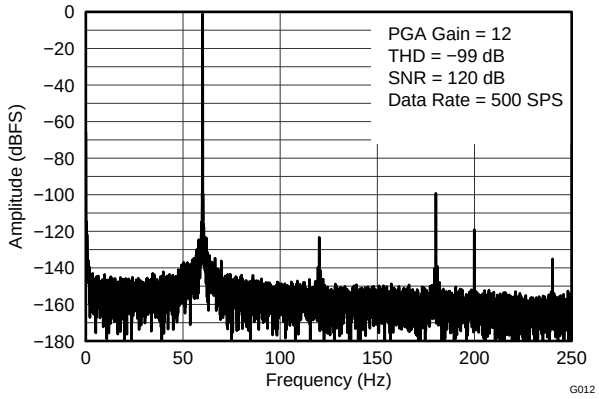


Figure 12. THD FFT Plot (60-Hz Signal)

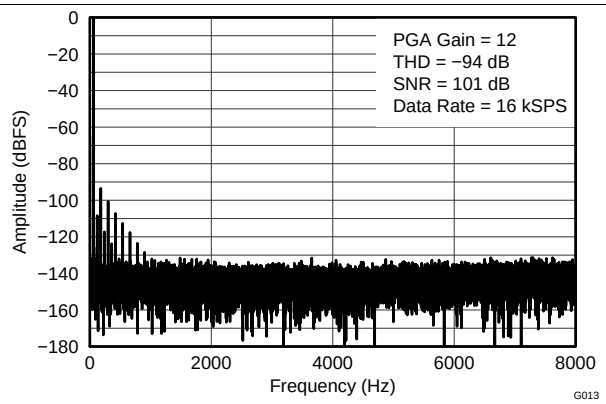


Figure 13. FFT Plot (60-Hz Signal)

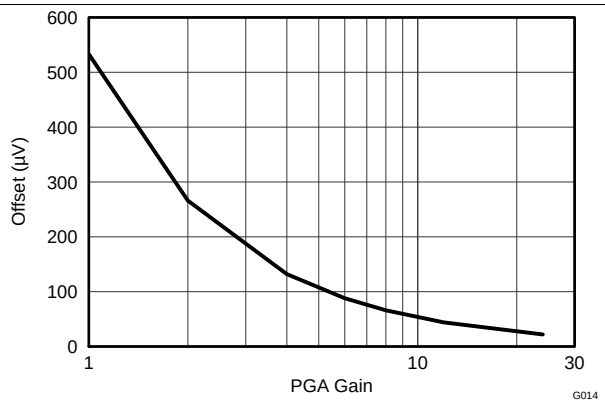


Figure 14. Offset vs PGA Gain (Absolute Value)

Typical Characteristics (continued)

At T_A = 25°C, AVDD = 5 V, AVSS = 0 V, DVDD = 3.3 V, internal VREFP = 4.5 V, VREFN = AVSS, external clock = 2.048 MHz, data rate = 250 SPS, and gain = 12 (unless otherwise noted)

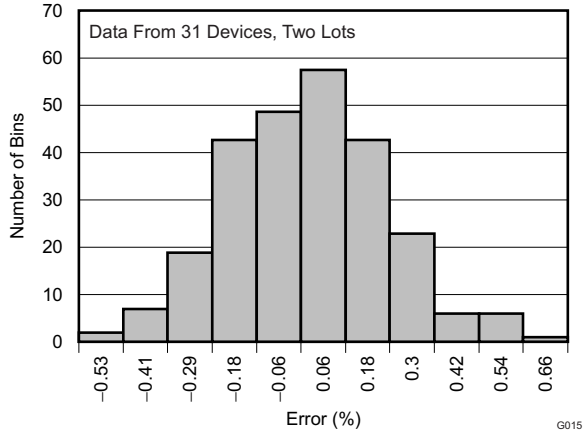


Figure 15. Test Signal Amplitude Accuracy

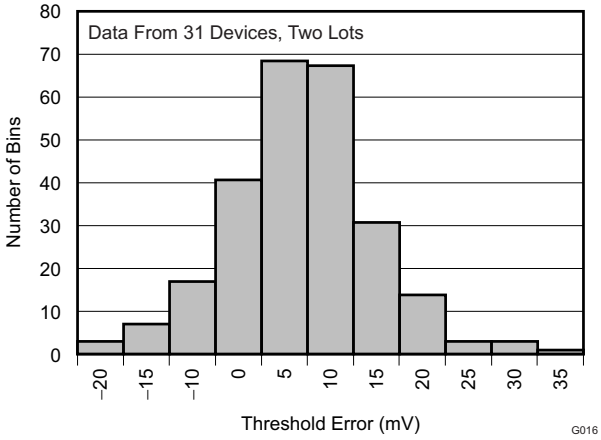


Figure 16. Lead-Off Comparator Threshold Accuracy

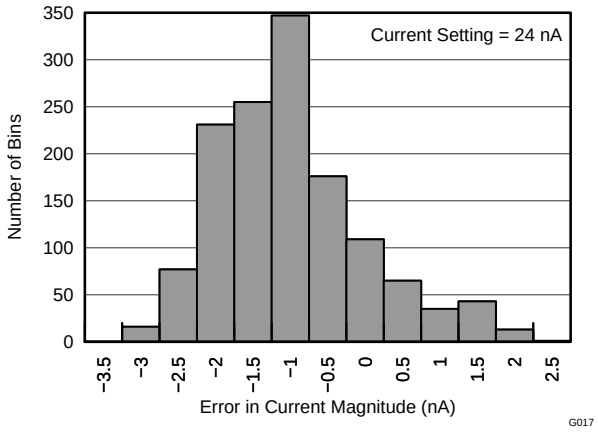


Figure 17. Lead-Off Current Source Accuracy Distribution

8 Parametric Measurement Information

8.1 Noise Measurements

NOTE

Unless otherwise noted, *ADS1299-x* refers to all specifications and functional descriptions of the ADS1299-4, ADS1299-6, and ADS1299.

Optimize the ADS1299-x noise performance by adjusting the data rate and PGA setting. Reduce the data rate to increase the averaging, and the noise drops correspondingly. Increase the PGA value to reduce the input-referred noise. This lowered noise level is particularly useful when measuring low-level biopotential signals. Table 1 to Table 4 summarize the ADS1299-x noise performance with a 5-V analog power supply. The data are representative of typical noise performance at T_A = +25°C. The data shown are the result of averaging the readings from multiple devices and are measured with the inputs shorted together. A minimum of 1000 consecutive readings are used to calculate the RMS and peak-to-peak noise for each reading. For the lower data rates, the ratio is approximately 6.6.

Table 1 shows measurements taken with an internal reference. The data are also representative of the ADS1299-x noise performance when using a low-noise external reference such as the REF5045.

Table 1, Table 2, Table 3, and Table 4 list the input-referred noise in units of μV_{RMS} and μV_{PP} for the conditions shown. The corresponding data in units of effective number of bits (ENOB) where ENOB for the RMS noise is defined as in Equation 1:

$$\text{ENOB} = \log_2\left(\frac{\text{VREF}}{\sqrt{2}\times\text{Gain}\times V_{RMS}}\right) \quad (1)$$

Noise-free bits for the peak-to-peak noise are calculated with the same method.

The dynamic range data in Table 1, Table 2, Table 3, and Table 4 are calculated using Equation 2:

$$\text{Dynamic Range} = 20\times\log\left(\frac{\text{VREF}}{\sqrt{2}\times\text{Gain}\times V_{RMS}}\right) \quad (2)$$

Table 1. Input-Referred Noise (μV_{RMS}, μV_{PP}) in Normal Mode 5-V Analog Supply and 4.5-V Reference(1)

DR BITS OF CONFIG1 REGISTER	OUTPUT DATA RATE (SPS)	–3-dB BANDWIDTH (Hz)	PGA GAIN = 1					PGA GAIN = 2				
			μV_{RMS}	μV_{PP}	DYNAMIC RANGE (dB)	NOISE-FREE BITS	ENOB	μV_{RMS}	μV_{PP}	DYNAMIC RANGE (dB)	NOISE-FREE BITS	ENOB
000	16000	4193	21.70	151.89	103.3	15.85	17.16	10.85	75.94	103.3	15.85	17.16
001	8000	2096	6.93	48.53	113.2	17.50	18.81	3.65	25.52	112.8	17.43	18.74
010	4000	1048	4.33	30.34	117.3	18.18	19.49	2.28	15.95	116.9	18.11	19.41
011	2000	524	3.06	21.45	120.3	18.68	19.99	1.61	11.29	119.9	18.60	19.91
100	1000	262	2.17	15.17	123.3	19.18	20.49	1.14	7.98	122.9	19.10	20.41
101	500	131	1.53	10.73	126.3	19.68	20.99	0.81	5.65	125.9	19.60	20.91
110	250	65	1.08	7.59	129.3	20.18	21.48	0.57	3.99	128.9	20.10	21.41
111	n/a	n/a	—	—	—	—	—	—	—	—	—	—

(1) At least 1000 consecutive readings were used to calculate the RMS and peak-to-peak noise values in this table.

Table 2. Input-Referred Noise (μV_{RMS}, μV_{PP}) in Normal Mode 5-V Analog Supply and 4.5-V Reference(1)

DR BITS OF CONFIG1 REGISTER	OUTPUT DATA RATE (SPS)	–3-dB BANDWIDTH (Hz)	PGA GAIN = 4					PGA GAIN = 6				
			μV_{RMS}	μV_{PP}	DYNAMIC RANGE (dB)	NOISE-FREE BITS	ENOB	μV_{RMS}	μV_{PP}	DYNAMIC RANGE (dB)	NOISE-FREE BITS	ENOB
000	16000	4193	5.60	39.23	103.0	15.81	17.12	3.87	27.10	102.7	15.76	17.06
001	8000	2096	1.98	13.87	112.1	17.31	18.62	1.31	9.19	112.1	17.32	18.62
010	4000	1048	1.24	8.66	116.1	17.99	19.29	0.93	6.50	115.1	17.82	19.12
011	2000	524	0.88	6.13	119.2	18.49	19.79	0.66	4.60	118.1	18.32	19.62
100	1000	262	0.62	4.34	122.2	18.99	20.29	0.46	3.25	121.1	18.81	20.12
101	500	131	0.44	3.07	125.2	19.49	20.79	0.33	2.30	124.1	19.31	20.62
110	250	65	0.31	2.16	128.2	19.99	21.30	0.23	1.62	127.2	19.82	21.13
111	n/a	n/a	—	—	—	—	—	—	—	—	—	—

(1) At least 1000 consecutive readings were used to calculate the RMS and peak-to-peak noise values in this table.

Table 3. Input-Referred Noise (μV_{RMS}, μV_{PP}) in Normal Mode 5-V Analog Supply and 4.5-V Reference(1)

DR BITS OF CONFIG1 REGISTER	OUTPUT DATA RATE (SPS)	–3-dB BANDWIDTH (Hz)	PGA GAIN = 8					PGA GAIN = 12				
			μV_{RMS}	μV_{PP}	DYNAMIC RANGE (dB)	NOISE-FREE BITS	ENOB	μV_{RMS}	μV_{PP}	DYNAMIC RANGE (dB)	NOISE-FREE BITS	ENOB
000	16000	4193	3.05	21.32	102.3	15.69	16.99	2.27	15.89	101.3	15.53	16.83
001	8000	2096	1.11	7.80	111.0	17.14	18.45	0.92	6.41	109.2	16.84	18.14
010	4000	1048	0.79	5.52	114.0	17.64	18.95	0.65	4.53	112.2	17.34	18.64
011	2000	524	0.56	3.90	117.1	18.14	19.44	0.46	3.20	115.2	17.84	19.14
100	1000	262	0.39	2.76	120.1	18.64	19.94	0.32	2.26	118.3	18.34	19.65
101	500	131	0.28	1.95	123.1	19.14	20.44	0.23	1.61	121.2	18.83	20.14
110	250	65	0.20	1.38	126.1	19.64	20.95	0.16	1.13	124.3	19.34	20.65
111	n/a	n/a	—	—	—	—	—	—	—	—	—	—

(1) At least 1000 consecutive readings were used to calculate the RMS and peak-to-peak noise values in this table.

Table 4. Input-Referred Noise (μV_{RMS}, μV_{PP}) in Normal Mode 5-V Analog Supply and 4.5-V Reference(1)

DR BITS OF CONFIG1 REGISTER	OUTPUT DATA RATE (SPS)	–3-dB BANDWIDTH (Hz)	PGA GAIN = 24				
			μV_{RMS}	μV_{PP}	DYNAMIC RANGE (dB)	NOISE-FREE BITS	ENOB
000	16000	4193	1.66	11.64	98.0	14.98	16.28
001	8000	2096	0.80	5.57	104.4	16.04	17.35
010	4000	1048	0.56	3.94	107.4	16.54	17.84
011	2000	524	0.40	2.79	110.4	17.04	18.35
100	1000	262	0.28	1.97	113.5	17.54	18.85
101	500	131	0.20	1.39	116.5	18.04	19.35
110	250	65	0.14	0.98	119.5	18.54	19.85
111	n/a	n/a	—	—	—	—	—

(1) At least 1000 consecutive readings were used to calculate the RMS and peak-to-peak noise values in this table.

9 Detailed Description

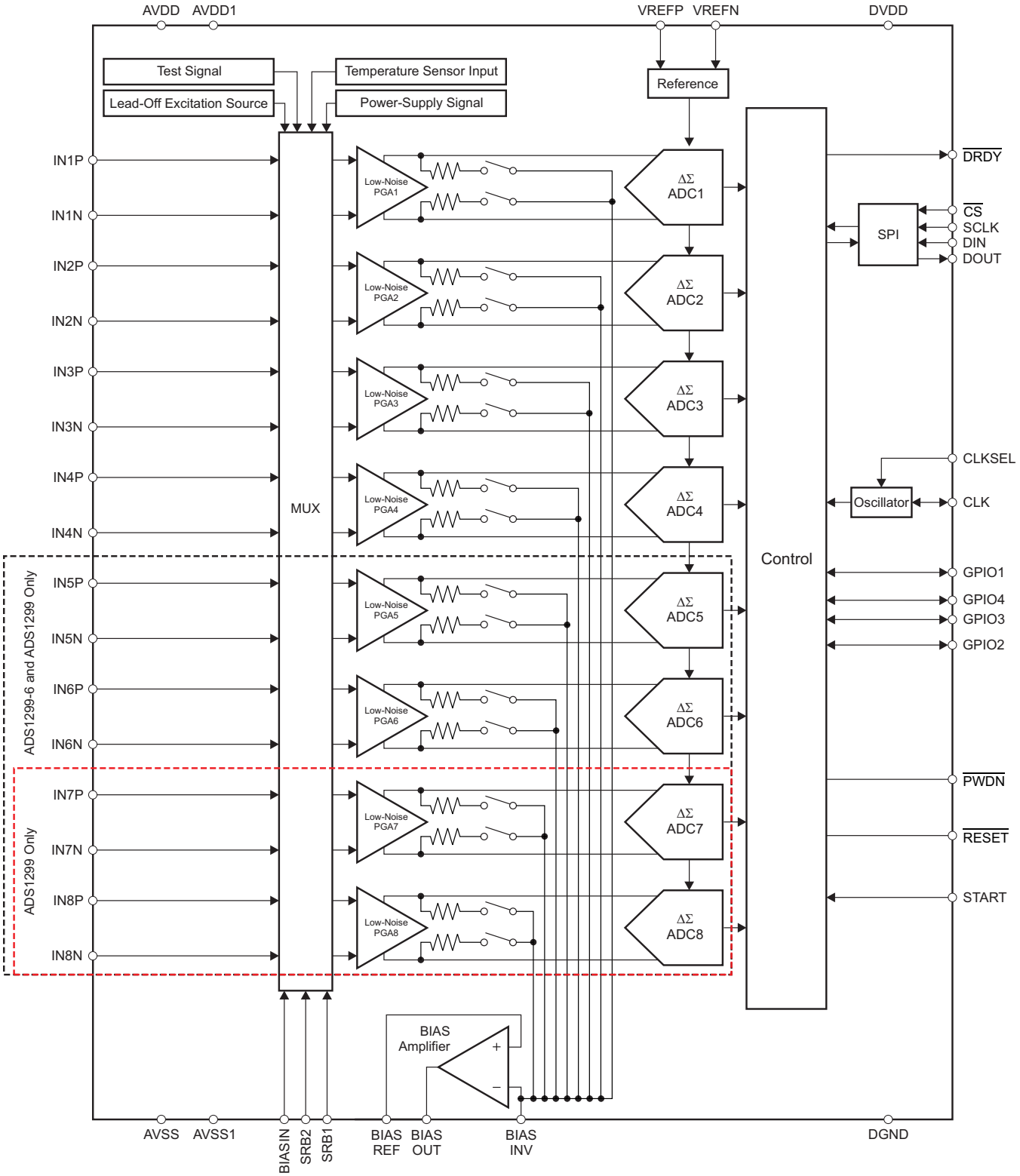
9.1 Overview

The ADS1299-x is a low-noise, low-power, multichannel, simultaneously-sampling, 24-bit, delta-sigma ($\Delta\Sigma$) analog-to-digital converter (ADC) with an integrated programmable gain amplifier (PGA). These devices integrate various EEG-specific functions that makes the family well-suited for scalable electrocardiogram (ECG), electroencephalography (EEG) applications. These devices can also be used in high-performance, multichannel, data acquisition systems by powering down the ECG or EEG-specific circuitry.

The devices have a highly-programmable multiplexer that allows for temperature, supply, input short, and bias measurements. Additionally, the multiplexer allows any input electrodes to be programmed as the patient reference drive. The PGA gain can be chosen from one of seven settings (1, 2, 4, 6, 8, 12, and 24). The ADCs in the device offer data rates from 250 SPS to 16 kSPS. Communication to the device is accomplished using an SPI-compatible interface. The device provides four general-purpose input/output (GPIO) pins for general use. Multiple devices can be synchronized using the START pin.

The internal reference generates a low noise 4.5 V internal voltage when enabled and the internal oscillator generates a 2.048-MHz clock when enabled. The versatile patient bias drive block allows the average of any electrode combination to be chosen in order to generate the patient drive signal. Lead-off detection can be accomplished by using a current source or sink. A one-time, in-band, lead-off option and a continuous, out-of-band, internal lead-off option are available.

9.2 Functional Block Diagram



9.3 Feature Description

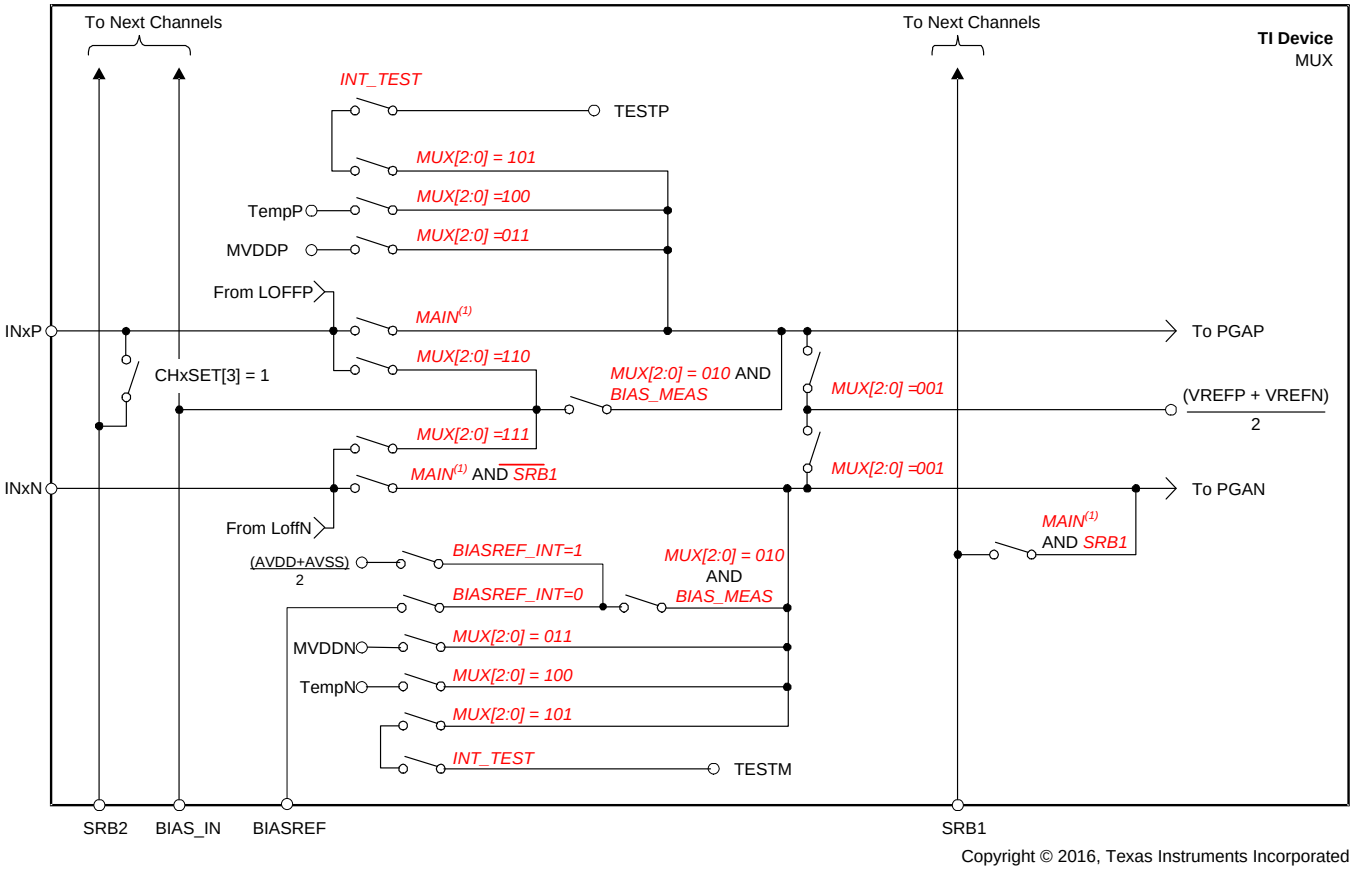
This section contains details of the ADS1299-x internal functional elements. The analog blocks are discussed first, followed by the digital interface. Blocks implementing EEG-specific functions are covered at the end of this section.

Throughout this document, f_{CLK} denotes the CLK pin signal frequency, t_{CLK} denotes the CLK pin signal period, f_{DR} denotes the output data rate, t_{DR} denotes the output data time period, and f_{MOD} denotes the frequency at which the modulator samples the input.

9.3.1 Analog Functionality

9.3.1.1 Input Multiplexer

The ADS1299-x input multiplexers are very flexible and provide many configurable signal-switching options. Figure 18 shows the multiplexer on a single channel of the device. Note that the device has either four (ADS1299-4), six (ADS1299-6) or eight (ADS1299) such blocks, one for each channel. SRB1, SRB2, and BIASIN are common to all blocks. INxP and INxN are separate for each of the four, six, or eight blocks. This flexibility allows for significant device and sub-system diagnostics, calibration, and configuration. Switch setting selections for each channel by writing the appropriate values to the CHnSET[3:0] register (see the *CHnSET: Individual Channel Settings* section for details) using the BIAS_MEAS bit in the CONFIG3 register and the SRB1 bit in the MISC1 register (see the *CONFIG3: Configuration Register 3* subsection of the *Register Maps* section for details). See the *Input Multiplexer* section for further information regarding the EEG-specific features of the multiplexer.



(1) MAIN is equal to either MUX[2:0] = 000, MUX[2:0] = 110, or MUX[2:0] = 111.

Figure 18. Input Multiplexer Block for One Channel

Feature Description (continued)

9.3.1.1.1 Device Noise Measurements

Setting CHnSET[2:0] = 001 sets the common-mode voltage of [(V_{VREFP} + V_{VREFN}) / 2] to both channel inputs. This setting can be used to test inherent device noise in the user system.

9.3.1.1.2 Test Signals (TestP and TestN)

Setting CHnSET[2:0] = 101 provides internally-generated test signals for use in sub-system verification at power-up. This functionality allows the device internal signal chain to be tested out.

Test signals are controlled through register settings (see the *CONFIG2: Configuration Register 2* subsection in the *Register Maps* section for details). TEST_AMP controls the signal amplitude and TEST_FREQ controls switching at the required frequency.

9.3.1.1.3 Temperature Sensor (TempP, TempN)

The ADS1299-x contains an on-chip temperature sensor. This sensor uses two internal diodes with one diode having a current density 16x that of the other, as shown in Figure 19. The difference in diode current densities yields a voltage difference proportional to absolute temperature.

As a result of the low thermal resistance of the package to the printed circuit board (PCB), the internal device temperature tracks PCB temperature closely. Note that self-heating of the ADS1299-x causes a higher reading than the temperature of the surrounding PCB.

The scale factor of Equation 3 converts the temperature reading to degrees Celsius. Before using this equation, the temperature reading code must first be scaled to microvolts.

$$\text{Temperature (°C)} = \left(\frac{\text{Temperature Reading } (\mu\text{V}) - 145{,}300\ \mu\text{V}}{490\ \mu\text{V/°C}} \right) + 25\text{°C} \tag{3}$$

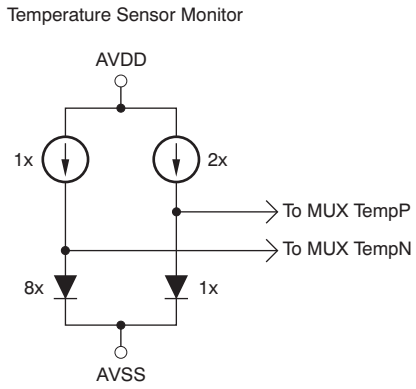


Figure 19. Temperature Sensor Measurement in the Input

9.3.1.1.4 Supply Measurements (MVDDP, MVDDN)

Setting CHnSET[2:0] = 011 sets the channel inputs to different supply voltages of the device. For channels 1, 2, 5, 6, 7, and 8, (MVDDP – MVDDN) is [0.5 × (AVDD + AVSS)].

For channels 3 and 4, (MVDDP – MVDDN) is DVDD / 4.

To avoid saturating the PGA when measuring power supplies, set the gain to 1.

9.3.1.1.5 Lead-Off Excitation Signals (LoffP, LoffN)

The lead-off excitation signals are fed into the multiplexer before the switches. The comparators that detect the lead-off condition are also connected to the multiplexer block before the switches. For a detailed description of the lead-off block, see the *Lead-Off Detection* section.

Feature Description (continued)

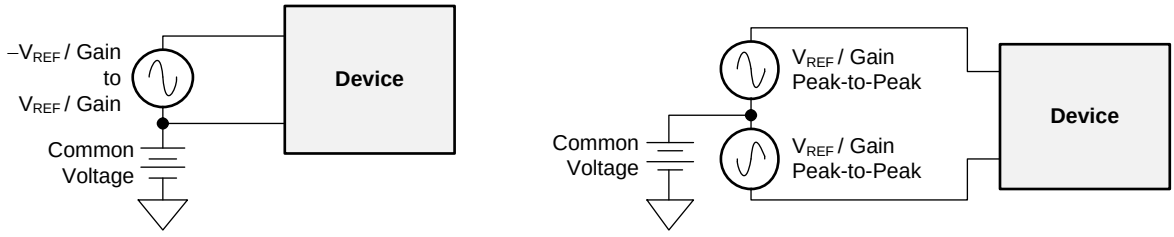
9.3.1.1.6 Auxiliary Single-Ended Input

The BIASIN pin is primarily used for routing the bias signal to any electrodes in case the bias electrode falls off. However, the BIASIN pin can be used as a multiple single-ended input channel. The signal at the BIASIN pin can be measured with respect to the voltage at the BIASREF pin using any of the eight channels. This measurement is done by setting the channel multiplexer setting to '010' and the BIAS_MEAS bit of the CONFIG3 register to '1'.

9.3.1.2 Analog Input

The analog inputs to the device connect directly to an integrated low-noise, low-drift, high input impedance, programmable gain amplifier. The amplifier is located following the individual channel multiplexer.

The ADS1299-x analog inputs are fully differential. The differential input voltage (V_{INxP} – V_{INxN}) can span from $–V_{REF}$ / gain to V_{REF} / gain. See the *Data Format* section for an explanation of the correlation between the analog input and digital codes. There are two general methods of driving the ADS1299-x analog inputs: pseudo-differential or fully-differential, as shown in Figure 20, Figure 21, and Figure 22.



a) Psuedo-Differential Input

b) Differential Input

Figure 20. Methods of Driving the ADS1299-x: Pseudo-Differential or Fully Differential

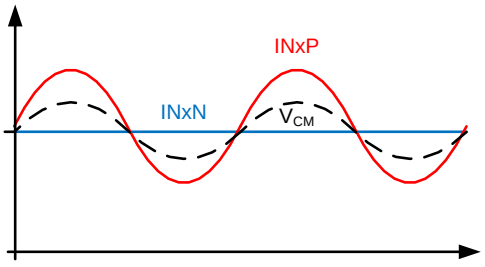


Figure 21. Pseudo-Differential Input Mode

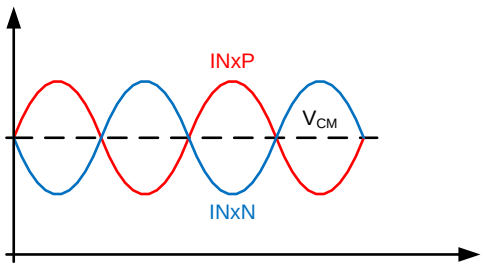


Figure 22. Fully-Differential Input Mode

Hold the INxN pin at a common voltage, preferably at mid supply, to configure the fully differential input for a pseudo-differential signal. Swing the INxP pin around the common voltage $–V_{REF}$ / gain to V_{REF} / gain and remain within the absolute maximum specifications. The common-mode voltage (V_{CM}) changes with varying signal level when the inputs are configured in pseudo-differential mode. Verify that the differential signal at the minimum and maximum points meets the common-mode input specification discussed in the *Input Common-Mode Range* section.

Configure the signals at INxP and INxN to be 180° out-of-phase centered around a common voltage to use a fully differential input method. Both the INxP and INxN inputs swing from the common voltage + ½ V_{REF} / gain to the common voltage – ½ V_{REF} / gain. The differential voltage at the maximum and minimum points is equal to $–V_{REF}$ / gain to V_{REF} / gain and centered around a fixed common-mode voltage (V_{CM}). Use the ADS1299-x in a differential configuration to maximize the dynamic range of the data converter. For optimal performance, the common voltage is recommended to be set at the midpoint of the analog supplies [(AVDD + AVSS) / 2].

9.3.1.3 PGA Settings and Input Range

The low-noise PGA is a differential input and output amplifier, as shown in Figure 23. The PGA has seven gain settings (1, 2, 4, 6, 8, 12, and 24) that can be set by writing to the CHnSET register (see the *CHnSET: Individual Channel Settings* subsection of the *Register Maps* section for details). The ADS1299-x has CMOS inputs and therefore has negligible current noise. Table 5 shows the typical bandwidth values for various gain settings. Note that Table 5 shows small-signal bandwidth. For large signals, performance is limited by PGA slew rate.

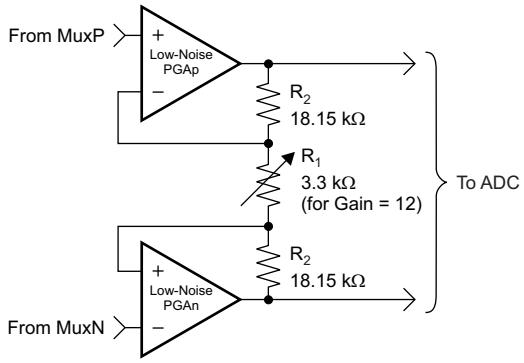


Figure 23. PGA Implementation

Table 5. PGA Gain versus Bandwidth

GAIN	NOMINAL BANDWIDTH AT ROOM TEMPERATURE (kHz)
1	662
2	332
4	165
6	110
8	83
12	55
24	27

The PGA resistor string that implements the gain has 39.6 kΩ of resistance for a gain of 12. This resistance provides a current path across the PGA outputs in the presence of a differential input signal. This current is in addition to the quiescent current specified for the device in the presence of a differential signal at the input.

9.3.1.3.1 Input Common-Mode Range

To stay within the linear operating range of the PGA, the input signals must meet certain requirements that are discussed in this section.

The outputs of the amplifiers in Figure 23 cannot swing closer to the supplies (AVSS and AVDD) than 200 mV. If the outputs of the amplifiers are driven to within 200 mV of the supply rails, then the amplifiers saturate and consequently become nonlinear. To prevent this nonlinear operating condition, the output voltages must not exceed the common-mode range of the front-end.

The usable input common-mode range of the front-end depends on various parameters, including the maximum differential input signal, supply voltage, PGA gain, and the 200 mV for the amplifier headroom. This range is described in Equation 4:

$$AVDD - 0.2\ V - \left(\frac{Gain \times V_{MAX\_DIFF}}{2}\right) > CM > AVSS + 0.2\ V + \left(\frac{Gain \times V_{MAX\_DIFF}}{2}\right) \quad (4)$$

where:

V_{MAX_DIFF} = maximum differential signal at the PGA input

CM = common-mode range

For example:

If AVDD = 5 V, gain = 12, and V_{MAX_DIFF} = 350 mV

Then 2.3 V < CM < 2.7 V

9.3.1.3.2 Input Differential Dynamic Range

The differential input voltage range (V_{INxP} – V_{INxN}) depends on the analog supply and reference used in the system. This range is shown in Equation 5.

$$\text{Full-Scale Range} = \frac{\pm V_{REF}}{\text{Gain}} = \frac{2V_{REF}}{\text{Gain}} \quad (5)$$

9.3.1.3.3 ADC ΔΣ Modulator

Each ADS1299-x channel has a 24-bit, ΔΣ ADC. This converter uses a second-order modulator optimized for low-noise applications. The modulator samples the input signal at the rate of (f_{MOD} = f_{CLK} / 2). As in the case of any ΔΣ modulator, the device noise is shaped until f_{MOD} / 2, as shown in Figure 24. The on-chip digital decimation filters explained in the next section can be used to filter out the noise at higher frequencies. These on-chip decimation filters also provide antialias filtering. This ΔΣ converter feature drastically reduces the complexity of the analog antialiasing filters typically required with nyquist ADCs.

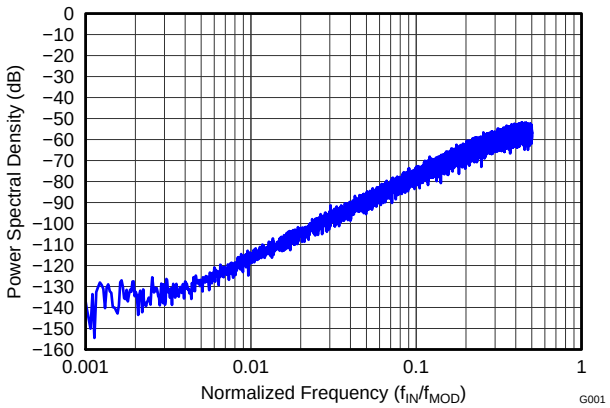
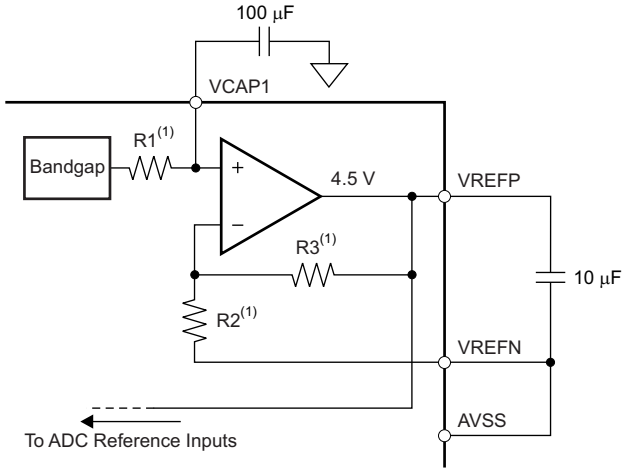


Figure 24. Modulator Noise Spectrum Up To 0.5 × f_{MOD}

9.3.1.3.4 Reference

Figure 25 shows a simplified block diagram of the ADS1299-x internal reference. The 4.5-V reference voltage is generated with respect to AVSS. When using the internal voltage reference, connect V_{REFN} to AVSS.



(1) For V_{REF} = 4.5 V: R1 = 9.8 kΩ, R2 = 13.4 kΩ, and R3 = 36.85 kΩ.

Figure 25. Internal Reference

The external band-limiting capacitors determine the amount of reference noise contribution. For high-end EEG systems, the capacitor values should be chosen such that the bandwidth is limited to less than 10 Hz so that the reference noise does not dominate system noise.

Alternatively, the internal reference buffer can be powered down and an external reference can be applied to VREFP. Figure 26 shows a typical external reference drive circuitry. Power-down is controlled by the PD_REFBUF bit in the CONFIG3 register. This power-down is also used to share internal references when two devices are cascaded. By default, the device wakes up in external reference mode.

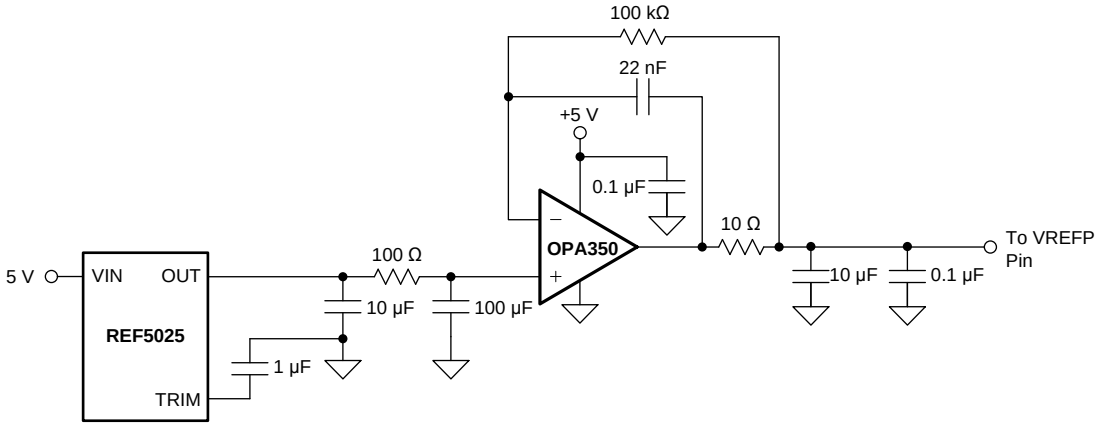


Figure 26. External Reference Driver

9.3.2 Digital Functionality

9.3.2.1 Digital Decimation Filter

The digital filter receives the modulator output and decimates the data stream. By adjusting the amount of filtering, tradeoffs can be made between resolution and data rate: filter more for higher resolution, filter less for higher data rates. Higher data rates are typically used in EEG applications for ac lead-off detection.

The digital filter on each channel consists of a third-order sinc filter. The sinc filter decimation ratio can be adjusted by the DR bits in the CONFIG1 register (see the *Register Maps* section for details). This setting is a global setting that affects all channels and, therefore, all channels operate at the same data rate in a device.

9.3.2.1.1 Sinc Filter Stage (sinx / x)

The sinc filter is a variable decimation rate, third-order, low-pass filter. Data are supplied to this section of the filter from the modulator at the rate of f_{MOD}. The sinc filter attenuates the modulator high-frequency noise, then decimates the data stream into parallel data. The decimation rate affects the overall converter data rate.

Equation 6 shows the scaled Z-domain transfer function of the sinc filter.

$$\left|H(z)\right| = \left|\frac{1 - Z^{-N}}{1 - Z^{-1}}\right|^{3} \tag{6}$$

The frequency domain transfer function of the sinc filter is shown in Equation 7.

$$\left|H(f)\right| = \left|\frac{\sin\left(\frac{N\pi f}{f_{MOD}}\right)}{N \times \sin\left(\frac{\pi f}{f_{MOD}}\right)}\right|^{3} \tag{7}$$

where:

N = decimation ratio

The sinc filter has notches (or zeroes) that occur at the output data rate and multiples thereof. At these frequencies, the filter has infinite attenuation. Figure 27 shows the sinc filter frequency response and Figure 28 shows the sinc filter roll-off. With a step change at input, the filter takes 3 × t_{DR} to settle. After a rising edge of the START signal, the filter takes t_{SETTLE} time to give the first data output. The settling time of the filters at various data rates are discussed in the *Start* subsection of the *SPI Interface* section. Figure 29 and Figure 30 show the filter transfer function until f_{MOD} / 2 and f_{MOD} / 16, respectively, at different data rates. Figure 31 shows the transfer function extended until 4 × f_{MOD}. The ADS1299-x pass band repeats itself at every f_{MOD}. The input R-C antialiasing filters in the system should be chosen such that any interference in frequencies around multiples of f_{MOD} are attenuated sufficiently.

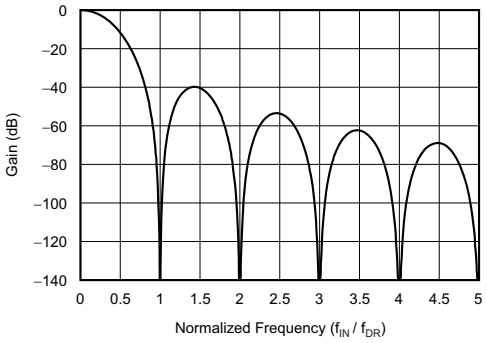


Figure 27. Sinc Filter Frequency Response

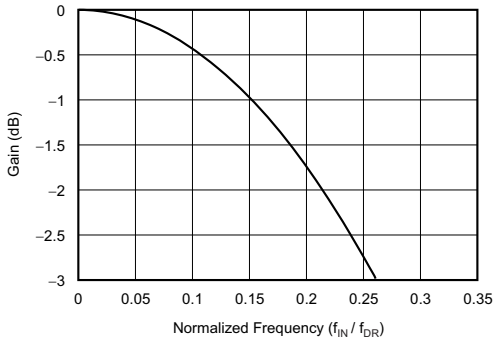


Figure 28. Sinc Filter Roll-Off

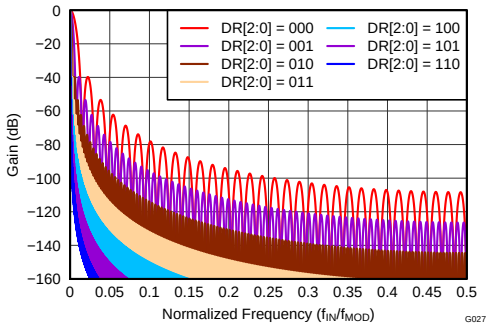


Figure 29. Transfer Function of On-Chip Decimation Filters Until f_{MOD} / 2

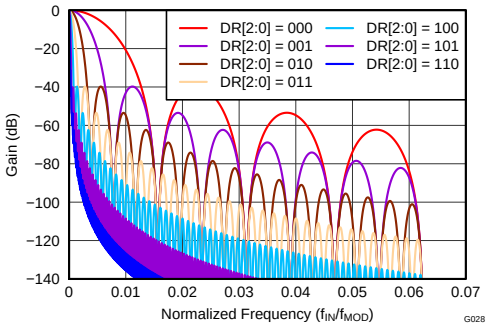


Figure 30. Transfer Function of On-Chip Decimation Filters Until f_{MOD} / 16

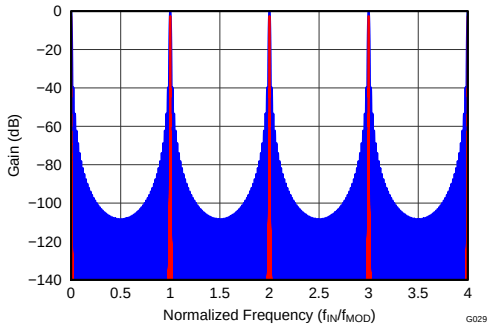


Figure 31. Transfer Function of On-Chip Decimation Filters Until 4 f_{MOD} for DR[2:0] = 000 and DR[2:0] = 110

9.3.2.2 Clock

The ADS1299-x provides two methods for device clocking: internal and external. Internal clocking is ideally suited for low-power, battery-powered systems. The internal oscillator is trimmed for accuracy at room temperature. Accuracy varies over the specified temperature range; see the *Electrical Characteristics*. Clock selection is controlled by the CLKSEL pin and the CLK_EN register bit.

The CLKSEL pin selects either the internal or external clock. The CLK_EN bit in the CONFIG1 register enables and disables the oscillator clock to be output in the CLK pin. A truth table for these two pins is shown in Table 6. The CLK_EN bit is useful when multiple devices are used in a daisy-chain configuration. During power-down, the external clock is recommended be shut down to save power.

Table 6. CLKSEL Pin and CLK_EN Bit

CLKSEL PIN	CONFIG1.CLK_EN BIT	CLOCK SOURCE	CLK PIN STATUS
0	X	External clock	Input: external clock
1	0	Internal clock oscillator	3-state
1	1	Internal clock oscillator	Output: internal clock oscillator

9.3.2.3 GPIO

The ADS1299-x has a total of four general-purpose digital I/O (GPIO) pins available in normal mode of operation. The digital I/O pins are individually configurable as either inputs or outputs through the GPIOC bits register. The GPIOD bits in the GPIO register control the pin level. When reading the GPIOD bits, the data returned are the logic level of the pins, whether they are programmed as inputs or outputs. When the GPIO pin is configured as an input, a write to the corresponding GPIOD bit has no effect. When configured as an output, a write to the GPIOD bit sets the output value.

If configured as inputs, these pins must be driven (do not float). The GPIO pins are set as inputs after power-on or after a reset. Figure 32 shows the GPIO port structure. The pins should be shorted to DGND if not used.

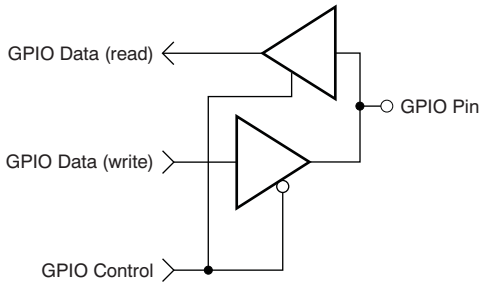
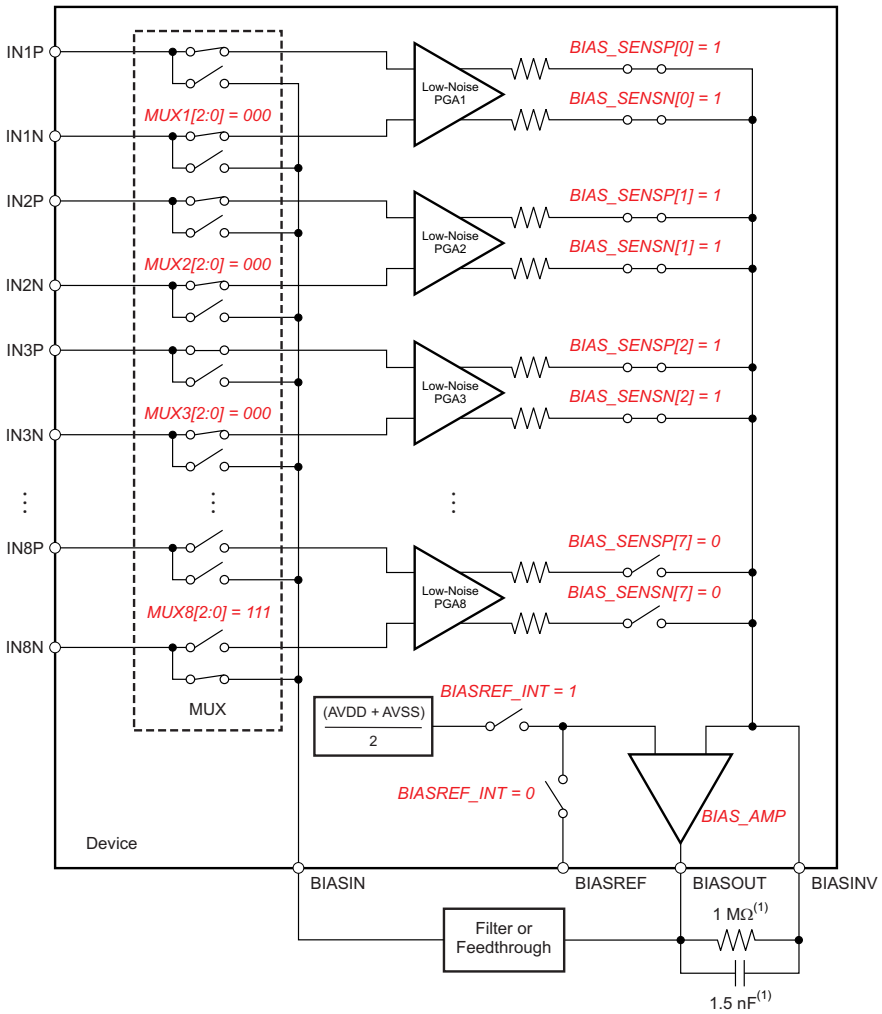


Figure 32. GPIO Port Pin

9.3.2.4 ECG and EEG Specific Features

9.3.2.4.1 Input Multiplexer (Rerouting the BIAS Drive Signal)

The input multiplexer has EEG-specific functions for the bias drive signal. The BIAS signal is available at the BIASOUT pin when the appropriate channels are selected for BIAS derivation, feedback elements are installed external to the chip, and the loop is closed. This signal can either be fed after filtering or fed directly into the BIASIN pin, as shown in Figure 33. This BIASIN signal can be multiplexed into any input electrode by setting the MUX bits of the appropriate channel set registers to '110' for P-side or '111' for N-side. Figure 33 shows the BIAS signal generated from channels 1, 2, and 3 and routed to the N-side of channel 8. This feature can be used to dynamically change the electrode that is used as the reference signal to drive the patient body.

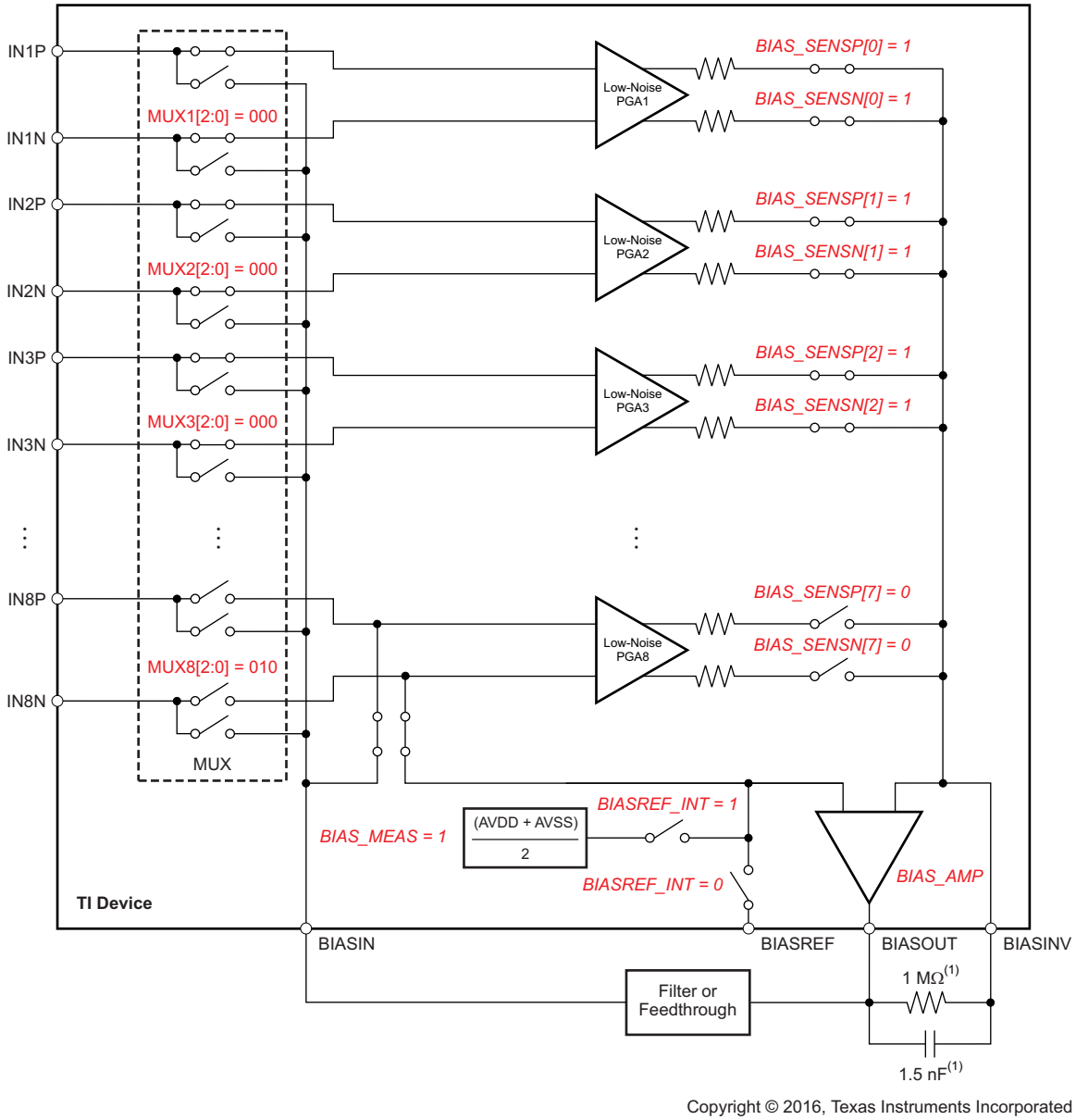


(1) Typical values for example only.

Figure 33. Example of BIASOUT Signal Configured to be Routed to IN8N

9.3.2.4.2 Input Multiplexer (Measuring the BIAS Drive Signal)

Also, the BIASOUT signal can be routed to a channel (that is not used for the calculation of BIAS) for measurement. Figure 34 shows the register settings to route the BIASIN signal to channel 8. The measurement is done with respect to the voltage on the BIASREF pin. If BIASREF is chosen to be internal, then BIASREF is at [(AVDD + AVSS) / 2]. This feature is useful for debugging purposes during product development.



(1) Typical values for example only.

Figure 34. BIASOUT Signal Configured to be Read Back by Channel 8

9.3.2.4.3 Lead-Off Detection

Patient electrode impedances are known to decay over time. These electrode connections must be continuously monitored to verify that a suitable connection is present. The ADS1299-x lead-off detection functional block provides significant flexibility to the user to choose from various lead-off detection strategies. Though called lead-off detection, this is in fact an *electrode-off* detection.

The basic principle is to inject an excitation current and measure the voltage to determine if the electrode is off. As shown in the lead-off detection functional block diagram in Figure 35, this circuit provides two different methods of determining the state of the patient electrode. The methods differ in the frequency content of the excitation signal. Lead-off can be selectively done on a per channel basis using the LOFF_SENSP and LOFF_SENSN registers. Also, the internal excitation circuitry can be disabled and just the sensing circuitry can be enabled.

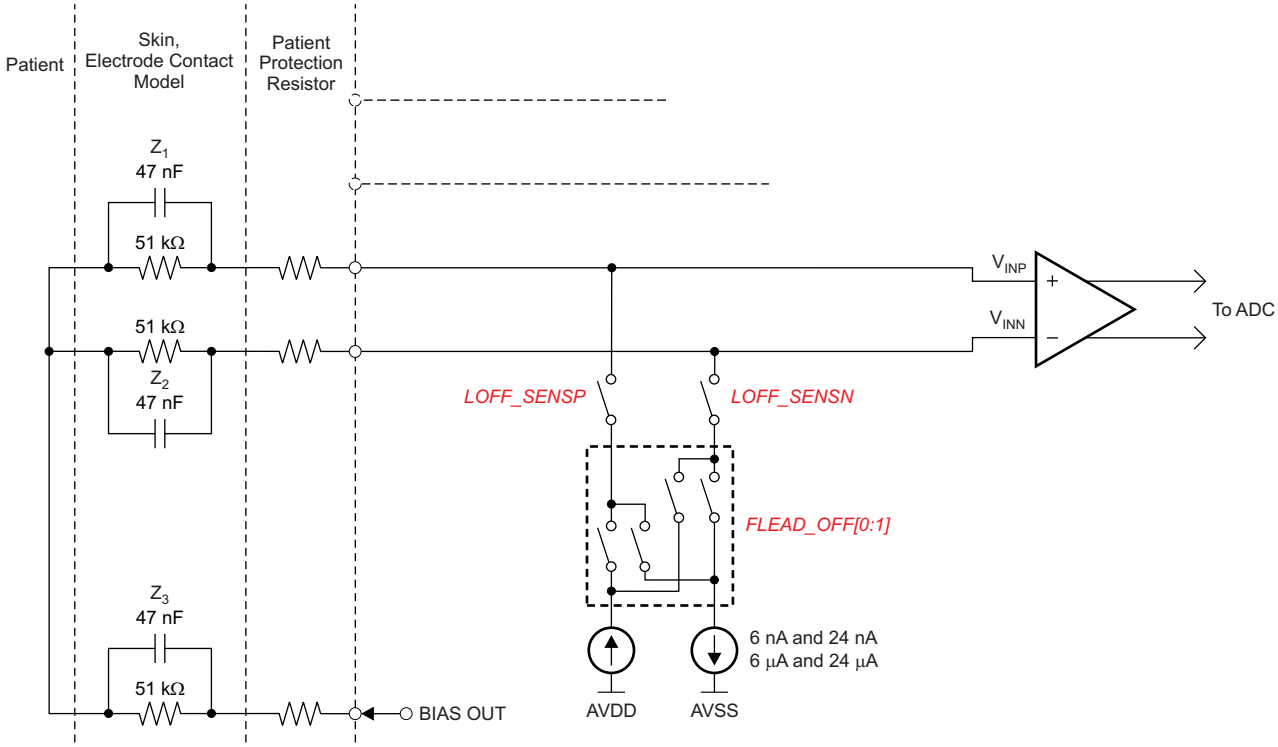


Figure 35. Lead-Off Detection

9.3.2.4.3.1 DC Lead-Off

In this method, the lead-off excitation is with a dc signal. The dc excitation signal can be chosen from either an external pull-up or pull-down resistor or an internal current source or sink, as shown in Figure 36. One side of the channel is pulled to supply and the other side is pulled to ground. The pull-up and pull-down current can be swapped (as shown in Figure 36b and Figure 36c) by setting the bits in the LOFF_FLIP register. In case of a current source or sink, the magnitude of the current can be set by using the ILEAD_OFF[1:0] bits in the LOFF register. The current source or sink gives larger input impedance compared to the 10-MΩ pull-up or pull-down resistor.

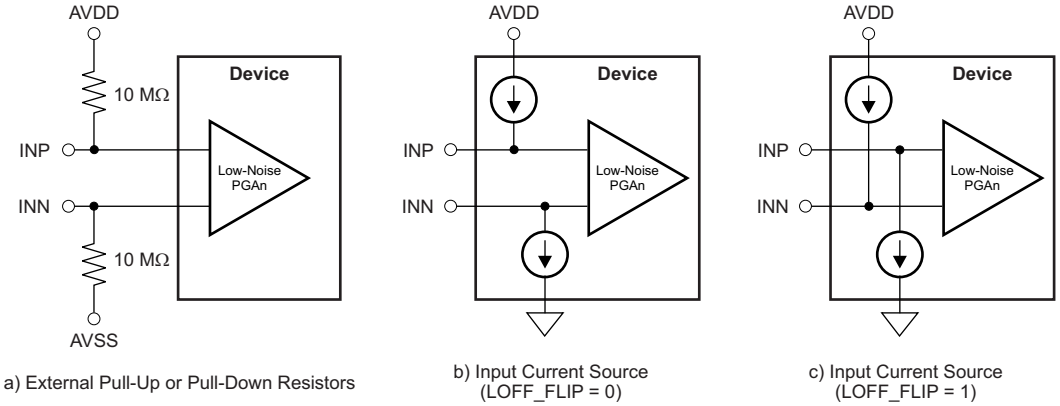


Figure 36. DC Lead-Off Excitation Options

Sensing of the response can be done either by searching the digital output code from the device or by monitoring the input voltages with an on-chip comparator. If either electrode is off, the pull-up and pull-down resistors saturate the channel. Searching the output code determines if either the P-side or the N-side is off. To pinpoint which one is off, the comparators must be used. The input voltage is also monitored using a comparator and a 3-bit DAC whose levels are set by the COMP_TH[2:0] bits in the LOFF register. The output of the comparators are stored in the LOFF_STATP and LOFF_STATN registers. These registers are available as a part of the output data stream. (See the *Data Output (DOUT)* subsection of the *SPI Interface* section.) If dc lead-off is not used, the lead-off comparators can be powered down by setting the PD_LOFF_COMP bit in the CONFIG4 register.

An example procedure to turn on dc lead-off is given in the *Lead-Off* section.

9.3.2.4.3.2 AC Lead-Off (One Time or Periodic)

In this method, an in-band ac signal is used for excitation. The ac signal is generated by alternatively providing a current source and sink at the input with a fixed frequency. The frequency can be chosen by the FLEAD_OFF[1:0] bits in the LOFF register. The excitation frequency is chosen to be one of the two in-band frequency selections (7.8 Hz or 31.2 Hz). This in-band excitation signal is passed through the channel and measured at the output.

Sensing of the ac signal is done by passing the signal through the channel to be digitized and then measured at the output. The ac excitation signals are introduced at a frequency that is in the band of interest. The signal can be filtered out separately and processed. By measuring the magnitude of the output at the excitation signal frequency, the electrode impedance can be calculated.

For continuous lead-off, an out-of-band ac current source or sink must be externally applied to the inputs. This signal can then be digitally processed to determine the electrode impedance.

9.3.2.4.4 Bias Lead-Off

BIAS Lead-Off Detection During Normal Operation

During normal operation, the ADS1299-x BIAS lead-off at power-up function cannot be used because the BIAS amplifier must be powered off.

BIAS Lead Off Detection At Power-Up

This feature is included in the ADS1299-x for use in determining whether the bias electrode is suitably connected. At power-up, the ADS1299-x uses a current source and comparator to determine the BIAS electrode connection status, as shown in Figure 37. The reference level of the comparator is set to determine the acceptable BIAS impedance threshold.

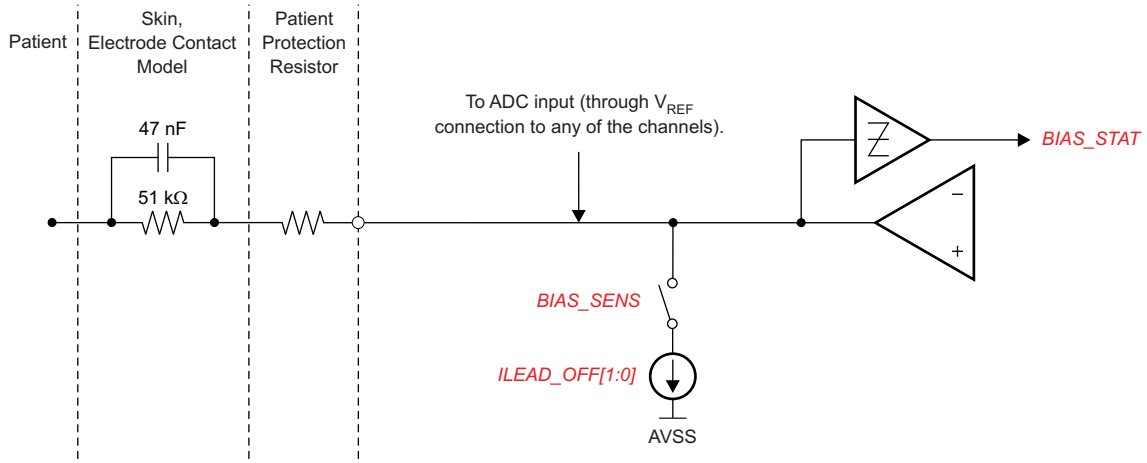
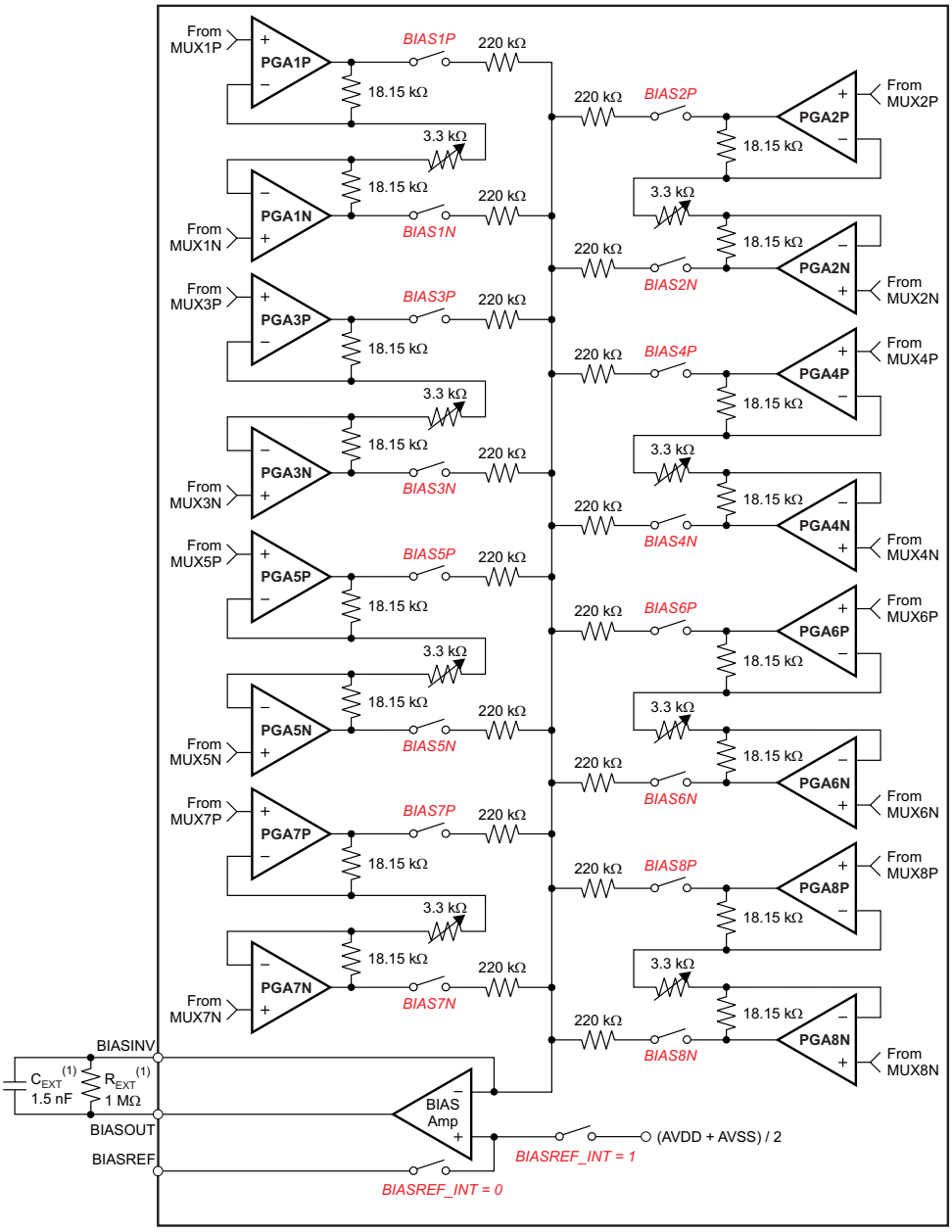


Figure 37. BIAS Lead-Off Detection at Power-Up

When the BIAS amplifier is powered on, the current source has no function. Only the comparator can be used to sense the voltage at the output of the BIAS amplifier. The comparator thresholds are set by the same LOFF[7:5] bits used to set the thresholds for other negative inputs.

9.3.2.4.5 Bias Drive (DC Bias Circuit)

Use the bias circuitry to counter the common-mode interference in a EEG system as a result of power lines and other sources, including fluorescent lights. The bias circuit senses the common-mode voltage of a selected set of electrodes and creates a negative feedback loop by driving the body with an inverted common-mode signal. The negative feedback loop restricts the common-mode movement to a narrow range, depending on the loop gain. Stabilizing the entire loop is specific to the individual user system based on the various poles in the loop. The ADS1299-x integrates the muxes to select the channel and an operational amplifier. All the amplifier terminals are available at the pins, allowing the user to choose the components for the feedback loop. The circuit in Figure 38 shows the overall functional connectivity for the bias circuit.



(1) Typical values.

Figure 38. Bias Drive Amplifier Channel Selection

The reference voltage for the bias drive can be chosen to be internally generated [(AVDD + AVSS) / 2] or provided externally with a resistive divider. The selection of an internal versus external reference voltage for the bias loop is defined by writing the appropriate value to the BIASREF_INT bit in the CONFIG2 register.

If the bias function is not used, the amplifier can be powered down using the PD_BIAS bit (see the *CONFIG3: Configuration Register 3* subsection of the *Register Maps* section for details). Use the PD_BIAS bit to power-down all but one of the bias amplifiers when daisy-chaining multiple ADS1299-x devices.

The BIASIN pin functionality is explained in the *Input Multiplexer* section. An example procedure to use the bias amplifier is shown in the *Bias Drive* section.

9.3.2.4.5.1 Bias Configuration with Multiple Devices

Figure 39 shows multiple devices connected to the bias drive.

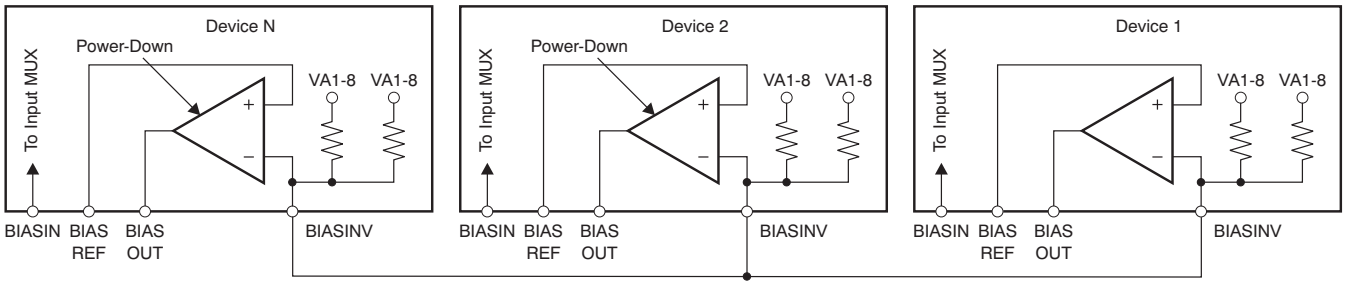


Figure 39. BIAS Drive Connection for Multiple Devices

9.4 Device Functional Modes

9.4.1 Start

Pull the START pin high for at least 2 t_{CLK} periods, or send the START command to begin conversions. When START is low and the START command has not been sent, the device does not issue a $\overline{DRDY}$ signal (conversions are halted).

When using the START command to control conversions, hold the START pin low. The ADS1299-x features two modes to control conversions: continuous mode and single-shot mode. The mode is selected by SINGLE_SHOT (bit 3 of the CONFIG4 register). In multiple device configurations, the START pin is used to synchronize devices (see the *Multiple Device Configuration* subsection of the *SPI Interface* section for more details).

9.4.1.1 Settling Time

The settling time (t_{SETTLE}) is the time required for the converter to output fully-settled data when the START signal is pulled high. When START is pulled high, $\overline{DRDY}$ is also pulled high. The next $\overline{DRDY}$ falling edge indicates that data are ready. Figure 40 shows the timing diagram and Table 7 lists the settling time for different data rates. The settling time depends on f_{CLK} and the decimation ratio (controlled by the DR[2:0] bits in the CONFIG1 register). When the initial settling time has passed, the $\overline{DRDY}$ falling edge occurs at the set data rate, t_{DR}. If data is not read back on DOUT and the output shift register needs to update, $\overline{DRDY}$ goes high for 4 t_{CLK} before returning back low indicating new data is ready. Table 7 lists the settling time as a function of t_{CLK}. Note that when START is held high and there is a step change in the input signal, 3 × t_{DR} is required for the filter to settle to the new value. Settled data are available on the fourth $\overline{DRDY}$ pulse.

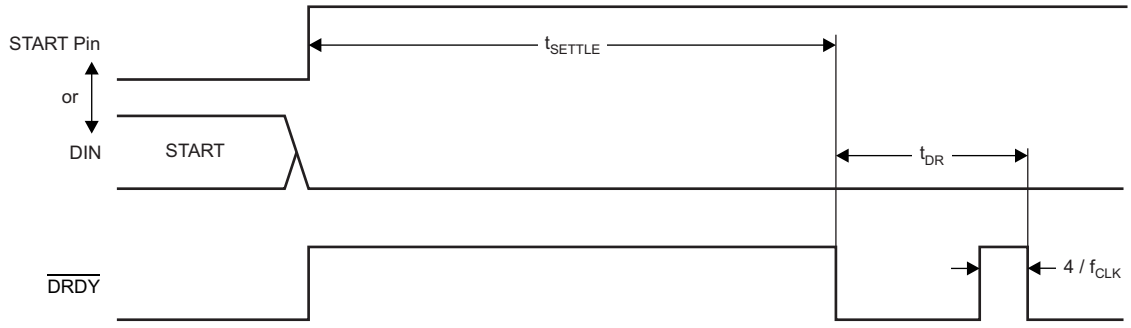


Figure 40. Settling Time

Device Functional Modes (continued)

Table 7. Settling Time for Different Data Rates

DR[2:0]	NORMAL MODE	UNIT
000	521	t_{CLK}
001	1033	t_{CLK}
010	2057	t_{CLK}
011	4105	t_{CLK}
100	8201	t_{CLK}
101	16393	t_{CLK}
110	32777	t_{CLK}

9.4.2 Reset ($\overline{\text{RESET}}$)

There are two methods to reset the ADS1299-x: pull the $\overline{\text{RESET}}$ pin low, or send the RESET command. When using the $\overline{\text{RESET}}$ pin, make sure to follow the minimum pulse duration timing specifications before taking the pin back high. The RESET command takes effect on the eighth SCLK falling edge of the command. After a reset, 18 t_{CLK} cycles are required to complete initialization of the configuration registers to default states and start the conversion cycle. Note that an internal reset is automatically issued to the digital filter whenever the CONFIG1 register is set to a new value with a WREG command.

9.4.3 Power-Down ($\overline{\text{PWDN}}$)

When $\overline{\text{PWDN}}$ is pulled low, all on-chip circuitry is powered down. To exit power-down mode, take the $\overline{\text{PWDN}}$ pin high. Upon exiting from power-down mode, the internal oscillator and the reference require time to wake up. During power-down, the external clock is recommended to be shut down to save power.

9.4.4 Data Retrieval

9.4.4.1 Data Ready ($\overline{\text{DRDY}}$)

$\overline{\text{DRDY}}$ is an output signal which transitions from high to low indicating new conversion data are ready. The $\overline{\text{CS}}$ signal has no effect on the data ready signal. $\overline{\text{DRDY}}$ behavior is determined by whether the device is in RDATAC mode or the RDATA command is used to read data on demand. (See the *RDATAC: Read Data Continuous* and *RDATA: Read Data* subsections of the *SPI Command Definitions* section for further details).

When reading data with the RDATA command, the read operation can overlap the next $\overline{\text{DRDY}}$ occurrence without data corruption.

The START pin or the START command places the device either in normal data capture mode or pulse data capture mode.

Figure 41 shows the relationship between $\overline{\text{DRDY}}$, DOUT, and SCLK during data retrieval (in case of an ADS1299). DOUT is latched out at the SCLK rising edge. $\overline{\text{DRDY}}$ is pulled high at the SCLK falling edge. Note that $\overline{\text{DRDY}}$ goes high on the first SCLK falling edge, regardless of whether data are being retrieved from the device or a command is being sent through the DIN pin.

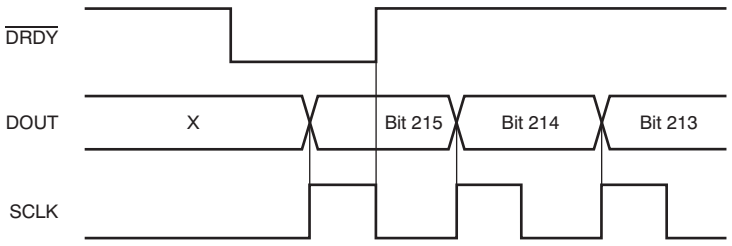


Figure 41. $\overline{\text{DRDY}}$ with Data Retrieval ($\overline{\text{CS}}$ = 0)

Device Functional Modes (continued)

9.4.4.2 Reading Back Data

Data retrieval can be accomplished in one of two methods:

1. RDATAC: the read data continuous command sets the device in a mode that reads data continuously without sending commands. See the *RDATAC: Read Data Continuous* section for more details.
2. RDATA: the read data command requires that a command is sent to the device to load the output shift register with the latest data. See the *RDATA: Read Data* section for more details.

Conversion data are read by shifting data out on DOUT. The MSB of the data on DOUT is clocked out on the first SCLK rising edge. DRDY returns high on the first SCLK falling edge. DIN should remain low for the entire read operation.

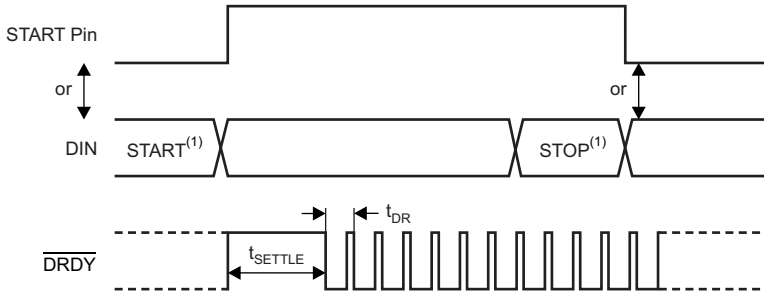
The number of bits in the data output depends on the number of channels and the number of bits per channel. For the 8-channel ADS1299, the number of data outputs is [(24 status bits + 24 bits × 8 channels) = 216 bits]. The format of the 24 status bits is: (1100 + LOFF_STATP + LOFF_STATN + bits[4:7] of the GPIO register). The data format for each channel data are twos complement and MSB first. When channels are powered down using the user register setting, the corresponding channel output is set to '0'. However, the channel output sequence remains the same.

The ADS1299-x also provides a multiple readback feature. Data can be read out multiple times by simply giving more SCLKs in RDATAC mode, in which case the MSB data byte repeats after reading the last byte. The DAISY_EN bit in the CONFIG1 register must be set to '1' for multiple readbacks.

9.4.5 Continuous Conversion Mode

Conversions begin when the START pin is taken high or when the START command is sent. As shown in Figure 42, the DRDY output goes high when conversions are started and goes low when data are ready. Conversions continue indefinitely until the START pin is taken low or the STOP command is transmitted. When the START pin is pulled low or the STOP command is issued, the conversion in progress is allowed to complete. Figure 43 and Table 8 illustrate the required DRDY timing to the START pin or the START and STOP commands when controlling conversions in this mode. The t_{SDSU} timing indicates when to take the START pin low or when to send the STOP command before the DRDY falling edge to halt further conversions. The t_{DSHD} timing indicates when to take the START pin low or send the STOP command after a DRDY falling edge to complete the current conversion and halt further conversions. To keep the converter running continuously, the START pin can be permanently tied high.

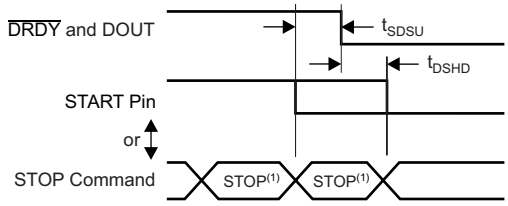
When switching from Single-Shot mode to Continuous Conversion mode, bring the START signal low and back high or send a STOP command followed by a START command. This conversion mode is ideal for applications that require a fixed continuous stream of conversions results.



(1) START and STOP commands take effect on the seventh SCLK falling edge.

Figure 42. Continuous Conversion Mode

Device Functional Modes (continued)



(1) START and STOP commands take effect on the seventh SCLK falling edge at the end of the command.

Figure 43. START to DRDY Timing

Table 8. Timing Characteristics for Figure 43(1)

		MIN	UNIT
t_{SDSU}	START pin low or STOP command to DRDY setup time to halt further conversions	16	t_{CLK}
t_{DSHD}	START pin low or STOP command to complete current conversion	16	t_{CLK}

(1) START and STOP commands take effect on the seventh SCLK falling edge at the end of the command.

9.4.6 Single-Shot Mode

Single-shot mode is enabled by setting the SINGLE_SHOT bit in the CONFIG4 register to '1'. In single-shot mode, the ADS1299-x performs a single conversion when the START pin is taken high or when the START command is sent. As shown in Figure 44, when a conversion is complete, DRDY goes low and further conversions are stopped. Regardless of whether the conversion data are read or not, DRDY remains low. To begin a new conversion, take the START pin low and then back high, or send the START command again. When switching from Continuous Conversion mode to Single-Shot mode, bring the START signal low and back high or send a STOP command followed by a START command.

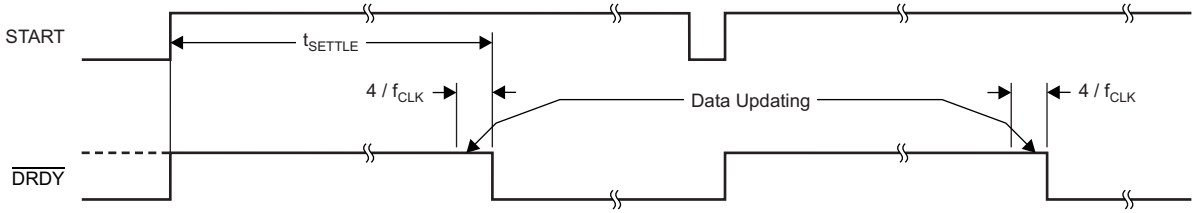


Figure 44. DRDY with No Data Retrieval in Single-Shot Mode

This conversion mode is ideal for applications that require non-standard or non-continuous data rates. Issuing a START command or toggling the START pin high resets the digital filter, effectively dropping the data rate by a factor of four. This mode leaves the system more susceptible to aliasing effects, requiring more complex analog or digital filtering. Loading on the host processor increases because the processor must toggle the START pin or send a START command to initiate a new conversion cycle.

9.5 Programming

9.5.1 Data Format

The device provides 24 bits of data in binary twos complement format. The size of one code (LSB) is calculated using Equation 8.

$$1\ \text{LSB} = (2 \times V_{REF} / \text{Gain}) / 2^{24} = +\text{FS} / 2^{23} \quad (8)$$

A positive full-scale input produces an output code of 7FFFFFh and the negative full-scale input produces an output code of 800000h. The output clips at these codes for signals exceeding full-scale. Table 9 summarizes the ideal output codes for different input signals. All 24 bits toggle when the analog input is at positive or negative full-scale.

Table 9. Ideal Output Code versus Input Signal

INPUT SIGNAL, V_{IN} (INxP - INxN)	IDEAL OUTPUT CODE(1)
≥ FS	7FFFFFh
+FS / (2^{23} – 1)	000001h
0	000000h
–FS / (2^{23} – 1)	FFFFFFh
≤ –FS (2^{23} / 2^{23} – 1)	800000h

(1) Excludes effects of noise, linearity, offset, and gain error.

9.5.2 SPI Interface

The SPI-compatible serial interface consists of four signals: $\overline{CS}$, SCLK, DIN, and DOUT. The interface reads conversion data, reads and writes registers, and controls ADS1299-x operation. The data-ready output, $\overline{DRDY}$ (see the *Data Ready (DRDY)* section), is used as a status signal to indicate when data are ready. $\overline{DRDY}$ goes low when new data are available.

9.5.2.1 Chip Select ($\overline{CS}$)

The $\overline{CS}$ pin activates SPI communication. $\overline{CS}$ must be low before data transactions and must stay low for the entire SPI communication period. When $\overline{CS}$ is high, the DOUT pin enters a high-impedance state. Therefore, reading and writing to the serial interface are ignored and the serial interface is reset. $\overline{DRDY}$ pin operation is independent of $\overline{CS}$. $\overline{DRDY}$ still indicates that a new conversion has completed and is forced high as a response to SCLK, even if $\overline{CS}$ is high.

Taking $\overline{CS}$ high deactivates only the SPI communication with the device and the serial interface is reset. Data conversion continues and the $\overline{DRDY}$ signal can be monitored to check if a new conversion result is ready. A master device monitoring the $\overline{DRDY}$ signal can select the appropriate slave device by pulling the $\overline{CS}$ pin low. After the serial communication is finished, always wait four or more t_{CLK} cycles before taking $\overline{CS}$ high.

9.5.2.2 Serial Clock (SCLK)

SCLK provides the clock for serial communication. SCLK is a Schmitt-trigger input, but TI recommends keeping SCLK as free from noise as possible to prevent glitches from inadvertently shifting the data. Data are shifted into DIN on the falling edge of SCLK and shifted out of DOUT on the rising edge of SCLK.

The absolute maximum SCLK limit is specified in Figure 1. When shifting in commands with SCLK, make sure that the entire set of SCLKs is issued to the device. Failure to do so can result in the device serial interface being placed into an unknown state requiring $\overline{CS}$ to be taken high to recover.

For a single device, the minimum speed required for SCLK depends on the number of channels, number of bits of resolution, and output data rate. (For multiple cascaded devices, see the *Cascaded Mode* subsection of the *Multiple Device Configuration* section.)

For example, if the ADS1299 is used in a 500-SPS mode (8 channels, 24-bit resolution), the minimum SCLK speed is 110 kHz.

Data retrieval can be accomplished either by placing the device in RDATAC mode or by issuing an RDATA command for data on demand. The SCLK rate limitation in Equation 9 applies to RDATAC. For the RDATA command, the limitation applies if data must be read in between two consecutive DRDY signals. Equation 9 assumes that there are no other commands issued in between data captures.

$$t_{SCLK} < \frac{t_{DR} - 4\,t_{CLK}}{N_{BITS} \times N_{CHANNELS} + 24} \quad (9)$$

9.5.2.3 Data Input (DIN)

DIN is used along with SCLK to send data to the device. Data on DIN are shifted into the device on the falling edge of SCLK.

The communication of this device is full-duplex in nature. The device monitors commands shifted in even when data are being shifted out. Data that are present in the output shift register are shifted out when sending in a command. Therefore, make sure that whatever is being sent on the DIN pin is valid when shifting out data. When no command is to be sent to the device when reading out data, send the NOP command on DIN. Make sure that the $t_{SDECODE}$ timing is met in the *Sending Multi-Byte Commands* section when sending multiple byte commands on DIN.

9.5.2.4 Data Output (DOUT)

DOUT is used with SCLK to read conversion and register data from the device. Data are clocked out on the rising edge of SCLK, MSB first. DOUT goes to a high-impedance state when $\overline{CS}$ is high. Figure 45 shows the ADS1299 data output protocol.

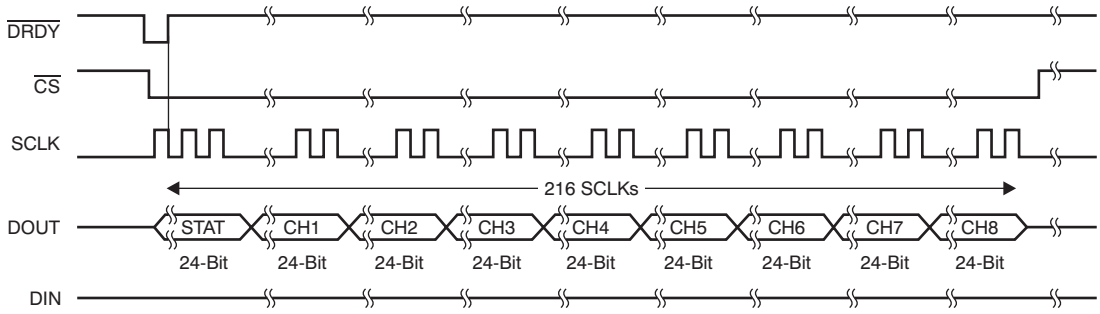


Figure 45. SPI Bus Data Output

9.5.3 SPI Command Definitions

The ADS1299-x provides flexible configuration control. The commands, summarized in Table 10, control and configure device operation. The commands are stand-alone, except for the register read and write operations that require a second command byte plus data. $\overline{CS}$ can be taken high or held low between commands but must stay low for the entire command operation (especially for multi-byte commands). System commands and the RDATA command are decoded by the device on the seventh SCLK falling edge. The register read and write commands are decoded on the eighth SCLK falling edge. Be sure to follow SPI timing requirements when pulling $\overline{CS}$ high after issuing a command.

Table 10. Command Definitions

COMMAND	DESCRIPTION	FIRST BYTE	SECOND BYTE
System Commands			
WAKEUP	Wake-up from standby mode	0000 0010 (02h)	
STANDBY	Enter standby mode	0000 0100 (04h)	
RESET	Reset the device	0000 0110 (06h)	
START	Start and restart (synchronize) conversions	0000 1000 (08h)	
STOP	Stop conversion	0000 1010 (0Ah)	
Data Read Commands			
RDATAC	Enable Read Data Continuous mode. This mode is the default mode at power-up.(1)	0001 0000 (10h)	
SDATAC	Stop Read Data Continuously mode	0001 0001 (11h)	
RDATA	Read data by command; supports multiple read back.	0001 0010 (12h)	
Register Read Commands			
RREG	Read *n nnnn* registers starting at address *r rrrr*	001*r rrrr* (2xh)(2)	000*n nnnn*(2)
WREG	Write *n nnnn* registers starting at address *r rrrr*	010*r rrrr* (4xh)(2)	000*n nnnn*(2)

(1) When in RDATAC mode, the RREG command is ignored.
(2) *n nnnn* = number of registers to be read or written – 1. For example, to read or write three registers, set *n nnnn* = 0 (0010). *r rrrr* = starting register address for read or write commands.

9.5.3.1 Sending Multi-Byte Commands

The ADS1299-x serial interface decodes commands in bytes and requires 4 t_{CLK} cycles to decode and execute. Therefore, when sending multi-byte commands (such as RREG or WREG), a 4 t_{CLK} period must separate the end of one byte (or command) and the next.

Assuming CLK is 2.048 MHz, then $t_{SDECODE}$ (4 t_{CLK}) is 1.96 µs. When SCLK is 16 MHz, one byte can be transferred in 500 ns. This byte transfer time does not meet the $t_{SDECODE}$ specification; therefore, a delay must be inserted so the end of the second byte arrives 1.46 µs later. If SCLK is 4 MHz, one byte is transferred in 2 µs. Because this transfer time exceeds the $t_{SDECODE}$ specification, the processor can send subsequent bytes without delay. In this later scenario, the serial port can be programmed to move from single-byte transfers per cycle to multiple bytes.

9.5.3.2 WAKEUP: Exit STANDBY Mode

The WAKEUP command exits low-power standby mode; see the *STANDBY: Enter STANDBY Mode* subsection of the *SPI Command Definitions* section. Time is required when exiting standby mode (see the *Electrical Characteristics* for details). **There are no SCLK rate restrictions for this command and can be issued at any time.** Any following commands must be sent after a delay of 4 t_{CLK} cycles.

9.5.3.3 STANDBY: Enter STANDBY Mode

The STANDBY command enters low-power standby mode. All parts of the circuit are shut down except for the reference section. The standby mode power consumption is specified in the *Electrical Characteristics*. **There are no SCLK rate restrictions for this command and can be issued at any time.** Do not send any other commands other than the wakeup command after the device enters standby mode.

9.5.3.4 RESET: Reset Registers to Default Values

The RESET command resets the digital filter cycle and returns all register settings to default values. See the *Reset (RESET)* subsection of the *SPI Interface* section for more details. **There are no SCLK rate restrictions for this command and can be issued at any time.** 18 t_{CLK} cycles are required to execute the RESET command. Avoid sending any commands during this time.

9.5.3.5 START: Start Conversions

The START command starts data conversions. Tie the START pin low to control conversions by command. If conversions are in progress, this command has no effect. The STOP command stops conversions. If the START command is immediately followed by a STOP command, then there must be a 4-t_{CLK} cycle delay between them. When the START command is sent to the device, keep the START pin low until the STOP command is issued. (See the *Start* subsection of the *SPI Interface* section for more details.) **There are no SCLK rate restrictions for this command and can be issued at any time.**

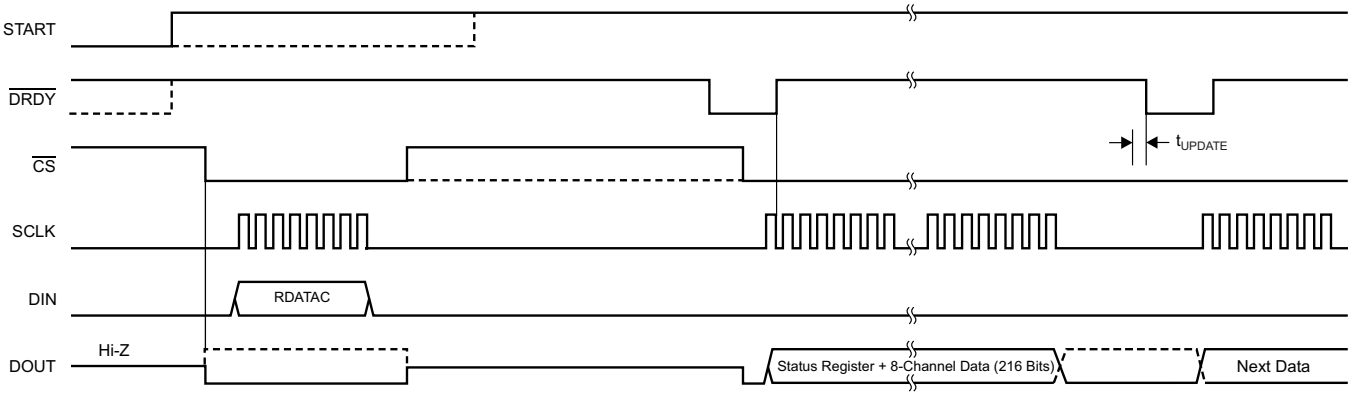
9.5.3.6 STOP: Stop Conversions

The STOP command stops conversions. Tie the START pin low to control conversions by command. When the STOP command is sent, the conversion in progress completes and further conversions are stopped. If conversions are already stopped, this command has no effect. **There are no SCLK rate restrictions for this command and can be issued at any time.**

9.5.3.7 RDATAC: Read Data Continuous

The RDATAC command enables conversion data output on each $\overline{\text{DRDY}}$ without the need to issue subsequent read data commands. This mode places the conversion data in the output register and may be shifted out directly. The read data continuous mode is the device default mode; the device defaults to this mode on power-up.

RDATAC mode is cancelled by the Stop Read Data Continuous command. If the device is in RDATAC mode, a SDATAC command must be issued before any other commands can be sent to the device. **There are no SCLK rate restrictions for this command.** However, subsequent data retrieval SCLKs or the SDATAC command should wait at least 4 t_{CLK} cycles before completion (see the *Sending Multi-Byte Commands* section). RDATAC timing is illustrated in Figure 46. As depicted in Figure 46, there is a *keep out* zone of 4 t_{CLK} cycles around the $\overline{\text{DRDY}}$ pulse where this command cannot be issued in. If no data are retrieved from the device, DOUT and $\overline{\text{DRDY}}$ behave similarly in this mode. To retrieve data from the device after the RDATAC command is issued, make sure either the START pin is high or the START command is issued. Figure 46 shows the recommended way to use the RDATAC command. RDATAC is ideally-suited for applications such as data loggers or recorders, where registers are set one time and do not need to be reconfigured.



(1) $t_{UPDATE} = 4 / f_{CLK}$. Do not read data during this time.

Figure 46. RDATAC Usage

9.5.3.8 SDATAC: Stop Read Data Continuous

The SDATAC command cancels the Read Data Continuous mode. **There are no SCLK rate restrictions for this command, but the next command must wait for 4 t_{CLK} cycles before completion.**

9.5.3.9 RDATA: Read Data

The RDATA command loads the output shift register with the latest data when not in Read Data Continuous mode. Issue this command after $\overline{\text{DRDY}}$ goes low to read the conversion result. There are no SCLK rate restrictions for this command, and there is no wait time needed for the subsequent commands or data retrieval SCLKs. To retrieve data from the device after the RDATA command is issued, make sure either the START pin is high or the START command is issued. When reading data with the RDATA command, the read operation can overlap the next $\overline{\text{DRDY}}$ occurrence without data corruption. Figure 47 shows the recommended way to use the RDATA command. RDATA is best suited for ECG- and EEG-type systems, where register settings must be read or changed often between conversion cycles.

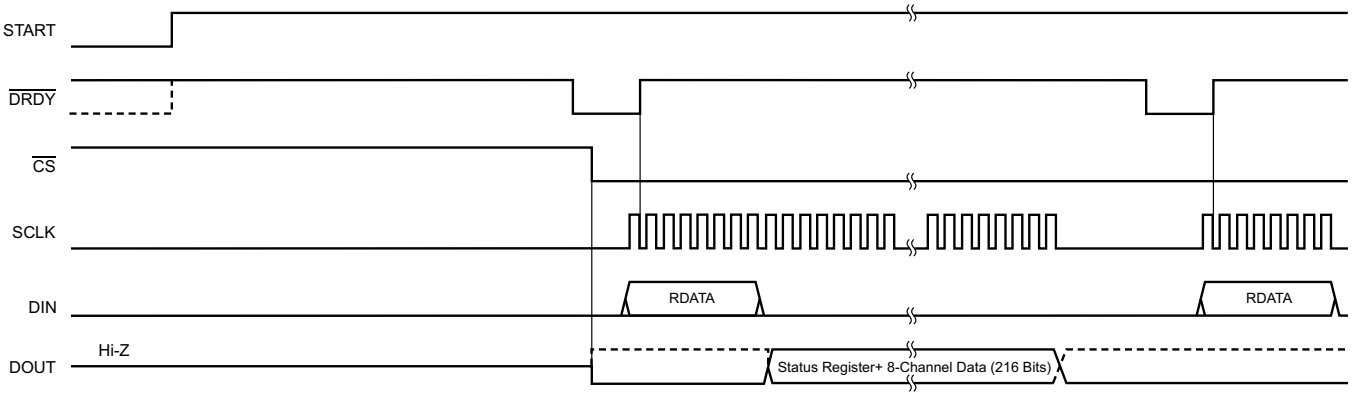


Figure 47. RDATA Usage

9.5.3.10 RREG: Read From Register

This command reads register data. The Register Read command is a two-byte command followed by the register data output. The first byte contains the command and register address. The second command byte specifies the number of registers to read – 1.

First command byte: 001*r rrrr*, where *r rrrr* is the starting register address.

Second command byte: 000*n nnnn*, where *n nnnn* is the number of registers to read – 1.

The 17th SCLK rising edge of the operation clocks out the MSB of the first register, as shown in Figure 48. When the device is in read data continuous mode, an SDATAC command must be issued before the RREG command can be issued. The RREG command can be issued any time. However, because this command is a multi-byte command, there are SCLK rate restrictions depending on how the SCLKs are issued to meet the $t_{SDECODE}$ timing. See the *Serial Clock (SCLK)* subsection of the *SPI Interface* section for more details. Note that $\overline{CS}$ must be low for the entire command.

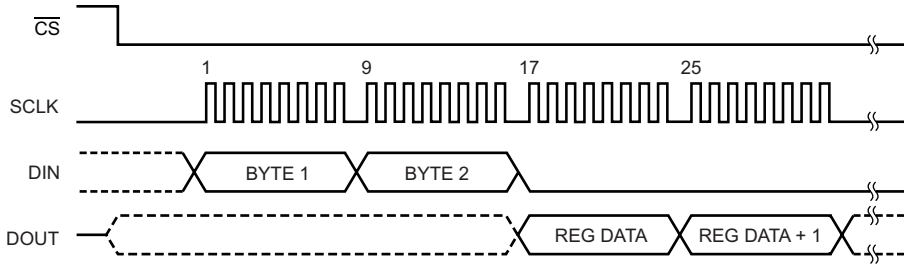


Figure 48. RREG Command Example: Read Two Registers Starting from Register 00h (ID Register) (BYTE 1 = 0010 0000, BYTE 2 = 0000 0001)

9.5.3.11 WREG: Write to Register

This command writes register data. The Register Write command is a two-byte command followed by the register data input. The first byte contains the command and register address. The second command byte specifies the number of registers to write – 1.

First command byte: 010*r rrrr*, where *r rrrr* is the starting register address.

Second command byte: 000*n nnnn*, where *n nnnn* is the number of registers to write – 1.

After the command bytes, the register data follows (in MSB-first format), as shown in Figure 49. The WREG command can be issued any time. However, because this command is a multi-byte command, there are SCLK rate restrictions depending on how the SCLKs are issued to meet the $t_{SDECODE}$ timing. See the *Serial Clock (SCLK)* subsection of the *SPI Interface* section for more details. Note that $\overline{CS}$ must be low for the entire command.

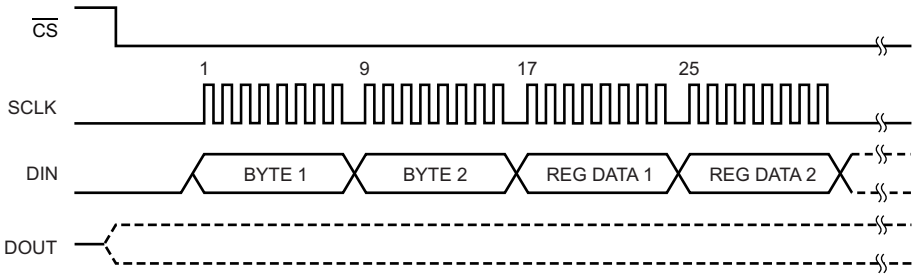


Figure 49. WREG Command Example: Write Two Registers Starting from 00h (ID Register) (BYTE 1 = 0100 0000, BYTE 2 = 0000 0001)

9.6 Register Maps

Table 11 describes the various ADS1299-x registers.

Table 11. Register Assignments

ADDRESS	REGISTER	DEFAULT SETTING	REGISTER BITS							
			7	6	5	4	3	2	1	0
Read Only ID Registers										
00h	ID	xxh	REV_ID[2:0]			1	DEV_ID[1:0]		NU_CH[1:0]	
Global Settings Across Channels										
01h	CONFIG1	96h	1	$\overline{\text{DAISY\_EN}}$	CLK_EN	1	0	DR[2:0]		
02h	CONFIG2	C0h	1	1	0	INT_CAL	0	CAL_AMP0	CAL_FREQ[1:0]	
03h	CONFIG3	60h	$\overline{\text{PD\_REFBUF}}$	1	1	BIAS_MEAS	BIASREF_INT	$\overline{\text{PD\_BIAS}}$	BIAS_LOFF_SENS	BIAS_STAT
04h	LOFF	00h	COMP_TH[2:0]			0	ILEAD_OFF[1:0]		FLEAD_OFF[1:0]	
Channel-Specific Settings										
05h	CH1SET	61h	PD1	GAIN1[2:0]			SRB2	MUX1[2:0]		
06h	CH2SET	61h	PD2	GAIN2[2:0]			SRB2	MUX2[2:0]		
07h	CH3SET	61h	PD3	GAIN3[2:0]			SRB2	MUX3[2:0]		
08h	CH4SET	61h	PD4	GAIN4[2:0]			SRB2	MUX4[2:0]		
09h	CH5SET (1)	61h	PD5	GAIN5[2:0]			SRB2	MUX5[2:0]		
0Ah	CH6SET (1)	61h	PD6	GAIN6[2:0]			SRB2	MUX6[2:0]		
0Bh	CH7SET (2)	61h	PD7	GAIN7[2:0]			SRB2	MUX7[2:0]		
0Ch	CH8SET (2)	61h	PD8	GAIN8[2:0]			SRB2	MUX8[2:0]		
0Dh	BIAS_SENSP	00h	BIASP8(2)	BIASP7(2)	BIASP6(1)	BIASP5(1)	BIASP4	BIASP3	BIASP2	BIASP1
0Eh	BIAS_SENSN	00h	BIASN8(2)	BIASN7(2)	BIASN6(1)	BIASN5(1)	BIASN4	BIASN3	BIASN2	BIASN1
0Fh	LOFF_SENSP	00h	LOFFP8(2)	LOFFP7(2)	LOFFP6(1)	LOFFP5(1)	LOFFP4	LOFFP3	LOFFP2	LOFFP1
10h	LOFF_SENSN	00h	LOFFM8(2)	LOFFM7(2)	LOFFM6(1)	LOFFM5(1)	LOFFM4	LOFFM3	LOFFM2	LOFFM1
11h	LOFF_FLIP	00h	LOFF_FLIP8(2)	LOFF_FLIP7(2)	LOFF_FLIP6(1)	LOFF_FLIP5(1)	LOFF_FLIP4	LOFF_FLIP3	LOFF_FLIP2	LOFF_FLIP1
Lead-Off Status Registers (Read-Only Registers)										
12h	LOFF_STATP	00h	IN8P_OFF	IN7P_OFF	IN6P_OFF	IN5P_OFF	IN4P_OFF	IN3P_OFF	IN2P_OFF	IN1P_OFF
13h	LOFF_STATN	00h	IN8M_OFF	IN7M_OFF	IN6M_OFF	IN5M_OFF	IN4M_OFF	IN3M_OFF	IN2M_OFF	IN1M_OFF
GPIO and OTHER Registers										
14h	GPIO	0Fh	GPIOD[4:1]				GPIOC[4:1]			
15h	MISC1	00h	0	0	SRB1	0	0	0	0	0
16h	MISC2	00h	0	0	0	0	0	0	0	0
17h	CONFIG4	00h	0	0	0	0	SINGLE_SHOT	0	$\overline{\text{PD\_LOFF\_COMP}}$	0

(1) Register or bit only available in the ADS1299-6 and ADS1299. Register bits set to 0h or 00h in the ADS1299-4.
(2) Register or bit only available in the ADS1299. Register bits set to 0h or 00h in the ADS1299-4 and ADS1299-6.

9.6.1 User Register Description

The read-only ID control register is programmed during device manufacture to indicate device characteristics.

9.6.1.1 ID: ID Control Register (address = 00h) (reset = xxh)

Figure 50. ID Control Register

7	6	5	4	3	2	1	0
REV_ID[2:0]			1	DEV_ID[1:0]		NU_CH[1:0]	
R-xh			R-1h	R-3h		R-xh	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 12. ID Control Register Field Descriptions

Bit	Field	Type	Reset	Description
7:5	REV_ID[2:0]	R	xh	**Reserved.** These bits indicate the revision of the device and are subject to change without notice.
4	Reserved	R	1h	**Reserved.** Always read 1.
3:2	DEV_ID[1:0]	R	3h	**Device Identification.** These bits indicates the device. 11 : ADS1299-x
1:0	NU_CH[1:0]	R	xh	**Number of Channels.** These bits indicates number of channels. 00 : 4-channel ADS1299-4 01 : 6-channel ADS1299-6 10 : 8-channel ADS1299

9.6.1.2 CONFIG1: Configuration Register 1 (address = 01h) (reset = 96h)

This register configures the $\overline{\text{DAISY\_EN}}$ bit, clock, and data rate.

Figure 51. CONFIG1: Configuration Register 1

7	6	5	4	3	2	1	0
1	$\overline{\text{DAISY\_EN}}$	CLK_EN	1	0	DR[2:0]		
R/W-1h	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-6h		

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 13. Configuration Register 1 Field Descriptions

Bit	Field	Type	Reset	Description
7	Reserved	R/W	1h	**Reserved** Always write 1h
6	$\overline{\text{DAISY\_EN}}$	R/W	0h	**Daisy-chain or multiple readback mode** This bit determines which mode is enabled. 0 : Daisy-chain mode 1 : Multiple readback mode
5	CLK_EN	R/W	0h	**CLK connection**(1) This bit determines if the internal oscillator signal is connected to the CLK pin when the CLKSEL pin = 1. 0 : Oscillator clock output disabled 1 : Oscillator clock output enabled
4:3	Reserved	R/W	2h	**Reserved** Always write 2h
2:0	DR[2:0]	R/W	6h	**Output data rate** These bits determine the output data rate of the device. $f_{MOD} = f_{CLK} / 2$. 000 : f_{MOD} / 64 (16 kSPS) 001 : f_{MOD} / 128 (8 kSPS) 010 : f_{MOD} / 256 (4 kSPS) 011 : f_{MOD} / 512 (2 kSPS) 100 : f_{MOD} / 1024 (1 kSPS) 101 : f_{MOD} / 2048 (500 SPS) 110 : f_{MOD} / 4096 (250 SPS) 111 : Reserved (do not use)

(1) Additional power is consumed when driving external devices.

9.6.1.3 CONFIG2: Configuration Register 2 (address = 02h) (reset = C0h)

This register configures the test signal generation. See the *Input Multiplexer* section for more details.

Figure 52. CONFIG2: Configuration Register 2

7	6	5	4	3	2	1	0
1	1	0	INT_CAL	0	CAL_AMP	CAL_FREQ[1:0]	
R/W-1h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 14. Configuration Register 2 Field Descriptions

Bit	Field	Type	Reset	Description
7:5	Reserved	R/W	6h	**Reserved** Always write 6h
4	INT_CAL	R/W	0h	**TEST source** This bit determines the source for the test signal. 0 : Test signals are driven externally 1 : Test signals are generated internally
3	Reserved	R/W	0h	**Reserved** Always write 0h
2	CAL_AMP	R/W	0h	**Test signal amplitude** These bits determine the calibration signal amplitude. 0 : 1 × –(VREFP – VREFN) / 2400 1 : 2 × –(VREFP – VREFN) / 2400
1:0	CAL_FREQ[1:0]	R/W	0h	**Test signal frequency** These bits determine the calibration signal frequency. 00 : Pulsed at f_{CLK} / 2^{21} 01 : Pulsed at f_{CLK} / 2^{20} 10 : Do not use 11 : At dc

9.6.1.4 CONFIG3: Configuration Register 3 (address = 03h) (reset = 60h)

Configuration register 3 configures either an internal or exteral reference and BIAS operation.

Figure 53. CONFIG3: Configuration Register 3

7	6	5	4	3	2	1	0
$\overline{\text{PD\_REFBUF}}$	1	1	BIAS_MEAS	BIASREF_INT	$\overline{\text{PD\_BIAS}}$	BIAS_LOFF_SENS	BIAS_STAT
R/W-0h	R/W-1h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 15. Configuration Register 3 Field Descriptions

Bit	Field	Type	Reset	Description
7	$\overline{\text{PD\_REFBUF}}$	R/W	0h	**Power-down reference buffer** This bit determines the power-down reference buffer state. 0 : Power-down internal reference buffer 1 : Enable internal reference buffer
6:5	Reserved	R/W	3h	**Reserved** Always write 3h.
4	BIAS_MEAS	R/W	0h	**BIAS measurement** This bit enables BIAS measurement. The BIAS signal may be measured with any channel. 0 : Open 1 : BIAS_IN signal is routed to the channel that has the MUX_Setting *010* (V_{REF})
3	BIASREF_INT	R/W	0h	**BIASREF signal** This bit determines the BIASREF signal source. 0 : BIASREF signal fed externally 1 : BIASREF signal (AVDD + AVSS) / 2 generated internally
2	$\overline{\text{PD\_BIAS}}$	R/W	0h	**BIAS buffer power** This bit determines the BIAS buffer power state. 0 : BIAS buffer is powered down 1 : BIAS buffer is enabled
1	BIAS_LOFF_SENS	R/W	0h	**BIAS sense function** This bit enables the BIAS sense function. 0 : BIAS sense is disabled 1 : BIAS sense is enabled
0	BIAS_STAT	R	0h	**BIAS lead-off status** This bit determines the BIAS status. 0 : BIAS is connected 1 : BIAS is not connected

9.6.1.5 LOFF: Lead-Off Control Register (address = 04h) (reset = 00h)

The lead-off control register configures the lead-off detection operation.

Figure 54. LOFF: Lead-Off Control Register

7	6	5	4	3	2	1	0
COMP_TH2[2:0]			0	ILEAD_OFF[1:0]		FLEAD_OFF[1:0]	
R/W-0h			R/W-0h	R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 16. Lead-Off Control Register Field Descriptions

Bit	Field	Type	Reset	Description
7:5	COMP_TH[2:0]	R/W	0h	**Lead-off comparator threshold** Comparator positive side 000 : 95% 001 : 92.5% 010 : 90% 011 : 87.5% 100 : 85% 101 : 80% 110 : 75% 111 : 70% Comparator negative side 000 : 5% 001 : 7.5% 010 : 10% 011 : 12.5% 100 : 15% 101 : 20% 110 : 25% 111 : 30%
4	Reserved	R/W	0h	**Reserved** Always write 0h.
3:2	ILEAD_OFF[1:0]	R/W	0h	**Lead-off current magnitude** These bits determine the magnitude of current for the current lead-off mode. 00 : 6 nA 01 : 24 nA 10 : 6 µA 11 : 24 µA
1:0	FLEAD_OFF[1:0]	R/W	0h	**Lead-off frequency** These bits determine the frequency of lead-off detect for each channel. 00 : DC lead-off detection 01 : AC lead-off detection at 7.8 Hz (f_{CLK} / 2^{18}) 10 : AC lead-off detection at 31.2 Hz (f_{CLK} / 2^{16}) 11 : AC lead-off detection at f_{DR} / 4

9.6.1.6 CHnSET: Individual Channel Settings (n = 1 to 8) (address = 05h to 0Ch) (reset = 61h)

The CH[1:8]SET control register configures the power mode, PGA gain, and multiplexer settings channels. See the *Input Multiplexer* section for details. CH[2:8]SET are similar to CH1SET, corresponding to the respective channels.

Figure 55. CHnSET: Individual Channel Settings Register

7	6	5	4	3	2	1	0
PD*n*	GAIN*n*[2:0]			SRB2	MUX*n*[2:0]		
R/W-0h	R/W-6h			R/W-0h	R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 17. Individual Channel Settings (*n* = 1 to 8) Field Descriptions

Bit	Field	Type	Reset	Description
7	PD*n*	R/W	0h	**Power-down** This bit determines the channel power mode for the corresponding channel. 0 : Normal operation 1 : Channel power-down. When powering down a channel, TI recommends that the channel be set to input short by setting the appropriate MUXn[2:0] = 001 of the CHnSET register.
6:4	GAIN*n*[2:0]	R/W	6h	**PGA gain** These bits determine the PGA gain setting. 000 : 1 001 : 2 010 : 4 011 : 6 100 : 8 101 : 12 110 : 24 111 : Do not use
3	SRB2	R/W	0h	**SRB2 connection** This bit determines the SRB2 connection for the corresponding channel. 0 : Open 1 : Closed
2:0	MUX*n*[2:0]	R/W	1h	**Channel input** These bits determine the channel input selection. 000 : Normal electrode input 001 : Input shorted (for offset or noise measurements) 010 : Used in conjunction with BIAS_MEAS bit for BIAS measurements. 011 : MVDD for supply measurement 100 : Temperature sensor 101 : Test signal 110 : BIAS_DRP (positive electrode is the driver) 111 : BIAS_DRN (negative electrode is the driver)

9.6.1.7 BIAS_SENSP: Bias Drive Positive Derivation Register (address = 0Dh) (reset = 00h)

This register controls the selection of the positive signals from each channel for bias voltage (BIAS) derivation. See the *Bias Drive (DC Bias Circuit)* section for details.

Registers bits[5:4] are not available for the ADS1299-4. Register bits[7:6] are not available for the ADS1299-4, or ADS1299-6. Set unavailable bits for the associated device to 0 when writing to the register.

Figure 56. BIAS_SENSP: BIAS Positive Signal Derivation Register

7	6	5	4	3	2	1	0
BIASP8	BIASP7	BIASP6	BIASP5	BIASP4	BIASP3	BIASP2	BIASP1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 18. BIAS Positive Signal Derivation Field Descriptions

Bit	Field	Type	Reset	Description
7	BIASP8	R/W	0h	**IN8P to BIAS** Route channel 8 positive signal into BIAS derivation 0 : Disabled 1 : Enabled
6	BIASP7	R/W	0h	**IN7P to BIAS** Route channel 7 positive signal into BIAS derivation 0 : Disabled 1 : Enabled
5	BIASP6	R/W	0h	**IN6P to BIAS** Route channel 6 positive signal into BIAS derivation 0 : Disabled 1 : Enabled
4	BIASP5	R/W	0h	**IN5P to BIAS** Route channel 5 positive signal into BIAS derivation 0 : Disabled 1 : Enabled
3	BIASP4	R/W	0h	**IN4P to BIAS** Route channel 4 positive signal into BIAS derivation 0 : Disabled 1 : Enabled
2	BIASP3	R/W	0h	**IN3P to BIAS** Route channel 3 positive signal into BIAS derivation 0 : Disabled 1 : Enabled
1	BIASP2	R/W	0h	**IN2P to BIAS** Route channel 2 positive signal into BIAS channel 0 : Disabled 1 : Enabled
0	BIASP1	R/W	0h	**IN1P to BIAS** Route channel 1 positive signal into BIAS channel 0 : Disabled 1 : Enabled

9.6.1.8 BIAS_SENSN: Bias Drive Negative Derivation Register (address = 0Eh) (reset = 00h)

This register controls the selection of the negative signals from each channel for bias voltage (BIAS) derivation. See the *Bias Drive (DC Bias Circuit)* section for details.

Registers bits[5:4] are not available for the ADS1299-4. Register bits[7:6] are not available for the ADS1299-4, or ADS1299-6. Set unavailable bits for the associated device to 0 when writing to the register.

Figure 57. BIAS_SENSN: BIAS Negative Signal Derivation Register

7	6	5	4	3	2	1	0
BIASN8	BIASN7	BIASN6	BIASN5	BIASN4	BIASN3	BIASN2	BIASN1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 19. BIAS Negative Signal Derivation Field Descriptions

Bit	Field	Type	Reset	Description
7	BIASN8	R/W	0h	**IN8N to BIAS** Route channel 8 negative signal into BIAS derivation 0 : Disabled 1 : Enabled
6	BIASN7	R/W	0h	**IN7N to BIAS** Route channel 7 negative signal into BIAS derivation 0 : Disabled 1 : Enabled
5	BIASN6	R/W	0h	**IN6N to BIAS** Route channel 6 negative signal into BIAS derivation 0 : Disabled 1 : Enabled
4	BIASN5	R/W	0h	**IN5N to BIAS** Route channel 5 negative signal into BIAS derivation 0 : Disabled 1 : Enabled
3	BIASN4	R/W	0h	**IN4N to BIAS** Route channel 4 negative signal into BIAS derivation 0 : Disabled 1 : Enabled
2	BIASN3	R/W	0h	**IN3N to BIAS** Route channel 3 negative signal into BIAS derivation 0 : Disabled 1 : Enabled
1	BIASN2	R/W	0h	**IN2N to BIAS** Route channel 2 negative signal into BIAS derivation 0 : Disabled 1 : Enabled
0	BIASN1	R/W	0h	**IN1N to BIAS** Route channel 1 negative signal into BIAS derivation 0 : Disabled 1 : Enabled

9.6.1.9 LOFF_SENSP: Positive Signal Lead-Off Detection Register (address = 0Fh) (reset = 00h)

This register selects the positive side from each channel for lead-off detection. See the *Lead-Off Detection* section for details. The LOFF_STATP register bits are only valid if the corresponding LOFF_SENSP bits are set to 1.

Registers bits[5:4] are not available for the ADS1299-4. Register bits[7:6] are not available for the ADS1299-4, or ADS1299-6. Set unavailable bits for the associated device to 0 when writing to the register.

Figure 58. LOFF_SENSP: Positive Signal Lead-Off Detection Register

7	6	5	4	3	2	1	0
LOFFP8	LOFFP7	LOFFP6	LOFFP5	LOFFP4	LOFFP3	LOFFP2	LOFFP1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 20. Positive Signal Lead-Off Detection Field Descriptions

Bit	Field	Type	Reset	Description
7	LOFFP8	R/W	0h	**IN8P lead off** Enable lead-off detection on IN8P 0 : Disabled 1 : Enabled
6	LOFFP7	R/W	0h	**IN7P lead off** Enable lead-off detection on IN7P 0 : Disabled 1 : Enabled
5	LOFFP6	R/W	0h	**IN6P lead off** Enable lead-off detection on IN6P 0 : Disabled 1 : Enabled
4	LOFFP5	R/W	0h	**IN5P lead off** Enable lead-off detection on IN5P 0 : Disabled 1 : Enabled
3	LOFFP4	R/W	0h	**IN4P lead off** Enable lead-off detection on IN4P 0 : Disabled 1 : Enabled
2	LOFFP3	R/W	0h	**IN3P lead off** Enable lead-off detection on IN3P 0 : Disabled 1 : Enabled
1	LOFFP2	R/W	0h	**IN2P lead off** Enable lead-off detection on IN2P 0 : Disabled 1 : Enabled
0	LOFFP1	R/W	0h	**IN1P lead off** Enable lead-off detection on IN1P 0 : Disabled 1 : Enabled

9.6.1.10 LOFF_SENSN: Negative Signal Lead-Off Detection Register (address = 10h) (reset = 00h)

This register selects the negative side from each channel for lead-off detection. See the *Lead-Off Detection* section for details. The LOFF_STATN register bits are only valid if the corresponding LOFF_SENSN bits are set to 1.

Registers bits[5:4] are not available for the ADS1299-4. Register bits[7:6] are not available for the ADS1299-4, or ADS1299-6. Set unavailable bits for the associated device to 0 when writing to the register.

Figure 59. LOFF_SENSN: Negative Signal Lead-Off Detection Register

7	6	5	4	3	2	1	0
LOFFM8	LOFFM7	LOFFM6	LOFFM5	LOFFM4	LOFFM3	LOFFM2	LOFFM1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 21. Negative Signal Lead-Off Detection Field Descriptions

Bit	Field	Type	Reset	Description
7	LOFFM8	R/W	0h	**IN8N lead off** Enable lead-off detection on IN8N 0 : Disabled 1 : Enabled
6	LOFFM7	R/W	0h	**IN7N lead off** Enable lead-off detection on IN7N 0 : Disabled 1 : Enabled
5	LOFFM6	R/W	0h	**IN6N lead off** Enable lead-off detection on IN6N 0 : Disabled 1 : Enabled
4	LOFFM5	R/W	0h	**IN5N lead off** Enable lead-off detection on IN5N 0 : Disabled 1 : Enabled
3	LOFFM4	R/W	0h	**IN4N lead off** Enable lead-off detectionn on IN4N 0 : Disabled 1 : Enabled
2	LOFFM3	R/W	0h	**IN3N lead off** Enable lead-off detectionion on IN3N 0 : Disabled 1 : Enabled
1	LOFFM2	R/W	0h	**IN2N lead off** Enable lead-off detectionction on IN2N 0 : Disabled 1 : Enabled
0	LOFFM1	R/W	0h	**IN1N lead off** Enable lead-off detectionction on IN1N 0 : Disabled 1 : Enabled

9.6.1.11 LOFF_FLIP: Lead-Off Flip Register (address = 11h) (reset = 00h)

This register controls the direction of the current used for lead-off derivation. See the *Lead-Off Detection* section for details.

Figure 60. LOFF_FLIP: Lead-Off Flip Register

7	6	5	4	3	2	1	0
LOFF_FLIP8	LOFF_FLIP7	LOFF_FLIP6	LOFF_FLIP5	LOFF_FLIP4	LOFF_FLIP3	LOFF_FLIP2	LOFF_FLIP1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 22. Lead-Off Flip Register Field Descriptions

Bit	Field	Type	Reset	Description
7	LOFF_FLIP8	R/W	0h	**Channel 8 LOFF polarity flip** Flip the pull-up or pull-down polarity of the current source on channel 8 for lead-off detection. 0 : No flip = IN8P is pulled to AVDD and IN8N pulled to AVSS 1 : Flipped = IN8P is pulled to AVSS and IN8N pulled to AVDD
6	LOFF_FLIP7	R/W	0h	**Channel 7 LOFF polarity flip** Flip the pull-up or pull-down polarity of the current source on channel 7 for lead-off detection. 0 : No flip = IN7P is pulled to AVDD and IN7N pulled to AVSS 1 : Flipped = IN7P is pulled to AVSS and IN7N pulled to AVDD
5	LOFF_FLIP6	R/W	0h	**Channel 6 LOFF polarity flip** Flip the pull-up or pull-down polarity of the current source on channel 6 for lead-off detection. 0 : No flip = IN6P is pulled to AVDD and IN6N pulled to AVSS 1 : Flipped = IN6P is pulled to AVSS and IN6N pulled to AVDD
4	LOFF_FLIP5	R/W	0h	**Channel 5 LOFF polarity flip** Flip the pull-up or pull-down polarity of the current source on channel 5 for lead-off detection. 0 : No flip = IN5P is pulled to AVDD and IN5N pulled to AVSS 1 : Flipped = IN5P is pulled to AVSS and IN5N pulled to AVDD
3	LOFF_FLIP4	R/W	0h	**Channel 4 LOFF polarity flip** Flip the pull-up or pull-down polarity of the current source on channel 4 for lead-off detection. 0 : No flip = IN4P is pulled to AVDD and IN4N pulled to AVSS 1 : Flipped = IN4P is pulled to AVSS and IN4N pulled to AVDD
2	LOFF_FLIP3	R/W	0h	**Channel 3 LOFF polarity flip** Flip the pull-up or pull-down polarity of the current source on channel 3 for lead-off detection. 0 : No flip = IN3P is pulled to AVDD and IN3N pulled to AVSS 1 : Flipped = IN3P is pulled to AVSS and IN3N pulled to AVDD
1	LOFF_FLIP2	R/W	0h	**Channel 2 LOFF Polarity Flip** Flip the pull-up or pull-down polarity of the current source on channel 2 for lead-off detection. 0 : No flip = IN2P is pulled to AVDD and IN2N pulled to AVSS 1 : Flipped = IN2P is pulled to AVSS and IN2N pulled to AVDD
0	LOFF_FLIP1	R/W	0h	**Channel 1 LOFF Polarity Flip** Flip the pull-up or pull-down polarity of the current source on channel 1 for lead-off detection. 0 : No flip = IN1P is pulled to AVDD and IN1N pulled to AVSS 1 : Flipped = IN1P is pulled to AVSS and IN1N pulled to AVDD

9.6.1.12 LOFF_STATP: Lead-Off Positive Signal Status Register (address = 12h) (reset = 00h)

This register stores the status of whether the positive electrode on each channel is on or off. See the *Lead-Off Detection* section for details. Ignore the LOFF_STATP values if the corresponding LOFF_SENSP bits are not set to 1.

When the LOFF_SENSEP bits are 0, the LOFF_STATP bits should be ignored.

Figure 61. LOFF_STATP: Lead-Off Positive Signal Status Register (Read-Only)

7	6	5	4	3	2	1	0
IN8P_OFF	IN7P_OFF	IN6P_OFF	IN5P_OFF	IN4P_OFF	IN3P_OFF	IN2P_OFF	IN1P_OFF
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 23. Lead-Off Positive Signal Status Field Descriptions

Bit	Field	Type	Reset	Description
7	IN8P_OFF	R	0h	**Channel 8 positive channel lead-off status** Status of whether IN8P electrode is on or off 0 : Electrode is on 1 : Electrode is off
6	IN7P_OFF	R	0h	**Channel 7 positive channel lead-off status** Status of whether IN7P electrode is on or off 0 : Electrode is on 1 : Electrode is off
5	IN6P_OFF	R	0h	**Channel 6 positive channel lead-off status** Status of whether IN6P electrode is on or off 0 : Electrode is on 1 : Electrode is off
4	IN5P_OFF	R	0h	**Channel 5 positive channel lead-off status** Status of whether IN5P electrode is on or off 0 : Electrode is on 1 : Electrode is off
3	IN4P_OFF	R	0h	**Channel 4 positive channel lead-off status** Status of whether IN4P electrode is on or off 0 : Electrode is on 1 : Electrode is off
2	IN3P_OFF	R	0h	**Channel 3 positive channel lead-off status** Status of whether IN3P electrode is on or off 0 : Electrode is on 1 : Electrode is off
1	IN2P_OFF	R	0h	**Channel 2 positive channel lead-off status** Status of whether IN2P electrode is on or off 0 : Electrode is on 1 : Electrode is off
0	IN1P_OFF	R	0h	**Channel 1 positive channel lead-off status** Status of whether IN1P electrode is on or off 0 : Electrode is on 1 : Electrode is off

9.6.1.13 LOFF_STATN: Lead-Off Negative Signal Status Register (address = 13h) (reset = 00h)

This register stores the status of whether the negative electrode on each channel is on or off. See the *Lead-Off Detection* section for details. Ignore the LOFF_STATN values if the corresponding LOFF_SENSN bits are not set to 1.

When the LOFF_SENSEN bits are 0, the LOFF_STATP bits should be ignored.

Figure 62. LOFF_STATN: Lead-Off Negative Signal Status Register (Read-Only)

7	6	5	4	3	2	1	0
IN8N_OFF	IN7N_OFF	IN6N_OFF	IN5N_OFF	IN4N_OFF	IN3N_OFF	IN2N_OFF	IN1N_OFF
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 24. Lead-Off Negative Signal Status Field Descriptions

Bit	Field	Type	Reset	Description
7	IN8N_OFF	R	0h	**Channel 8 negative channel lead-off status** Status of whether IN8N electrode is on or off 0 : Electrode is on 1 : Electrode is off
6	IN7N_OFF	R	0h	**Channel 7 negative channel lead-off status** Status of whether IN7N electrode is on or off 0 : Electrode is on 1 : Electrode is off
5	IN6N_OFF	R	0h	**Channel 6 negative channel lead-off status** Status of whether IN6N electrode is on or off 0 : Electrode is on 1 : Electrode is off
4	IN5N_OFF	R	0h	**Channel 5 negative channel lead-off status** Status of whether IN5N electrode is on or off 0 : Electrode is on 1 : Electrode is off
3	IN4N_OFF	R	0h	**Channel 4 negative channel lead-off status** Status of whether IN4N electrode is on or off 0 : Electrode is on 1 : Electrode is off
2	IN3N_OFF	R	0h	**Channel 3 negative channel lead-off status** Status of whether IN3N electrode is on or off 0 : Electrode is on 1 : Electrode is off
1	IN2N_OFF	R	0h	**Channel 2 negative channel lead-off status** Status of whether IN2N electrode is on or off 0 : Electrode is on 1 : Electrode is off
0	IN1N_OFF	R	0h	**Channel 1 negative channel lead-off status** Status of whether IN1N electrode is on or off 0 : Electrode is on 1 : Electrode is off

9.6.1.14 GPIO: General-Purpose I/O Register (address = 14h) (reset = 0Fh)

The general-purpose I/O register controls the action of the three GPIO pins. When RESP_CTRL[1:0] is in mode 01 and 11, the GPIO2, GPIO3, and GPIO4 pins are not available for use.

Figure 63. GPIO: General-Purpose I/O Register

7	6	5	4	3	2	1	0
GPIOD[4:1]				GPIOC[4:1]			
R/W-0h				R/W-Fh			

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 25. General-Purpose I/O Field Descriptions

Bit	Field	Type	Reset	Description
7:4	GPIOD[4:1]	R/W	0h	**GPIO data** These bits are used to read and write data to the GPIO ports. When reading the register, the data returned correspond to the state of the GPIO external pins, whether they are programmed as inputs or as outputs. As outputs, a write to the GPIOD sets the output value. As inputs, a write to the GPIOD has no effect. GPIO is not available in certain respiration modes.
3:0	GPIOC[4:1]	R/W	Fh	**GPIO control (corresponding GPIOD)** These bits determine if the corresponding GPIOD pin is an input or output. 0 : Output 1 : Input

9.6.1.15 MISC1: Miscellaneous 1 Register (address = 15h) (reset = 00h)

This register provides the control to route the SRB1 pin to all inverting inputs of the four, six, or eight channels (ADS1299-4, ADS1299-6, or ADS1299).

Figure 64. MISC1: Miscellaneous 1 Register

7	6	5	4	3	2	1	0
0	0	SRB1	0	0	0	0	0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 26. Miscellaneous 1 Register Field Descriptions

Bit	Field	Type	Reset	Description
7:6	Reserved	R/W	0h	**Reserved** Always write 0h
5	SRB1	R/W	0h	**Stimulus, reference, and bias 1** This bit connects the SRB1 to all 4, 6, or 8 channels inverting inputs 0 : Switches open 1 : Switches closed
4:0	Reserved	R/W	0h	**Reserved** Always write 0h

9.6.1.16 MISC2: Miscellaneous 2 (address = 16h) (reset = 00h)

This register is reserved for future use.

Figure 65. MISC1: Miscellaneous 1 Register

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 27. Miscellaneous 1 Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	Reserved	R/W	0h	**Reserved** Always write 0h

9.6.1.17 CONFIG4: Configuration Register 4 (address = 17h) (reset = 00h)

This register configures the conversion mode and enables the lead-off comparators.

Figure 66. CONFIG4: Configuration Register 4

7	6	5	4	3	2	1	0
0	0	0	0	SINGLE_SHOT	0	$\overline{\text{PD\_LOFF\_COMP}}$	0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 28. Configuration Register 4 Field Descriptions

Bit	Field	Type	Reset	Description
7:4	Reserved	R/W	0h	**Reserved** Always write 0h
3	SINGLE_SHOT	R/W	0h	**Single-shot conversion** This bit sets the conversion mode. 0 : Continuous conversion mode 1 : Single-shot mode
2	Reserved	R/W	0h	**Reserved** Always write 0h
1	$\overline{\text{PD\_LOFF\_COMP}}$	R/W	0h	**Lead-off comparator power-down** This bit powers down the lead-off comparators. 0 : Lead-off comparators disabled 1 : Lead-off comparators enabled
0	Reserved	R/W	0h	**Reserved** Always write 0h

10 Applications and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

10.1 Application Information

10.1.1 Unused Inputs and Outputs

Power down unused analog inputs and connect them directly to AVDD.

Power down the Bias amplifier if unused and float BIASOUT and BIASINV. BIASIN can also float or can be tied directly to AVSS if unused.

Tie BIASREF directly to AVSS or leave floating if unused.

Tie SRB1 and SRB2 directly to AVSS or leave them floating if unused.

Do not float unused digital inputs because excessive power-supply leakage current might result. Set the two-state mode setting pins high to DVDD or low to DGND through ≥10-kΩ resistors.

Pull $\overline{\text{DRDY}}$ to supply using weak pullup resistor if unused.

If not daisy-chaining devices, tie DAISYIN directly to DGND.

10.1.2 Setting the Device for Basic Data Capture

Figure 67 outlines the procedure to configure the device in a basic state and capture data. This procedure puts the device into a configuration that matches the parameters listed in the specifications section, in order to check if the device is working properly in the user system. Follow this procedure initially until familiar with the device settings. After this procedure has been verified, the device can be configured as needed. For details on the timings for commands, see the appropriate sections in the data sheet. Sample programming codes are added for the ECG and EEG-specific functions.

Application Information (continued)

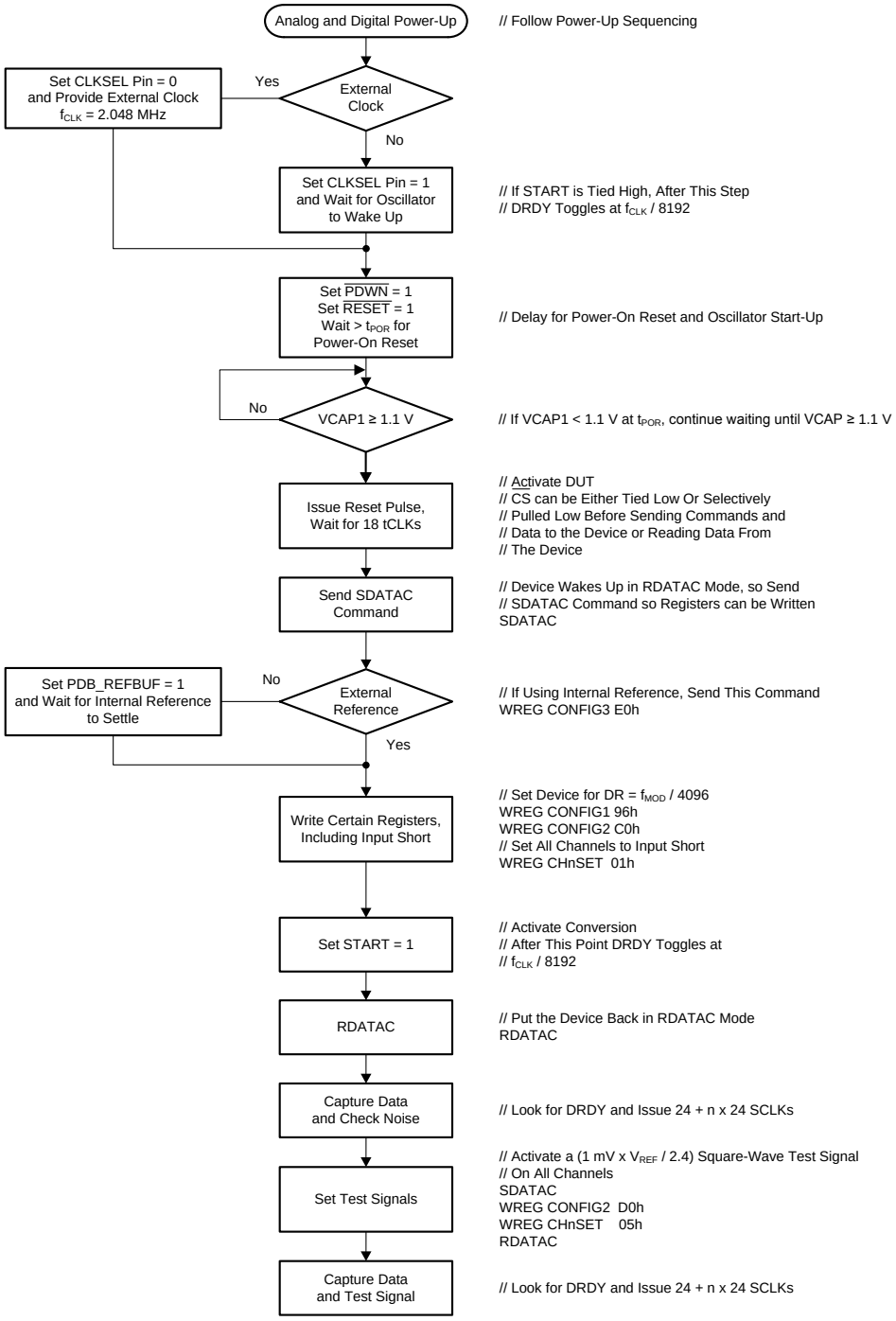


Figure 67. Initial Flow at Power-Up

Application Information (continued)

10.1.2.1 Lead-Off

Sample code to set dc lead-off with pull-up and pull-down resistors on all channels.

```
WREG LOFF       0x13        // Comparator threshold at 95% and 5%, pullup or pulldown resistor
                            // dc lead-off
WREG CONFIG4    0x02        // Turn on dc lead-off comparators
WREG LOFF_SENSP 0xFF        // Turn on the P-side of all channels for lead-off sensing
WREG LOFF_SENSN 0xFF        // Turn on the N-side of all channels for lead-off sensing
```

Observe the status bits of the output data stream to monitor lead-off status.

10.1.2.2 Bias Drive

Sample code to choose bias as an average of the first three channels.

```
WREG RLD_SENSP 0x07        // Select channel 1-3 P-side for RLD sensing
WREG RLD_SENSN 0x07        // Select channel 1-3 N-side for RLD sensing
WREG CONFIG3   b'x1xx 1100 // Turn on BIAS amplifier, set internal BIASREF voltage
```

Sample code to route the BIASOUT signal through channel 4 N-side and measure bias with channel 5. Make sure the external side to the chip BIASOUT is connected to BIASIN.

```
WREG CONFIG3 b'xxx1 1100  // Turn on BIAS amp, set internal BIASREF voltage, set BIAS measurement bit
WREG CH4SET  b'xxxx 0111  // Route BIASIN to channel 4 N-side
WREG CH5SET  b'xxxx 0010  // Route BIASIN to be measured at channel 5 w.r.t BIASREF
```

10.1.3 Establishing the Input Common-Mode

The ADS1299-x measures fully-differential signals where the common-mode voltage point is the midpoint of the positive and negative analog input. The internal PGA restricts the common-mode input range because of the headroom required for operation. The human body is prone to common-mode drifts because noise easily couples onto the human body, similar to an antenna. These common-mode drifts may push the ADS1299-x input common-mode voltage out of the measurable range of the ADC.

If a patient-drive electrode is used by the system, the ADS1299-x includes an on-chip bias drive (BIAS) amplifier that connects to the patient drive electrode. The BIAS amplifier function is to bias the patient to maintain the other electrode common-mode voltages within the valid range. When powered on, the amplifier uses either the analog midsupply voltage, or the voltage present at the BIASREF pin, as a reference input to drive the patient to that voltage.

The ADS1299-x provides the option to use input electrode voltages as feedback to the amplifier to more effectively stabilize the output to the amplifier reference voltage by setting corresponding bits in the BIAS_SENSP and BIAS_SENSN registers. Figure 68 shows an example of a three-electrode system that leverages this technique.

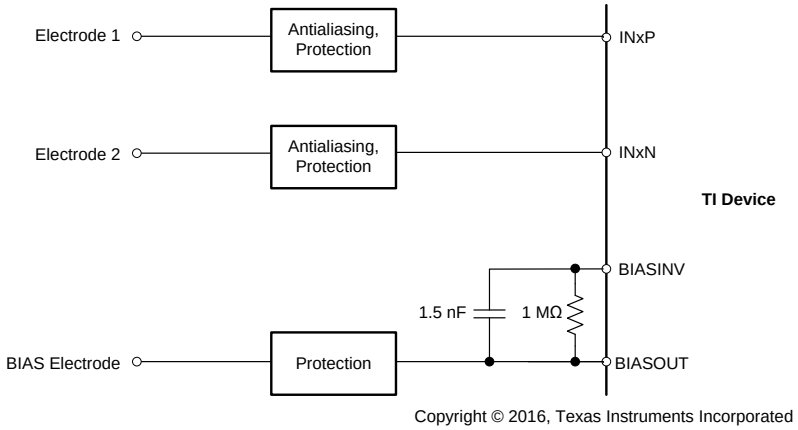


Figure 68. Setting Common-Mode Using BIAS Electrode

Application Information (continued)

10.1.4 Multiple Device Configuration

The ADS1299-x is designed to provide configuration flexibility when multiple devices are used in a system. The serial interface typically needs four signals: DIN, DOUT, SCLK, and $\overline{CS}$. With one additional chip select signal per device, multiple devices can be connected together. The number of signals needed to interface *n* devices is 3 + *n*.

The BIAS drive amplifiers can be daisy-chained, as explained in the *Bias Configuration with Multiple Devices* section. To use the internal oscillator in a daisy-chain configuration, one device must be set as the master for the clock source with the internal oscillator enabled (CLKSEL pin = 1) and the internal oscillator clock brought out of the device by setting the $\overline{CLK}\_EN$ register bit to '1'. This master device clock is used as the external clock source for other devices.

When using multiple devices, the devices can be synchronized with the START signal. The delay from START to the $\overline{DRDY}$ signal is fixed for a given data rate (see the *Start* subsection of the *SPI Interface* section for more details on the settling times). Figure 69 shows the behavior of two devices when synchronized with the START signal.

There are two ways to connect multiple devices with a optimal number of interface pins: cascade mode and daisy-chain mode.

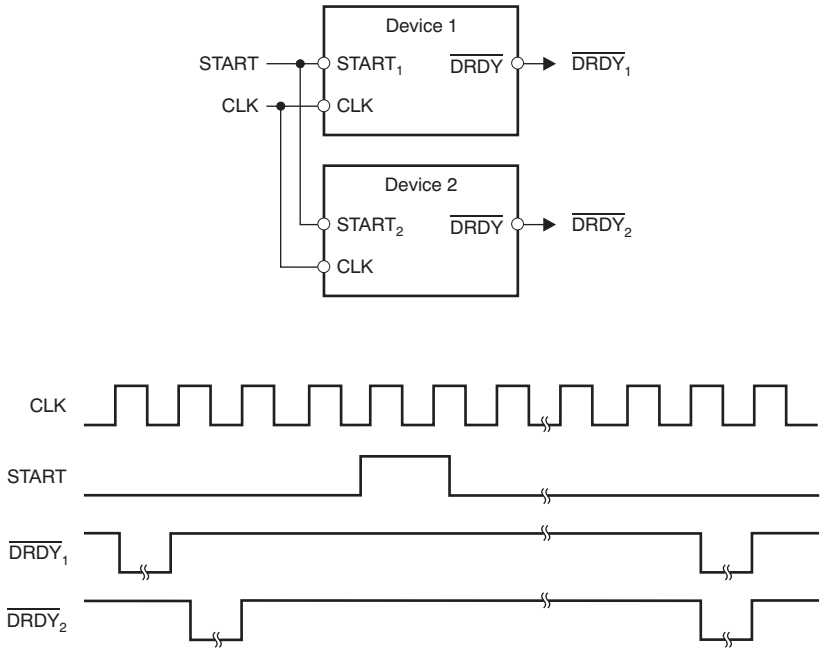


Figure 69. Synchronizing Multiple Converters

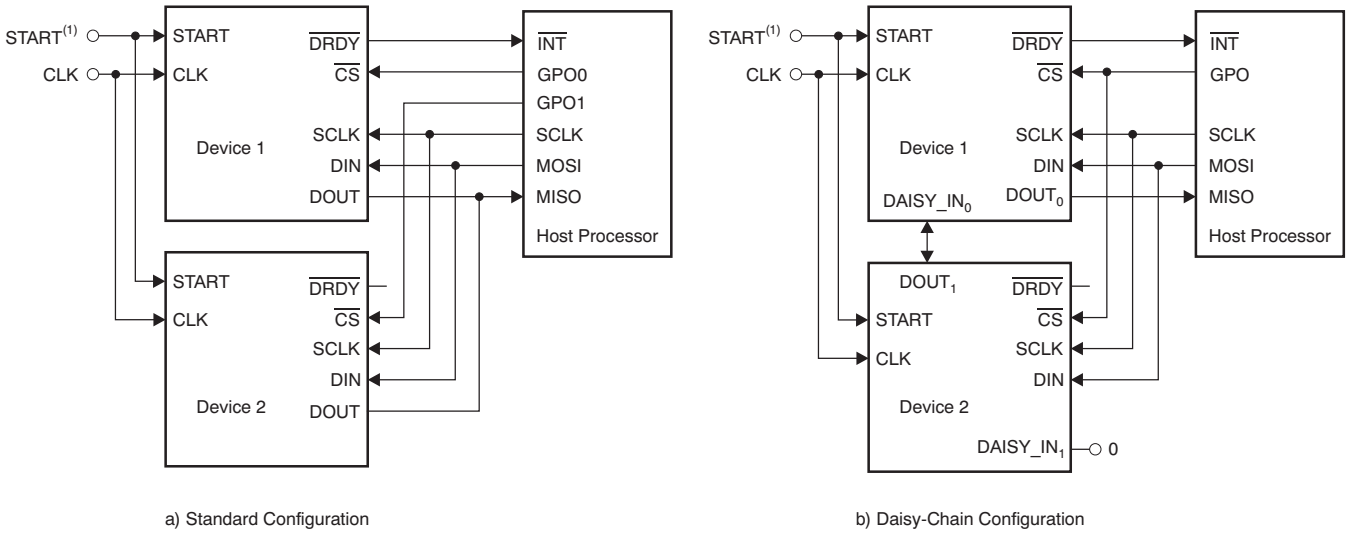
10.1.4.1 Cascaded Mode

Figure 70a illustrates a configuration with two devices cascaded together. Together, the devices create a system with 16 channels. DOUT, SCLK, and DIN are shared. Each device has its own chip select. When a device is not selected by the corresponding $\overline{CS}$ being driven to logic 1, the DOUT of this device is high-impedance. This structure allows the other device to take control of the DOUT bus. This configuration method is suitable for the majority of applications.

Application Information (continued)

10.1.4.2 Daisy-Chain Mode

Daisy-chain mode is enabled by setting the $\overline{\text{DAISY\_EN}}$ bit in the CONFIG1 register. Figure 70b shows the daisy-chain configuration. In this mode SCLK, DIN, and $\overline{\text{CS}}$ are shared across multiple devices. The DOUT of the second device is connected to the DAISY_IN of the first device, thereby creating a chain. When using daisy-chain mode, the multiple readback feature is not available. Short the DAISY_IN pin to digital ground if not used. Figure 2 describes the required timing for the device shown in the configurations of Figure 70. Status and data from device 1 appear first on DOUT, followed by the status and data from device 2. The ADS1299 can be daisy chained with a second ADS1299, an ADS1299-6, or an ADS1299-4.



a) Standard Configuration b) Daisy-Chain Configuration

(1) To reduce pin count, set the START pin low and use the START serial command to synchronize and start conversions.

Figure 70. Multiple Device Configurations

When all devices in the chain operate in the same register setting, DIN can be shared as well. This configuration reduces the SPI communication signals to four, regardless of the number of devices. The BIAS driver cannot be shared among the multiple devices and an external clock must be used because the individual devices cannot be programmed when sharing a common DIN.

Note that from Figure 2, the SCLK rising edge shifts data out of the device on DOUT. The SCLK negative edge is used to latch data into the device DAISY_IN pin down the chain. This architecture allows for a faster SCLK rate speed, but also makes the interface sensitive to board-level signal delays. The more devices in the chain, the more challenging adhering to setup and hold times becomes. A star-pattern connection of SCLK to all devices, minimizing DOUT length, and other printed circuit board (PCB) layout techniques helps. Placing delay circuits (such as buffers) between DOUT and DAISY_IN are ways to mitigate this challenge. One other option is to insert a *D* flip-flop between DOUT and DAISY_IN clocked on an inverted SCLK. Note also that daisy-chain mode requires some software overhead to recombine data bits spread across byte boundaries. Figure 71 shows a timing diagram for this mode.

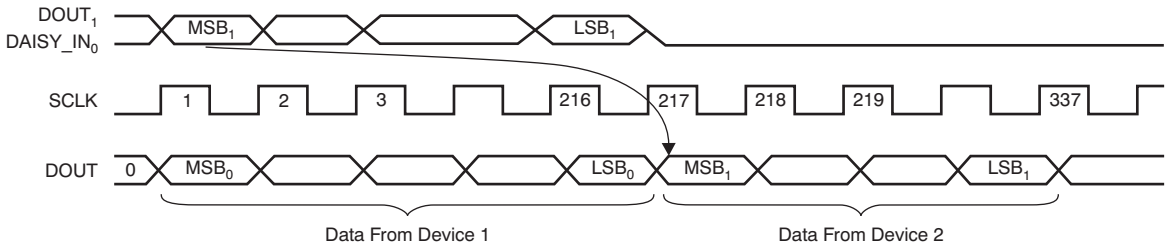


Figure 71. Daisy-Chain Timing

Application Information (continued)

The maximum number of devices that can be daisy-chained depends on the data rate at which the device is operated at. The maximum number of devices can be approximately calculated with Equation 10.

$$N_{DEVICES} = \frac{f_{SCLK}}{f_{DR}\,(N_{BITS})(N_{CHANNELS}) + 24} \quad (10)$$

where:

N_{BITS} = device resolution (depending on data rate), and

$N_{CHANNELS}$ = number of channels in the device.

For example, when the 8-channel ADS1299 is operated at a 2-kSPS data rate with a 4-MHz f_{SCLK}, 10 devices can be daisy-chained.

10.2 Typical Application

The biopotential signals that are measured in electroencephalography (EEG) are small when compared to other types of biopotential signals. The ADS1299 is equipped to measure such small signals due to its extremely low input-referred noise from its high performance internal PGA. Figure 72 and Figure 73 are examples of how the ADS1299 may be configured in typical EEG measurement setups. Figure 72 shows how to measure electrode potentials in a sequential montage, whereas Figure 73 illustrates referential montage measurement connections.

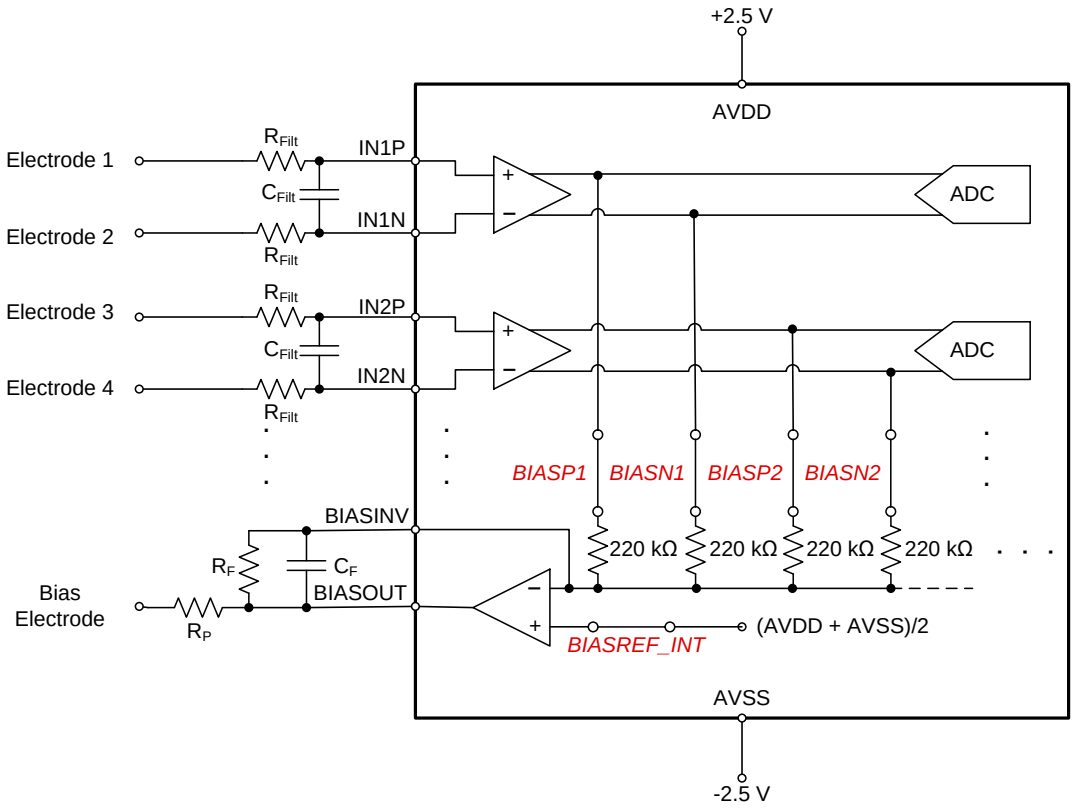


Figure 72. Example Schematic Using the ADS1299 in an EEG Data Acquisition Application, Sequential Montage

Typical Application (continued)

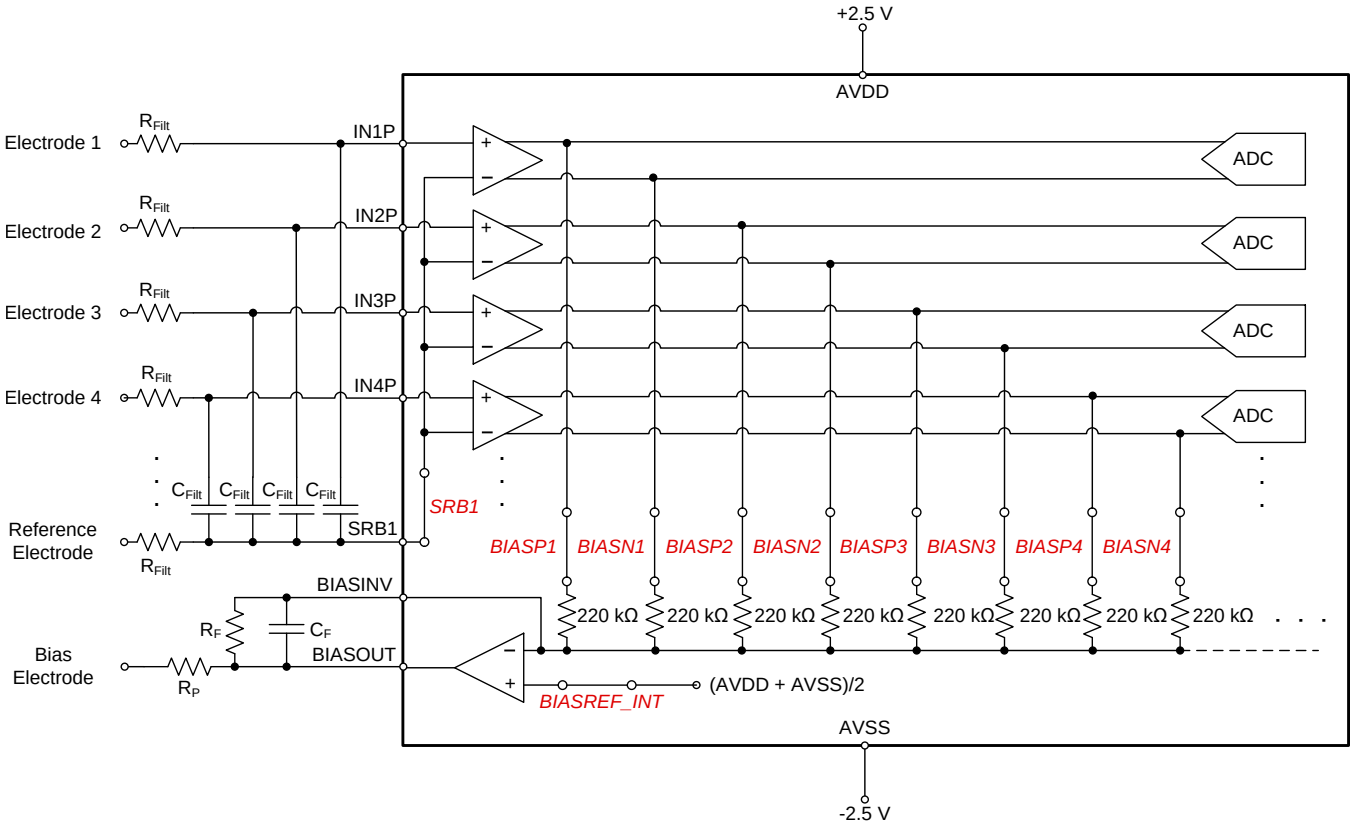


Figure 73. Example Schematic Using the ADS1299 in an EEG Data Acquisition Application, Referential Montage

10.2.1 Design Requirements

Table 29 shows the design requirements for a typical EEG measurement system.

Table 29. EEG Data Acquisition Design Requirements

DESIGN PARAMETER	VALUE
Bandwidth	1 Hz - 50 Hz
Minimum signal bandwidth	10 μV_{Pk}
Input Impedance	> 10 MΩ
Coupling	dc

10.2.2 Detailed Design Procedure

Each channel on the ADS1299 is optimized to measure a separate EEG waveform. The specific connections depend on the EEG montage. The sequential montage is a configuration where each channel represents the voltage between two adjacent electrodes. For example, to measure the potential between electrode Fp1 and F7 on channel 1 of the ADS1299, route the Fp1 electrode to IN1P and the F7 electrode to IN1N. The connections for a sequential montage are illustrated in Figure 72.

Alternatively, EEG electrodes can be measured in a referential montage in which each of the electrodes is measured with respect to a single *reference electrode*. This montage also allows calculation of the waveforms that would have been measured in a sequential montage by finding the difference between two electrode waveforms which were measured with respect to the same electrode. The ADS1299 allows for such a configuration through the use of the SRB1 pin. The SRB1 pin on the ADS1299 may be internally routed to each channel negative input by setting the SRB1 bit in the MISC1 register. When the reference electrode is connected to the SRB1 pin and all other electrodes are connected to the respective positive channel inputs, the electrode voltages can be measured with a referential montage. The referential montage is illustrated in Figure 73. See Figure 18 for a diagram of the channel input multiplexer options.

The ADS1299 is designed to be an EEG front end such that no additional amplification or buffer stage is needed between the electrodes and ADS1299. The ADS1299 has a low-noise PGA with excellent input-referred noise performance. For certain data rate and gain settings, the ADS1299 introduces significantly less than 1 μV_{RMS} of input-referred noise to the signal chain making the device more than capable of handling the 10-μV_{Pk} minimum signal amplitude. ADS1299 noise performance for different PGA gains and data rate settings is listed in Table 1, Table 2, Table 3, and Table 4.

Traditional EEG data acquisition systems high-pass filter the signals in the front-end to remove dc signal content. This topology allows the signal to be amplified by a large gain so the signal can be digitized by a 12- to 16-bit ADC. The ADS1299 24-bit resolution allows the signal to be dc-coupled to the ADC because small EEG signal information can be measured in addition to a significant dc offset.

The ADS1299 channel inputs have very low input bias current allowing electrodes to be connected to the inputs of the ADS1299 with very little leakage current flowing on the patient cables. The ADS1299 has a minimum dc input impedance of 1 GΩ when the lead-off current sources are disabled and 500 MΩ typically when the lead-off current sources are enabled.

The passive components R_{Filt} and C_{Filt} form low-pass filters. In general, the filter is advised to be formed by using a differential capacitor C_{Filt} that shunts the inputs rather than individual RC filters whose capacitors shunt to ground. The differential capacitor configuration significantly improves common-mode rejection because this approach removes dependence on component mismatch.

The cutoff frequency for the filter can be placed well past the data rate of the ADC because of the delta-sigma ADC filter-then-decimate topology. Take care to prevent aliasing around the first repetition of the digital decimation filter response at f_{MOD}. Assuming a 2.048-MHz f_{CLK}, f_{MOD} = 1.024 MHz. The value of R_{Filt} has a minimum set by technical standards for medical electronics. The capacitor value must be set to arrange the proper cutoff frequency.

If the system is likely to be exposed to high-frequency EMI, adding very small-value, common-mode capacitors to the inputs is advisable to filter high-frequency common-mode signals. If these capacitors are added, then the capacitors should be 10 or 20 times smaller than the differential capacitor to ensure their effect of CMRR is minimized.

The integrated bias amplifier serves two purposes in an EEG data acquisition system with the ADS1299. The bias amplifier provides a bias voltage that, when applied to the patient, keeps the measurement electrode common-mode voltage within the rails of the ADS1299. This scenario allows for dc coupling. In addition, the bias amplifier can be configured to provide negative common-mode feedback to the patient to cancel unwanted common-mode signals appearing on the electrodes. This feature is especially helpful because biopotential acquisition systems are notoriously prone to mains-frequency common-mode interference.

The bias amplifier is powered on by setting the $\overline{\text{PD\_BIAS}}$ bit in the CONFIG3 register. Set the BIASREF_INT bit in the CONFIG3 register to input the internally generated analog mid-supply voltage the noninverting input of the bias amplifier. To enable an electrode as an input to the bias amplifier, set the corresponding bit in the BIAS_SENSP or BIAS_SENSN register.

The dc gain of the bias amplifier is determined by R_{Bias} and the number of channel inputs enabled as inputs to the bias amplifier. The bias amplifier circuit only passes common-mode signals. Therefore, the 330-kΩ resistors at each PGA output are *in parallel* for common-mode signals. The bias amplifier is configured in an inverting gain scheme. The formula for determining dc gain for common-mode signals input to the bias amplifier is shown in Equation 11. The capacitor C_f sets the bandwidth for the bias amplifier. Ensure that the amplifier has enough bandwidth to output all the intended common-mode signals.

$$\frac{V_{out}}{V_{in}} = -\frac{R_f \times N}{330k\Omega} \tag{11}$$

Another advantage to a dc-coupled EEG data acquisition system is the ability to detect when an electrode no longer makes good contact with the patient. The ADS1299 features integrated lead-off detection electronics. The *Lead-Off Detection* section explains how to use the lead-off feature on the ADS1299. Note that when configured in a referential montage, only use one lead-off current source with the reference electrode.

10.2.3 Application Curves

Testing the capability of the ADS1299 to measure signals in the band and near the amplitude of typical EEG signals can be done with a precision signal generator. The ADS1299 was tested in a configuration like the one shown in Figure 74.

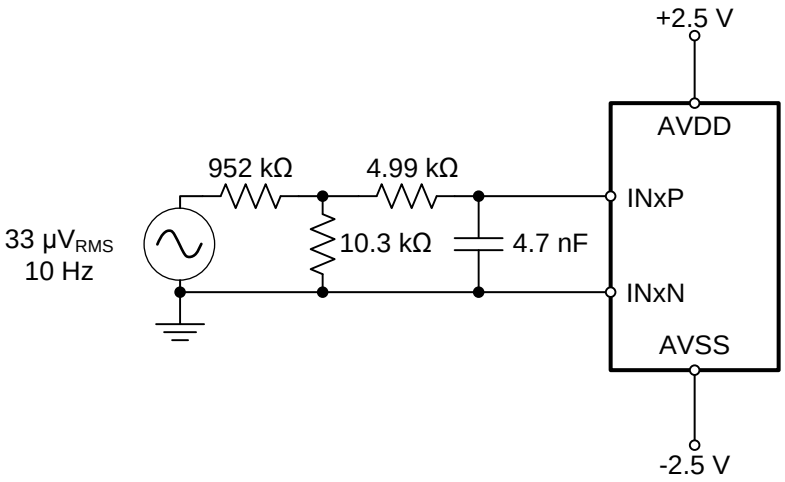


Figure 74. Example Schematic Using the ADS1299 in an EEG Data Acquisition Application, Referential Montage

The 952-kΩ and 10.3-kΩ resistors were used to attenuate the voltage from the signal source because the source could not reach the desired magnitude directly. With the voltage divider, the signal appearing at the inputs was a 3.5-μV_{RMS}, 10-Hz sine wave. Figure 75 shows the input-referred conversion results from the ADS1299 following calibration for offset. The signal that is measured is similar to some of the smallest extracranial EEG signals that can be measured with typical EEG acquisition systems. The signal can be clearly identified. Given this measurement setup was a single-ended configuration without shielding, the measurement setup was subject to significant mains interference. A digital low-pass filter was applied to remove the interference.

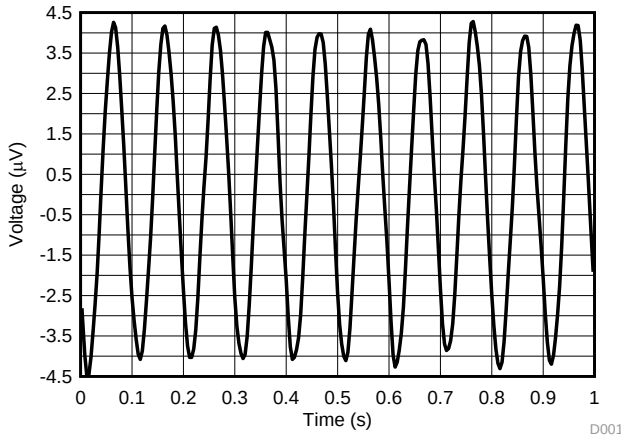


Figure 75. ADS1299 10-Hz Input Signal Results

11 Power Supply Recommendations

The ADS1299-x has three power supplies: AVDD, AVDD1, and DVDD. For best performance, both AVDD and AVDD1 must be as quiet as possible. AVDD1 provides the supply to the charge pump block and has transients at f_{CLK}. Therefore, star connect AVDD1 to the AVDD pins and AVSS1 to the AVSS pins. AVDD and AVDD1 noise that is nonsynchronous with the ADS1299-x operation must be eliminated. Bypass each device supply with 10-µF and 0.1-µF solid ceramic capacitors. For best performance, place the digital circuits (DSP, microcontrollers, FPGAs, and so forth) in the system so that the return currents on those devices do not cross the analog return path of the device. Power the ADS1299-x from unipolar or bipolar supplies.

Use surface-mount, low-cost, low-profile, multilayer ceramic-type capacitors for decoupling. In most cases, the VCAP1 capacitor is also a multilayer ceramic; however, in systems where the board is subjected to high- or low-frequency vibration, install a nonferroelectric capacitor, such as a tantalum or class 1 capacitor (C0G or NPO). EIA class 2 and class 3 dielectrics such as (X7R, X5R, X8R, and so forth) are ferroelectric. The piezoelectric property of these capacitors can appear as electrical noise coming from the capacitor. When using internal reference, noise on the VCAP1 node results in performance degradation.

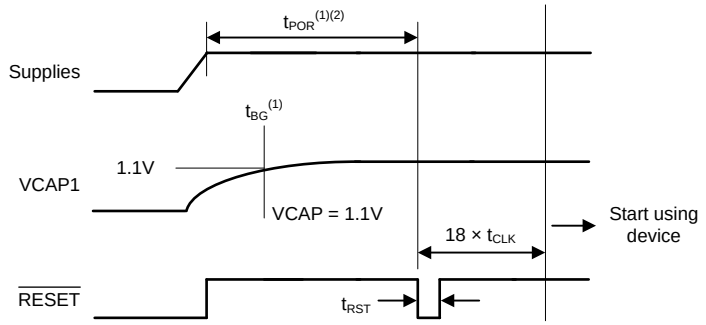
11.1 Power-Up Sequencing

Before device power up, all digital and analog inputs must be low. At the time of power up, keep all of these signals low until the power supplies have stabilized, as shown in Figure 76.

Allow time for the supply voltages to reach their final value, and then begin supplying the master clock signal to the CLK pin. Wait for time t_{POR}, then transmit a reset pulse using either the $\overline{\text{RESET}}$ pin or RESET command to initialize the digital portion of the chip. Issue the reset after t_{POR} or after the VCAP1 voltage is greater than 1.1 V, whichever time is longer. Note that:

- t_{POR} is described in Table 30.
- The VCAP1 pin charge time is set by the RC time constant set by the capacitor value on VCAP1; see Figure 25.

After releasing the $\overline{\text{RESET}}$ pin, program the configuration registers. The power-up sequence timing is shown in Table 30.



(1) Timing to reset pulse is t_{POR} or after t_{BG}, whichever is longer.

(2) When using an external clock, t_{POR} timing does not start until CLK is present and valid.

Figure 76. Power-Up Timing Diagram

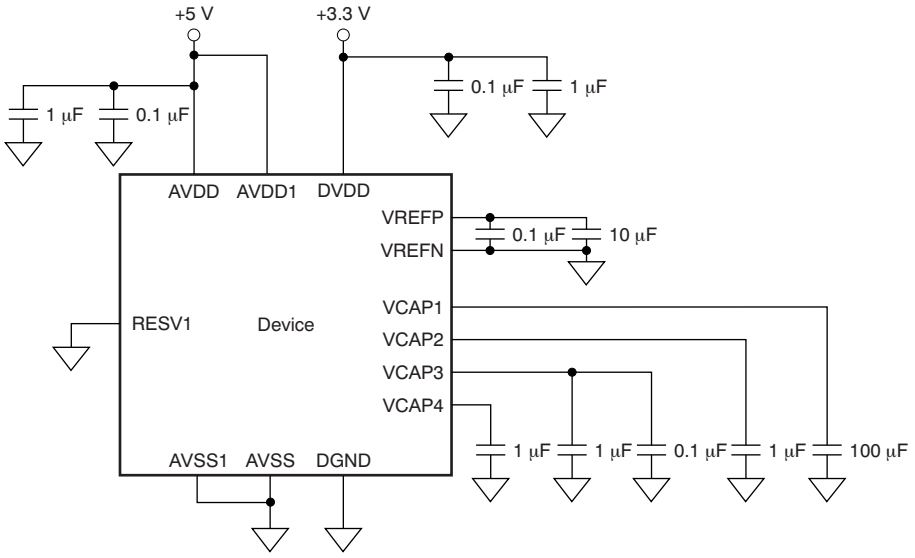
Table 30. Timing Requirements for Figure 76

		MIN	MAX	UNIT
t_{POR}	Wait after power up until reset	2^{18}		t_{CLK}
t_{RST}	Reset low duration	2		t_{CLK}

11.2 Connecting the Device to Unipolar (5 V and 3.3 V) Supplies

Figure 77 illustrates the ADS1299-x connected to a unipolar supply. In this example, analog supply (AVDD) is referenced to analog ground (AVSS) and digital supply (DVDD) is referenced to digital ground (DGND).

Connecting the Device to Unipolar (5 V and 3.3 V) Supplies (continued)

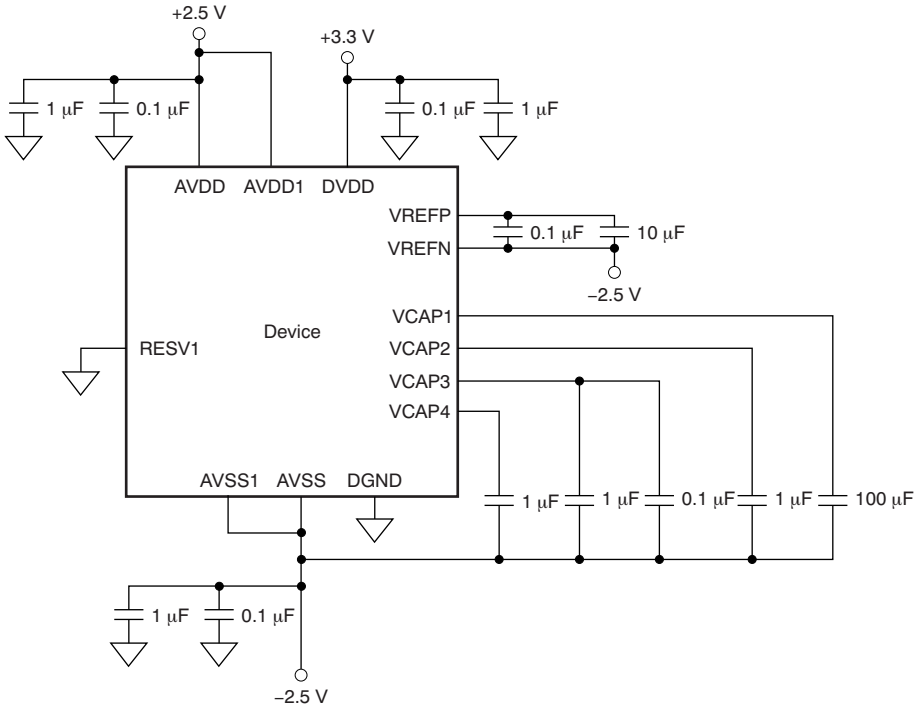


NOTE: Place the capacitors for supply, reference, and VCAP1 to VCAP4 as close to the package as possible.

Figure 77. Single-Supply Operation

11.3 Connecting the Device to Bipolar (±2.5 V and 3.3 V) Supplies

Figure 78 shows the ADS1299-x connected to a bipolar supply. In this example, the analog supplies connect to the device analog supply (AVDD). This supply is referenced to the device analog return (AVSS), and the digital supply (DVDD) is referenced to the device digital ground return (DGND).



NOTE: Place the capacitors for supply, reference, and VCAP1 to VCAP4 as close to the package as possible.

Figure 78. Bipolar Supply Operation

12 Layout

12.1 Layout Guidelines

TI recommends employing best design practices when laying out a printed-circuit board (PCB) for both analog and digital components. This recommendation generally means that the layout separates analog components [such as ADCs, amplifiers, references, digital-to-analog converters (DACs), and analog MUXs] from digital components [such as microcontrollers, complex programmable logic devices (CPLDs), field-programmable gate arrays (FPGAs), radio frequency (RF) transceivers, universal serial bus (USB) transceivers, and switching regulators]. An example of good component placement is shown in Figure 79. Although Figure 79 provides a good example of component placement, the best placement for each application is unique to the geometries, components, and PCB fabrication capabilities employed. That is, there is no single layout that is perfect for every design and careful consideration must always be used when designing with any analog component.

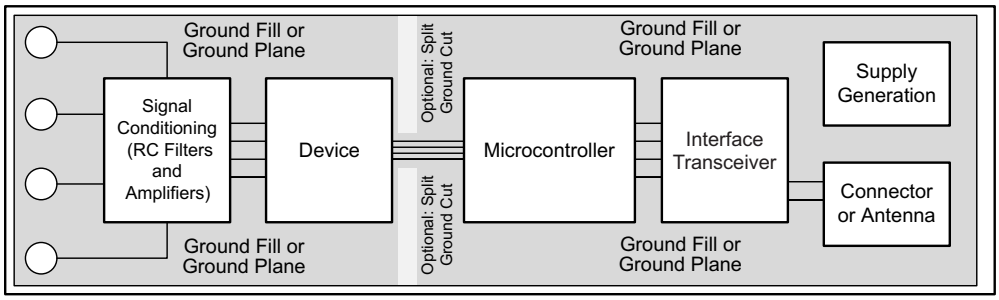


Figure 79. System Component Placement

The following outlines some basic recommendations for the layout of the ADS1299-x to get the best possible performance of the ADC. A good design can be ruined with a bad circuit layout.

- Separate analog and digital signals. To start, partition the board into analog and digital sections where the layout permits. Route digital lines away from analog lines. This configuration prevents digital noise from coupling back into analog signals.
- The ground plane can be split into an analog plane (AGND) and digital plane (DGND), but is not necessary. Place digital signals over the digital plane, and analog signals over the analog plane. As a final step in the layout, the split between the analog and digital grounds must be connected together at the ADC.
- Fill void areas on signal layers with ground fill.
- Provide good ground return paths. Signal return currents flow on the path of least impedance. If the ground plane is cut or has other traces that block the current from flowing right next to the signal trace, then the current must find another path to return to the source and complete the circuit. If current is forced into a longer path, the chances that the signal radiates increases. Sensitive signals are more susceptible to EMI interference.
- Use bypass capacitors on supplies to reduce high-frequency noise. Do not place vias between bypass capacitors and the active device. Placing the bypass capacitors on the same layer as close to the active device yields the best results.
- Analog inputs with differential connections must have a capacitor placed differentially across the inputs. The differential capacitors must be of high quality. The best ceramic chip capacitors are C0G (NPO), which have stable properties and low noise characteristics.

12.2 Layout Example

Figure 80 is an example layout of the ADS1299 requiring a minimum of two PCB layers. The example circuit is shown for either a single analog supply or a bipolar-supply connection. In this example, polygon pours are used as supply connections around the device. If a three- or four-layer PCB is used, the additional inner layers can be dedicated to route power traces. The PCB is partitioned with analog signals routed from the left, digital signals routed to the right, and power routed above and below the device.

Layout Example (continued)

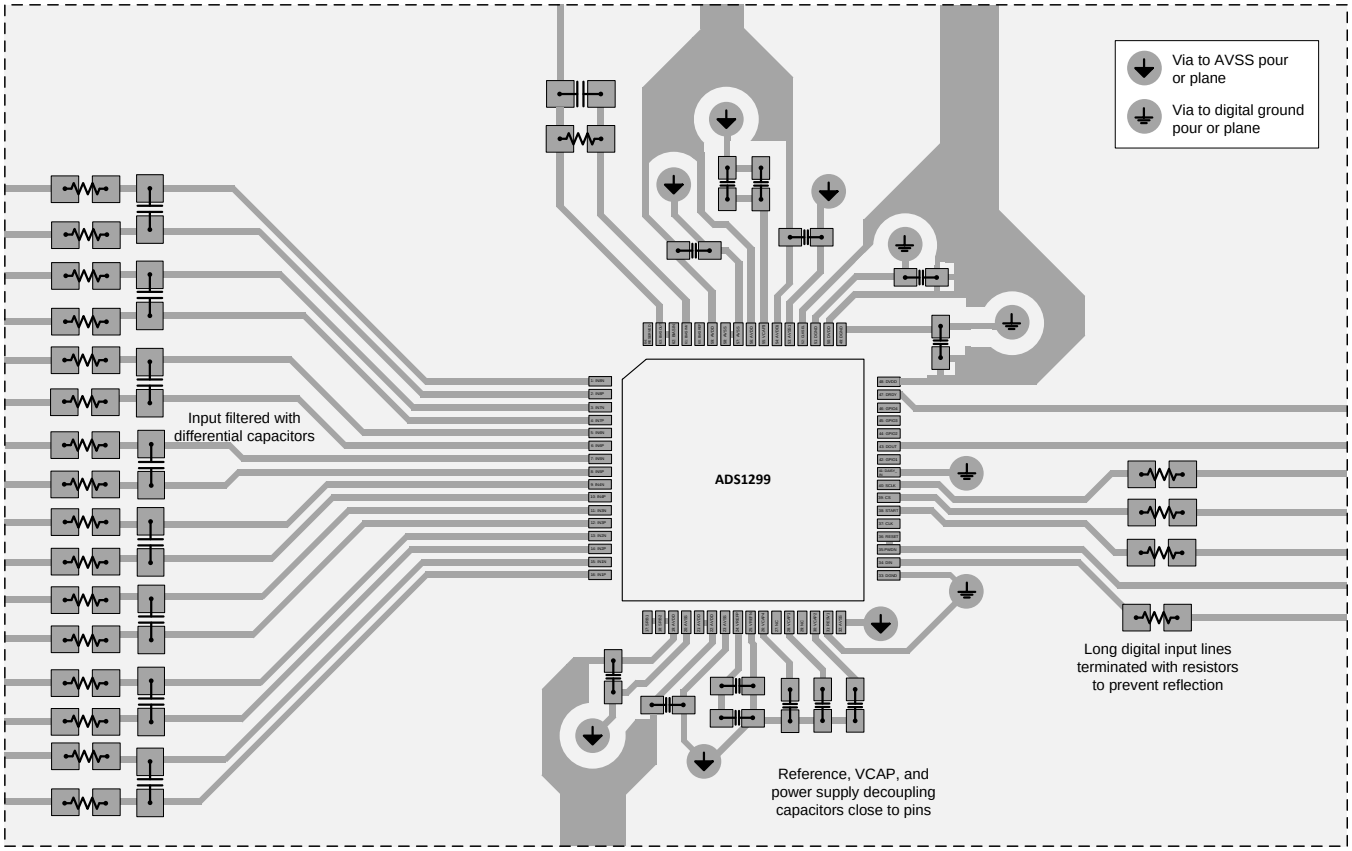


Figure 80. ADS1299 Example Layout

13 Device and Documentation Support

13.1 Documentation Support

13.1.1 Related Documentation

For related documentation see the following:

- *ADS129x Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements*
- *REF50xx Low-Noise, Very Low Drift, Precision Voltage Reference*
- *Improving Common-Mode Rejection Using the Right-Leg Drive Amplifier*
- *ADS1299EEG-FE EEG Front-End Performance Demonstration Kit*

13.2 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to order now.

Table 31. Related Links

PARTS	PRODUCT FOLDER	ORDER NOW	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
ADS1299	Click here	Click here	Click here	Click here	Click here
ADS1299-4	Click here	Click here	Click here	Click here	Click here
ADS1299-6	Click here	Click here	Click here	Click here	Click here

13.3 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. In the upper right corner, click on *Alert me* to register and receive a weekly digest of any product information that has changed. For change details, review the revision history included in any revised document.

13.4 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

TI E2E™ Online Community ***TI's Engineer-to-Engineer (E2E) Community.*** Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

Design Support ***TI's Design Support*** Quickly find helpful E2E forums along with design support tools and contact information for technical support.

13.5 Trademarks

E2E is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

13.6 Electrostatic Discharge Caution

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

13.7 Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

14 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
ADS1299-4PAG	ACTIVE	TQFP	PAG	64	160	RoHS & Green	NIPDAU	Level-3-260C-168 HR	-40 to 85	ADS1299-4	Samples
ADS1299-4PAGR	ACTIVE	TQFP	PAG	64	1500	RoHS & Green	NIPDAU	Level-3-260C-168 HR	-40 to 85	ADS1299-4	Samples
ADS1299-6PAG	ACTIVE	TQFP	PAG	64	160	RoHS & Green	NIPDAU	Level-3-260C-168 HR	-40 to 85	ADS1299-6	Samples
ADS1299-6PAGR	ACTIVE	TQFP	PAG	64	1500	RoHS & Green	NIPDAU	Level-3-260C-168 HR	-40 to 85	ADS1299-6	Samples
ADS1299IPAG	ACTIVE	TQFP	PAG	64	160	RoHS & Green	NIPDAU	Level-3-260C-168 HR	-40 to 85	ADS1299	Samples
ADS1299IPAGR	ACTIVE	TQFP	PAG	64	1500	RoHS & Green	NIPDAU	Level-3-260C-168 HR	-40 to 85	ADS1299	Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

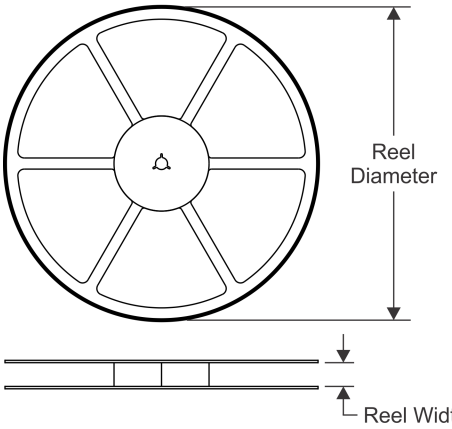
(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

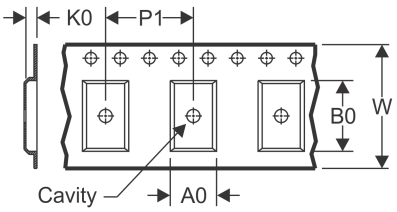
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

TAPE AND REEL INFORMATION

REEL DIMENSIONS

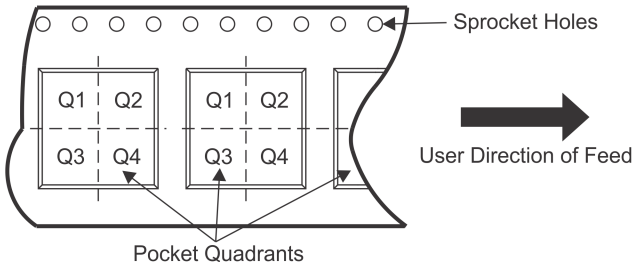


TAPE DIMENSIONS



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

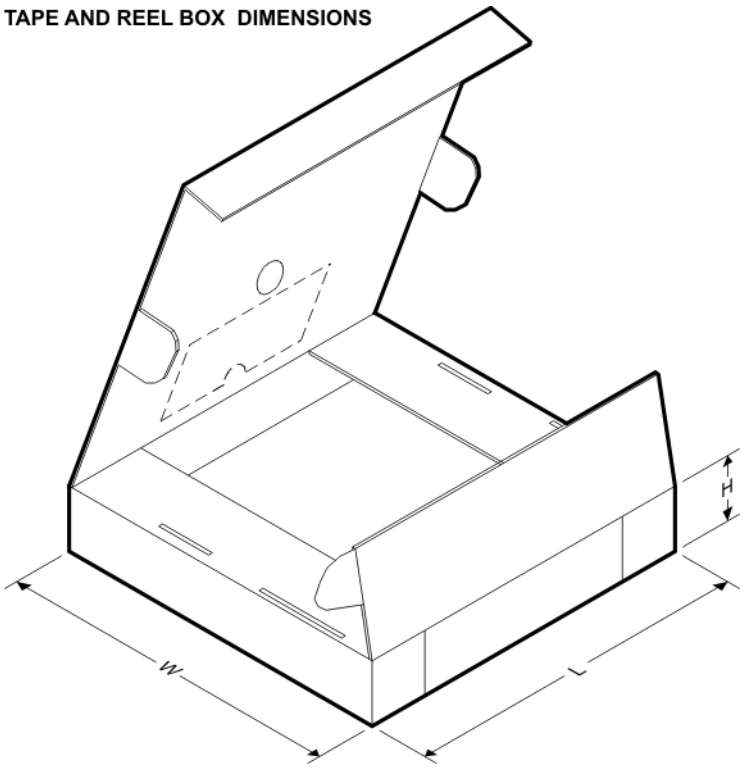
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
ADS1299-4PAGR	TQFP	PAG	64	1500	330.0	24.4	13.0	13.0	1.5	16.0	24.0	Q2
ADS1299-6PAGR	TQFP	PAG	64	1500	330.0	24.4	13.0	13.0	1.5	16.0	24.0	Q2
ADS1299IPAGR	TQFP	PAG	64	1500	330.0	24.4	13.0	13.0	1.5	16.0	24.0	Q2

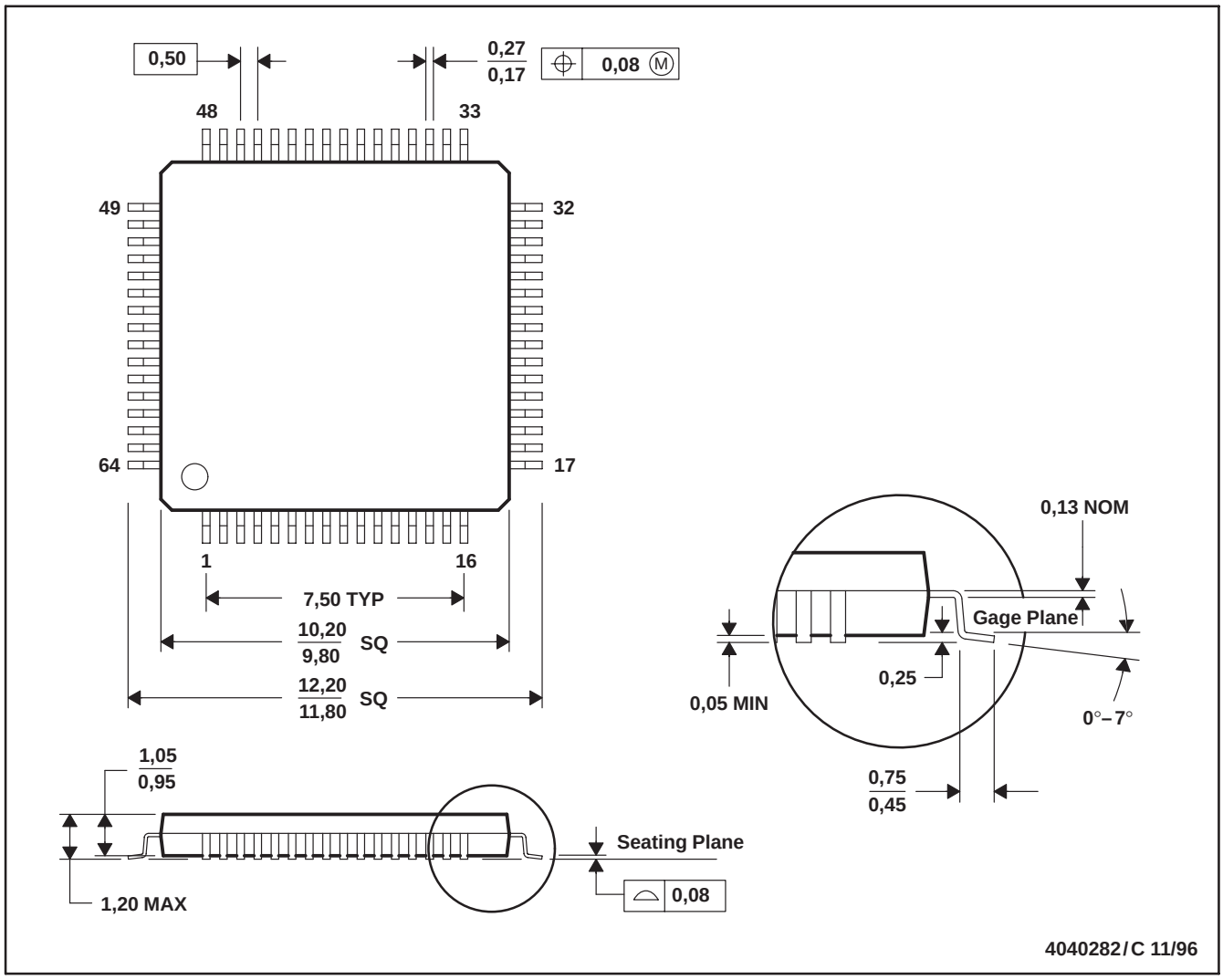
TAPE AND REEL BOX DIMENSIONS



*All dimensions are nominal

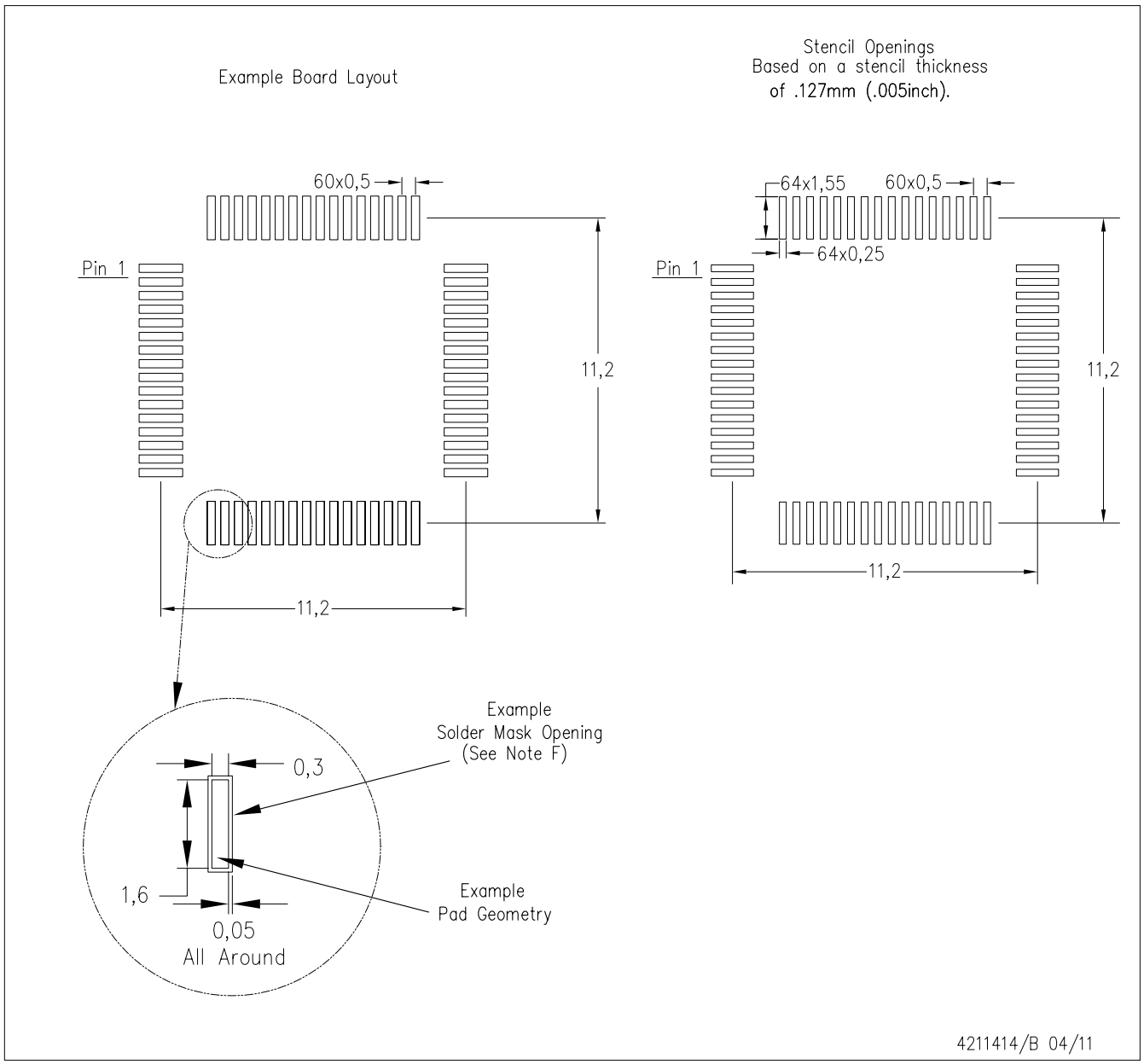
Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
ADS1299-4PAGR	TQFP	PAG	64	1500	350.0	350.0	43.0
ADS1299-6PAGR	TQFP	PAG	64	1500	350.0	350.0	43.0
ADS1299IPAGR	TQFP	PAG	64	1500	350.0	350.0	43.0

PAG (S-PQFP-G64) **PLASTIC QUAD FLATPACK**



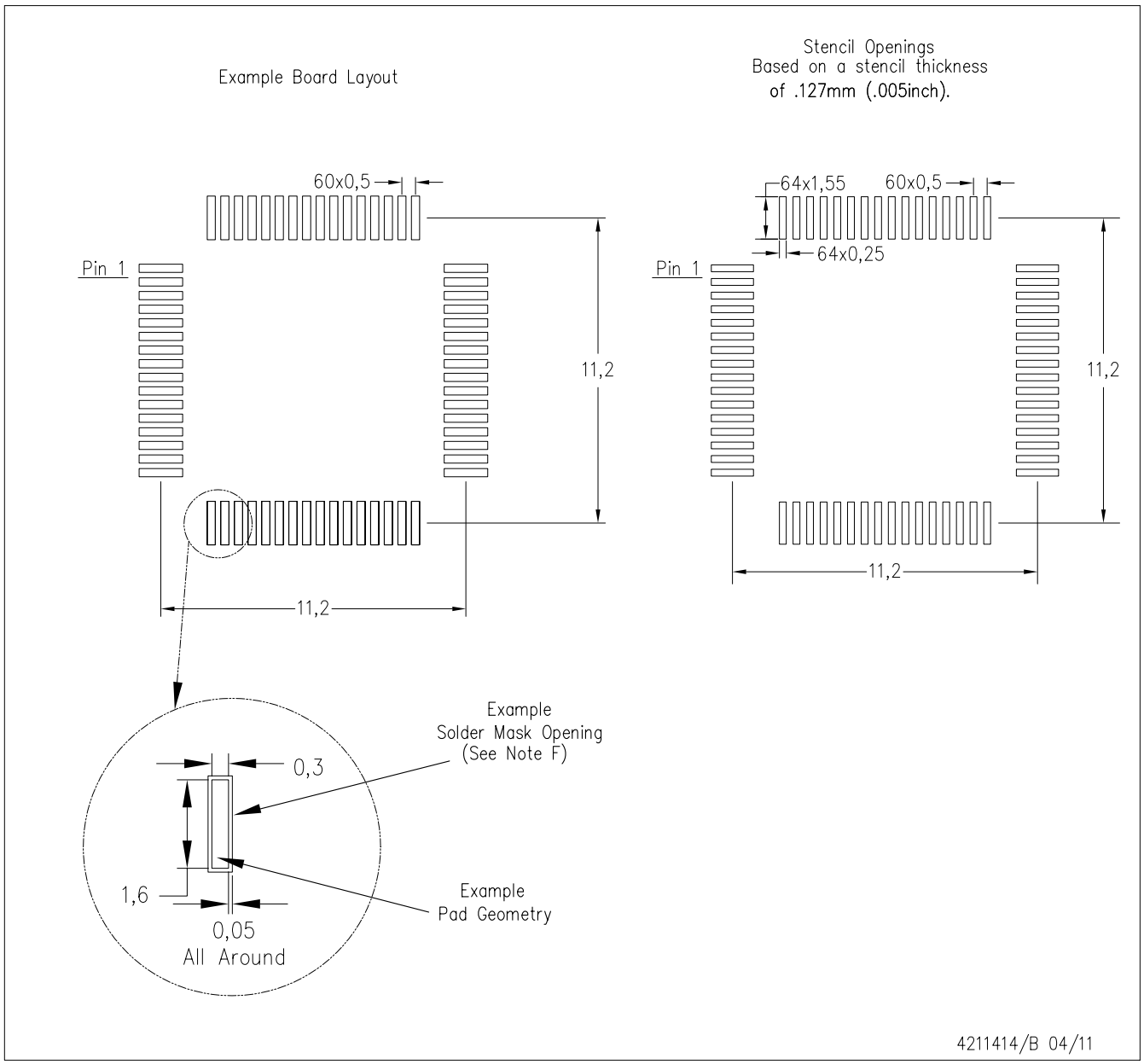
NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026

PAG (S–PQFP–G64) PLASTIC QUAD FLATPACK



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Example stencil design based on a 50% volumetric metal load solder paste. Refer to IPC–7525 for other stencil recommendations.
D. Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

PAG (S–PQFP–G64) PLASTIC QUAD FLATPACK



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Example stencil design based on a 50% volumetric metal load solder paste. Refer to IPC–7525 for other stencil recommendations.
D. Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

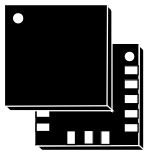
TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

LIS3DH

MEMS digital output motion sensor: ultra-low-power high-performance 3-axis "nano" accelerometer

Datasheet - production data



LGA-16 (3x3x1 mm)

Features

- Wide supply voltage, 1.71 V to 3.6 V
- Independent IO supply (1.8 V) and supply voltage compatible
- Ultra-low-power mode consumption down to 2 µA
- ±2*g*/±4*g*/±8*g*/±16*g* dynamically selectable full scale
- I^2C/SPI digital output interface
- 16-bit data output
- 2 independent programmable interrupt generators for free-fall and motion detection
- 6D/4D orientation detection
- Free-fall detection
- Motion detection
- Embedded temperature sensor
- Embedded self-test
- Embedded 32 levels of 16-bit data output FIFO
- 10000 *g* high shock survivability
- ECOPACK®, RoHS and "Green" compliant

Applications

- Motion activated functions
- Free-fall detection
- Click/double-click recognition
- Intelligent power saving for handheld devices
- Pedometers
- Display orientation
- Gaming and virtual reality input devices
- Impact recognition and logging
- Vibration monitoring and compensation

Description

The LIS3DH is an ultra-low-power high-performance three-axis linear accelerometer belonging to the "nano" family, with digital I^2C/SPI serial interface standard output. The device features ultra-low-power operational modes that allow advanced power saving and smart embedded functions.

The LIS3DH has dynamically user-selectable full scales of ±2*g*/±4*g*/±8*g*/±16*g* and is capable of measuring accelerations with output data rates from 1 Hz to 5.3 kHz. The self-test capability allows the user to check the functioning of the sensor in the final application. The device may be configured to generate interrupt signals using two independent inertial wake-up/free-fall events as well as by the position of the device itself. Thresholds and timing of interrupt generators are programmable by the end user on the fly. The LIS3DH has an integrated 32-level first-in, first-out (FIFO) buffer allowing the user to store data in order to limit intervention by the host processor. The LIS3DH is available in small thin plastic land grid array package (LGA) and is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

Table 1. Device summary

Order codes	Temp. range [°C]	Package	Packaging
LIS3DHTR	-40 to +85	LGA-16	Tape and reel

Contents

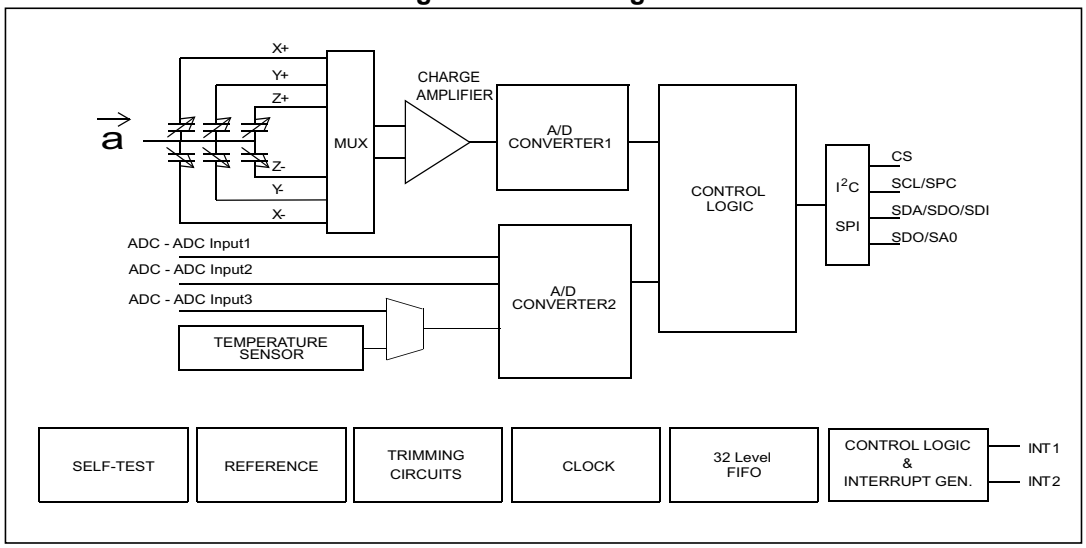
List of tables

List of figures

1 Block diagram and pin description

1.1 Block diagram

Figure 1. Block diagram



1.2 Pin description

Figure 2. Pin connections

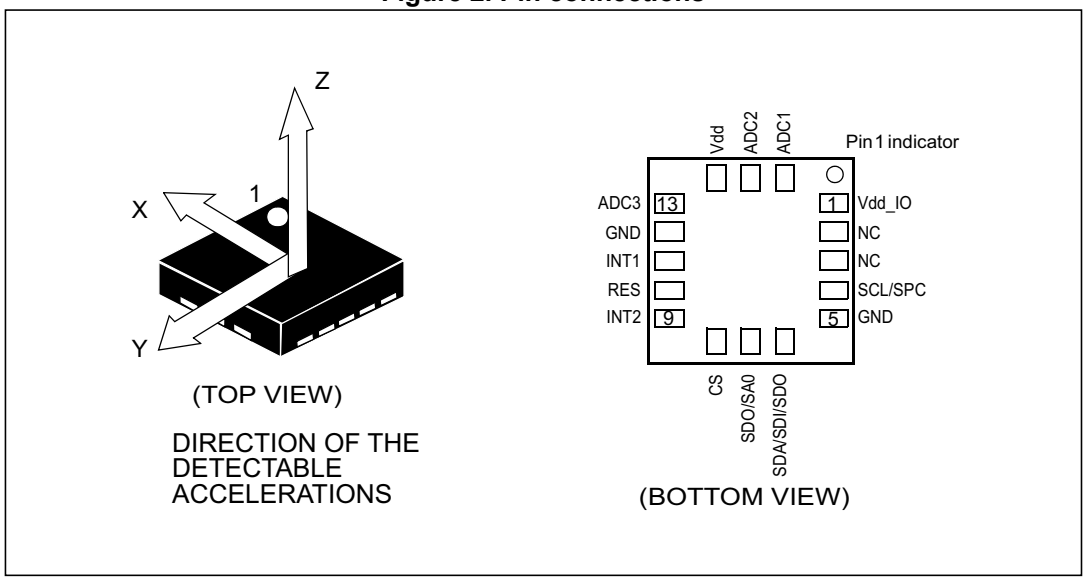


Table 2. Pin description

Pin#	Name	Function
1	Vdd_IO	Power supply for I/O pins
2	NC	Not connected
3	NC	Not connected
4	SCL SPC	I^2C serial clock (SCL) SPI serial port clock (SPC)
5	GND	0 V supply
6	SDA SDI SDO	I^2C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)
7(1)	SDO SA0	SPI serial data output (SDO) I^2C less significant bit of the device address (SA0)
8	CS	SPI enable I^2C/SPI mode selection: 1: SPI idle mode / I^2C communication enabled 0: SPI communication mode / I^2C disabled
9	INT2	Inertial interrupt 2
10	RES	Connect to GND
11	INT1	Inertial interrupt 1
12	GND	0 V supply
13	ADC3	Analog-to-digital converter input 3
14	Vdd	Power supply
15	ADC2	Analog-to-digital converter input 2
16	ADC1	Analog-to-digital converter input 1

1. SDO/SA0 pin is internally pulled up. Refer to *Table 3* for the internal pull-up values (typ.).

Table 3. Internal pull-up values (typ.) for SDO/SA0 pin

Vdd_IO	Resistor value for SDO/SA0 pin
	Typ. (kΩ)
1.7 V	54.4
1.8 V	49.2
2.5 V	30.4
3.6 V	20.4

2 Mechanical and electrical specifications

2.1 Mechanical characteristics

Vdd = 2.5 V, T = 25 °C unless otherwise noted [a]

Table 4. Mechanical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
FS	Measurement range[2]	FS bit set to 00		±2.0		
		FS bit set to 01		±4.0		
		FS bit set to 10		±8.0		
		FS bit set to 11		±16.0		*g*
So	Sensitivity	FS bit set to 00; High-resolution mode		1		mg/digit
		FS bit set to 00; Normal mode		4		
		FS bit set to 00; Low-power mode		16		
		FS bit set to 01; High-resolution mode		2		mg/digit
		FS bit set to 01; Normal mode		8		
		FS bit set to 01; Low-power mode		32		
		FS bit set to 10; High-resolution mode		4		mg/digit
		FS bit set to 10; Normal mode		16		
		FS bit set to 10; Low-power mode		64		
		FS bit set to 11; High-resolution mode		12		mg/digit
		FS bit set to 11; Normal mode		48		
		FS bit set to 11; Low-power mode		192		
TCSo	Sensitivity change vs temperature	FS bit set to 00		0.01		%/°C
TyOff	Typical zero-*g* level offset accuracy[3],[4]	FS bit set to 00		±40		m*g*

a. The product is factory calibrated at 2.5 V. The operational power supply range is from 1.71 V to 3.6 V.

Table 4. Mechanical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.(1)	Max.	Unit
TCOff	Zero-*g* level change vs temperature	Max delta from 25 °C		±0.5		m*g*/°C
An	Acceleration noise density	FS bit set to 00, High-Resolution mode (*Table 10*), ODR > 1300 Hz		220		µ*g*/√Hz
Vst	Self-test output change(5)(6)(7)	FS bit set to 00 X-axis; Normal mode	17		360	LSb
		FS bit set to 00 Y-axis; Normal mode	17		360	LSb
		FS bit set to 00 Z-axis; Normal mode	17		360	LSb
Top	Operating temperature range		-40		+85	°C

1. Typical specifications are not guaranteed.
2. Verified by wafer level test and measurement of initial offset and sensitivity.
3. Typical zero-*g* level offset value after MSL3 preconditioning.
4. Offset can be eliminated by enabling the built-in high-pass filter.
5. The sign of “Self-test output change” is defined by the ST bits in *CTRL_REG4 (23h)*, for all axes.
6. “Self-test output change” is defined as the absolute value of: $OUTPUT[LSb]_{(Self\ test\ enabled)}$ - $OUTPUT[LSb]_{(Self\ test\ disabled)}$. 1LSb = 4 m*g* at 10-bit representation, ±2 *g* full scale.
7. After enabling the self-test, correct data is obtained after two samples (low-power mode / normal mode) or after eight samples (high-resolution mode).

2.2 Temperature sensor characteristics

Vdd = 2.5 V, T = 25 °C unless otherwise noted [b]

Table 5. Temperature sensor characteristics

Symbol	Parameter	Test condition	Min.	Typ.(1)	Max.	Unit
TSDr	Temperature sensor output change vs temperature			1		digit/°C(2)
TODR	Temperature refresh rate			ODR		Hz
Top	Operating temperature range		-40		+85	°C

1. Typical specifications are not guaranteed.
2. 8-bit resolution.

2.3 Electrical characteristics

Vdd = 2.5 V, T = 25 °C unless otherwise noted [c]

Table 6. Electrical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.(1)	Max.	Unit
Vdd	Supply voltage		1.71	2.5	3.6	V
Vdd_IO	I/O pins supply voltage(2)		1.71		Vdd+0.1	V
Idd	Current consumption in normal mode	50 Hz ODR		11		µA
Idd	Current consumption in normal mode	1 Hz ODR		2		µA
IddLP	Current consumption in low-power mode	50 Hz ODR		6		µA
IddPdn	Current consumption in power-down mode			0.5		µA
VIH	Digital high-level input voltage		0.8*Vdd_IO			V
VIL	Digital low-level input voltage				0.2*Vdd_IO	V
VOH	High-level output voltage		0.9*Vdd_IO			V
VOL	Low-level output voltage				0.1*Vdd_IO	V
BW	System bandwidth(3)			ODR/2		Hz
Top	Operating temperature range		-40		+85	°C

1. Typical specification are not guaranteed.
2. It is possible to remove Vdd maintaining Vdd_IO without blocking the communication busses, in this condition the measurement chain is powered off.
3. Refer to *Table 25* for the ODR value and configuration.

b. The product is factory calibrated at 2.5 V. Temperature sensor operation is guaranteed in the range 2 V - 3.6 V.

c. The product is factory calibrated at 2.5 V. The operational power supply range is from 1.71 V to 3.6 V.

2.4 Communication interface characteristics

2.4.1 SPI - serial peripheral interface

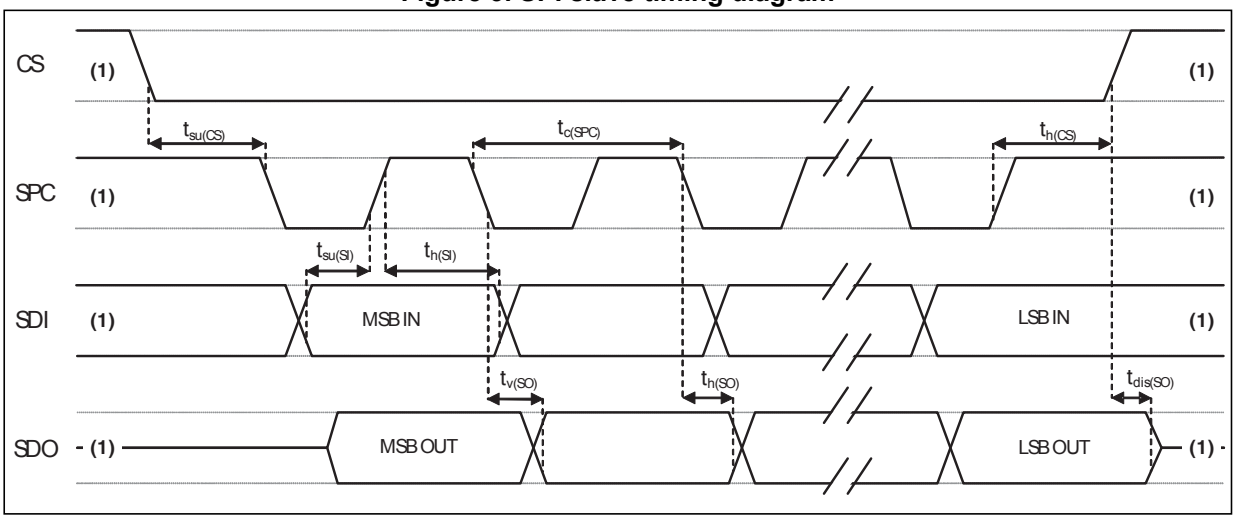
Subject to general operating conditions for Vdd and Top.

Table 7. SPI slave timing values

Symbol	Parameter	Value (1)		Unit
		Min	Max	
$t_{c(SPC)}$	SPI clock cycle	100		ns
$f_{c(SPC)}$	SPI clock frequency		10	MHz
$t^{su(CS)}$	CS setup time	5		ns
$t_{h(CS)}$	CS hold time	20		
$t_{su(SI)}$	SDI input setup time	5		
$t_{h(SI)}$	SDI input hold time	15		
$t_{v(SO)}$	SDO valid output time		50	
$t_{h(SO)}$	SDO output hold time	5		
$t_{dis(SO)}$	SDO output disable time		50	

1. Values are guaranteed at 10 MHz clock frequency for SPI with both 4 and 3 wires, based on characterization results, not tested in production.

Figure 3. SPI slave timing diagram



1. When no communication is ongoing, data on SDO is driven by internal pull-up resistors.

Note: Measurement points are done at 0.2·Vdd_IO and 0.8·Vdd_IO, for both input and output ports.

2.4.2 I²C - Inter IC control interface

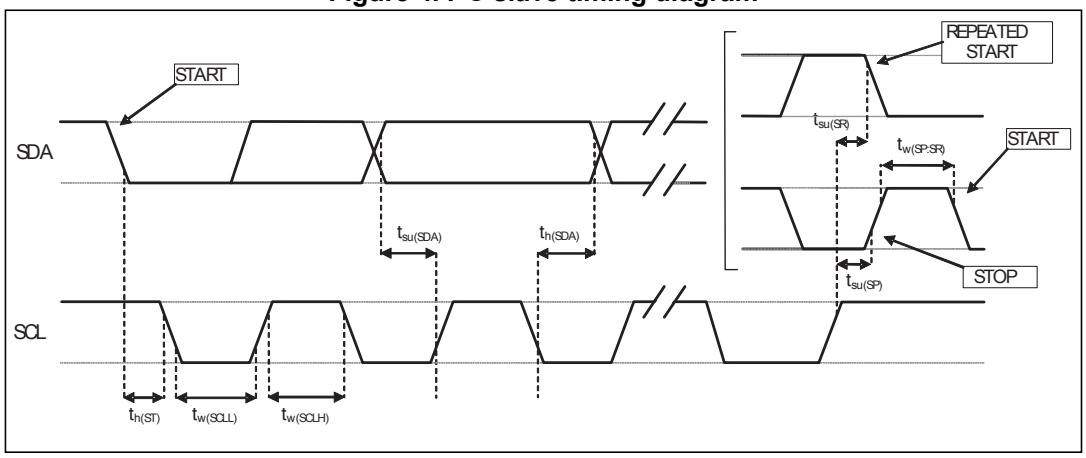
Subject to general operating conditions for Vdd and top.

Table 8. I²C slave timing values

Symbol	Parameter	I²C standard		I²C fast mode (1)		Unit
		Min	Max	Min	Max	
$f_{(SCL)}$	SCL clock frequency	0	100	0	400	kHz
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		µs
$t_{w(SCLH)}$	SCL clock high time	4.0		0.6		
$t_{su(SDA)}$	SDA setup time	250		100		ns
$t_{h(SDA)}$	SDA data hold time	0	3.45	0	0.9	µs
$t_{h(ST)}$	START condition hold time	4		0.6		µs
$t_{su(SR)}$	Repeated START condition setup time	4.7		0.6		
$t_{su(SP)}$	STOP condition setup time	4		0.6		
$t_{w(SP:SR)}$	Bus free time between STOP and START condition	4.7		1.3		

1. Data based on standard I²C protocol requirement, not tested in production.

Figure 4. I²C slave timing diagram



Note: Measurement points are done at 0.2·Vdd_IO and 0.8·Vdd_IO, for both ports.

2.5 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 9. Absolute maximum ratings

Symbol	Ratings	Maximum value	Unit
Vdd	Supply voltage	-0.3 to 4.8	V
Vdd_IO	I/O pins Supply voltage	-0.3 to 4.8	V
Vin	Input voltage on any control pin (CS, SCL/SPC, SDA/SDI/SDO, SDO/SA0)	-0.3 to Vdd_IO +0.3	V
A_{POW}	Acceleration (any axis, powered, Vdd = 2.5 V)	3000 *g* for 0.5 ms	
		10000 *g* for 0.2 ms	
A_{UNP}	Acceleration (any axis, unpowered)	3000 *g* for 0.5 ms	
		10000 *g* for 0.2 ms	
T_{OP}	Operating temperature range	-40 to +85	°C
T_{STG}	Storage temperature range	-40 to +125	°C
ESD	Electrostatic discharge protection	2 (HBM)	kV

Note: Supply voltage on any pin should never exceed 4.8 V

This device is sensitive to mechanical shock, improper handling can cause permanent damage to the part.

This device is sensitive to electrostatic discharge (ESD), improper handling can cause permanent damage to the part.

3 Terminology and functionality

3.1 Terminology

3.1.1 Sensitivity

Sensitivity describes the gain of the sensor and can be determined, for example, by applying 1 *g* acceleration to it. As the sensor can measure DC accelerations this can be done easily by pointing the axis of interest towards the center of the Earth, noting the output value, rotating the sensor by 180 degrees (pointing to the sky) and noting the output value again. By doing so, ±1 *g* acceleration is applied to the sensor. Subtracting the larger output value from the smaller one, and dividing the result by 2, leads to the actual sensitivity of the sensor. This value changes very little over temperature and also time. The sensitivity tolerance describes the range of sensitivities of a large population of sensors.

3.1.2 Zero-*g* level

The zero-*g* level offset (TyOff) describes the deviation of an actual output signal from the ideal output signal if no acceleration is present. A sensor in a steady state on a horizontal surface measures 0 *g* for the X-axis and 0 *g* for the Y-axis whereas the Z-axis measures 1 *g*. The output is ideally in the middle of the dynamic range of the sensor (content of OUT registers 00h, data expressed as 2's complement number). A deviation from ideal value in this case is called Zero-*g* offset. Offset is to some extent a result of stress to MEMS sensor and therefore the offset can slightly change after mounting the sensor onto a printed circuit board or exposing it to extensive mechanical stress. Offset changes little over temperature, see *Table 4* "Zero-*g* level change vs. temperature" (TCOff). The zero-*g* level tolerance (TyOff) describes the standard deviation of the range of zero-*g* levels of a population of sensors.

3.2 Functionality

3.2.1 High-resolution, normal mode, low-power mode

LIS3DH provides three different operating modes: *high-resolution mode*, *normal mode* and *low-power mode*.

The table below summarizes how to select the operating mode.

Table 10. Operating mode selection

Operating mode	CTRL_REG1[3] (LPen bit)	CTRL_REG4[3] (HR bit)	BW [Hz]	Turn-on time [ms]	So @ ±2*g* [mg/digit]
Low-power mode (8-bit data output)	1	0	ODR/2	1	16
Normal mode (10-bit data output)	0	0	ODR/2	1.6	4
High-resolution mode (12-bit data output)	0	1	ODR/9	7/ODR	1
Not allowed	1	1	--	--	--

The turn-on time to transition to another operating mode is given in *Table 11*.

Table 11. Turn-on time for operating mode transition

Operating mode change	Turn-on time [ms]
12-bit mode to 8-bit mode	1/ODR
12-bit mode to 10-bit mode	1/ODR
10-bit mode to 8-bit mode	1/ODR
10-bit mode to 12-bit mode	7/ODR
8-bit mode to 10-bit mode	1/ODR
8-bit mode to 12-bit mode	7/ODR

Table 12. Current consumption of operating modes

Operating mode [Hz]	Low-power mode (8-bit data output) [µA]	Normal mode (10-bit data output) [µA]	High resolution (12-bit data output) [µA]
1	2	2	2
10	3	4	4
25	4	6	6
50	6	11	11
100	10	20	20
200	18	38	38
400	36	73	73
1344	--	185	185
1620	100	--	--
5376	185	--	--

3.2.2 Self-test

The self-test allows the user to check the sensor functionality without moving it. The self-test function is off when the self-test bit (ST) is programmed to ‘0‘. When the self-test bit is programmed to ‘1‘, an actuation force is applied to the sensor, simulating a definite input acceleration. In this case the sensor outputs exhibit a change in their DC levels which are related to the selected full scale through the device sensitivity. When the self-test is activated, the device output level is given by the algebraic sum of the signals produced by the acceleration acting on the sensor and by the electrostatic test-force. If the output signals change within the amplitude specified inside *Table 4*, then the sensor is working properly and the parameters of the interface chip are within the defined specifications.

3.2.3 6D / 4D orientation detection

The LIS3DH provides the capability to detect the orientation of the device in space, enabling easy implementation of energy-saving procedures and automatic image rotation for mobile devices.

The 4D detection is a subset of the 6D function especially defined to be implemented in mobile devices for portrait and landscape computation. In 4D configuration, the Z-axis position detection is disabled.

3.2.4 “Sleep-to-wake” and “Return-to-sleep”

The LIS3DH can be programmed to automatically switch to low-power mode upon recognition of a determined event.
Once the event condition is over, the device returns back to the preset normal or high-resolution mode.

To enable this function the desired threshold value must be stored inside the *ACT_THS (3Eh)* register while the duration value is written inside the *ACT_DUR (3Fh)* register.

When the acceleration falls below the threshold value, the device automatically switches to low-power mode (10 Hz ODR).

During this condition, the ODR[3:0] bits and the LPen bit inside *CTRL_REG1 (20h)* and the HR bit in *CTRL_REG4 (23h)* are not considered.

As soon as the acceleration rises above threshold, the module restores the operating mode and ODRs as determined by the *CTRL_REG1 (20h)* and *CTRL_REG4 (23h)* settings.

3.3 Sensing element

A proprietary process is used to create a surface micro-machined accelerometer. The technology allows carrying out suspended silicon structures which are attached to the substrate in a few points called anchors and are free to move in the direction of the sensed acceleration. To be compatible with the traditional packaging techniques a cap is placed on top of the sensing element to avoid blocking the moving parts during the moulding phase of the plastic encapsulation.

When an acceleration is applied to the sensor the proof mass displaces from its nominal position, causing an imbalance in the capacitive half-bridge. This imbalance is measured using charge integration in response to a voltage pulse applied to the capacitor.

At steady state the nominal value of the capacitors are few pF and when an acceleration is applied the maximum variation of the capacitive load is in the fF range.

3.4 IC interface

The complete measurement chain is composed by a low-noise capacitive amplifier which converts the capacitive unbalancing of the MEMS sensor into an analog voltage that is finally available to the user by an analog-to-digital converter.

The acceleration data may be accessed through an I^2C/SPI interface thus making the device particularly suitable for direct interfacing with a microcontroller.

The LIS3DH features a Data-Ready signal (DRDY) which indicates when a new set of measured acceleration data is available, thus simplifying data synchronization in the digital system that uses the device.

The LIS3DH may also be configured to generate an inertial wake-up and free-fall interrupt signal accordingly to a programmed acceleration event along the enabled axes. Both free-fall and wake-up can be available simultaneously on two different pins.

3.5 Factory calibration

The IC interface is factory calibrated for sensitivity (So) and Zero-*g* level (TyOff).

The trim values are stored inside the device in non-volatile memory. Any time the device is turned on, these values are downloaded into the registers to be used during active operation. This allows using the device without further calibration.

3.6 FIFO

The LIS3DH contains a 10-bit, 32-level FIFO. Buffered output allows 4 operation modes: FIFO, Stream, Stream-to-FIFO and FIFO bypass. When FIFO bypass mode is activated, FIFO is not operating and remains empty. In FIFO mode, measurement data from acceleration detection on the x, y, and z axes are stored in the FIFO buffer.

3.7 Auxiliary ADC and temperature sensor

The LIS3DH contains an auxiliary ADC with 3 separate dedicated inputs: pins ADC1, ADC2, ADC3.

The user can retrieve the converted data from registers *OUT_ADC1_L (08h), OUT_ADC1_H (09h)*, *OUT_ADC2_L (0Ah), OUT_ADC2_H (0Bh)* and *OUT_ADC3_L (0Ch), OUT_ADC3_H (0Dh)*.

In order to use the auxiliary ADC, the user must set the BDU bit (bit 7) to 1 in *CTRL_REG4 (23h)* and the ADC_EN bit (bit 7) to 1 in *TEMP_CFG_REG (1Fh)*. The ADC sampling frequency is the same as that of the ODR in *CTRL_REG1 (20h)*.

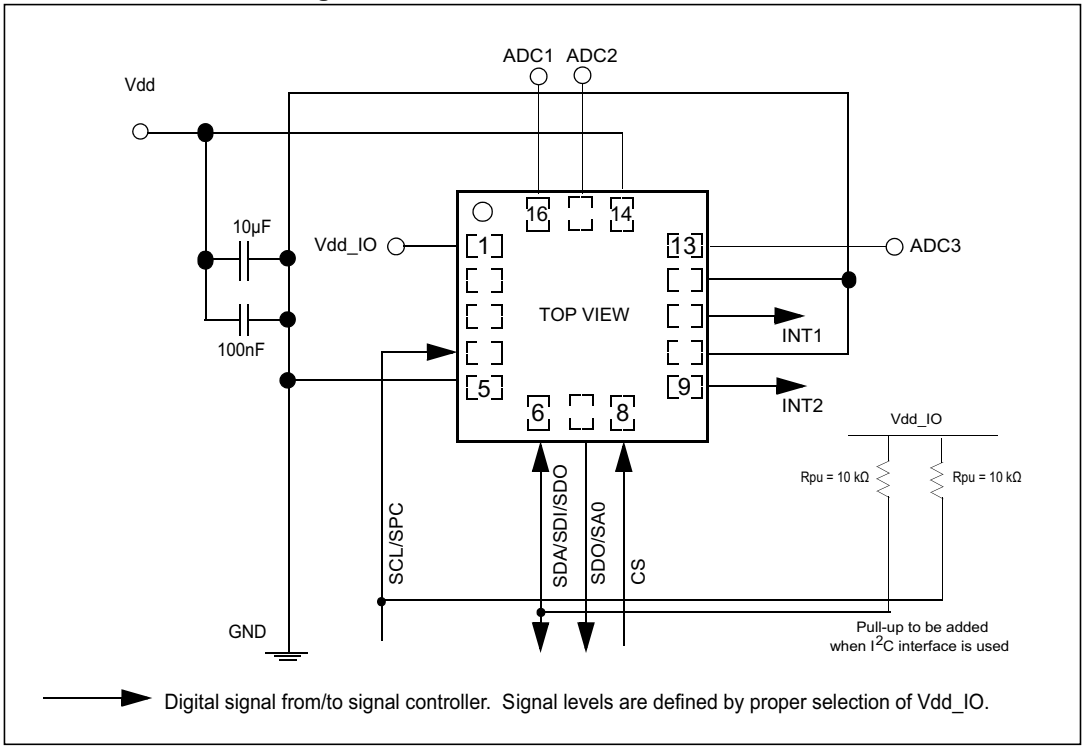
The input range is 1200 mv ±400 mV and the data output is expressed in 2's complement left-aligned.

The ADC resolution is 10 bits if the LPen (bit 3) in *CTRL_REG1 (20h)* is cleared (high-resolution / normal mode), otherwise, in low-power mode, the ADC resolution is 8-bit.

Channel 3 of the ADC can be connected to the temperature sensor by setting the TEMP_EN bit (bit 6) to 1 in *TEMP_CFG_REG (1Fh)*. Refer to *Table 5: Temperature sensor characteristics* for the conversion factor.

4 Application hints

Figure 5. LIS3DH electrical connections



The device core is supplied through the Vdd line while the I/O pads are supplied through the Vdd_IO line. Power supply decoupling capacitors (100 nF ceramic, 10 µF aluminum) should be placed as near as possible to pin 14 of the device (common design practice).

All the voltage and ground supplies must be present at the same time to have proper behavior of the IC (refer to *Figure 5*). It is possible to remove Vdd maintaining Vdd_IO without blocking the communication bus, in this condition the measurement chain is powered off.

The functionality of the device and the measured acceleration data is selectable and accessible through the I^2C or SPI interfaces.When using the I^2C, CS must be tied high.

ADC1, ADC2 & ADC3 if not used can be left floating or connected to Vdd or GND.

The functions, the threshold and the timing of the two interrupt pins (INT1 and INT2) can be completely programmed by the user through the I^2C/SPI interface.

Table 13. Internal pin status

Pin#	Name	Function	Pin status
1	Vdd_IO	Power supply for I/O pins	
2	NC	Not connected	
3	NC	Not connected	
4	SCL SPC	I^2C serial clock (SCL) SPI serial port clock (SPC)	Default: input high impedance
5	GND	0 V supply	
6	SDA SDI SDO	I^2C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)	Default: (SDA) input high impedance
7	SDO SA0	SPI serial data output (SDO) I^2C less significant bit of the device address (SA0)	Default: input with internal pull-up(1)
8	CS	SPI enable I^2C/SPI mode selection: 1: SPI idle mode / I^2C communication enabled 0: SPI communication mode / I^2C disabled	Default: input high impedance
9	INT2	Inertial interrupt 2	Default: push-pull output forced to GND
10	RES	Connect to GND	
11	INT1	Inertial interrupt 1	Default: push-pull output forced to GND
12	GND	0 V supply	
13	ADC3	Analog-to-digital converter input 3	Default: input high impedance
14	Vdd	Power supply	
15	ADC2	Analog-to-digital converter input 2	Default: input high impedance
16	ADC1	Analog-to-digital converter input 1	Default: input high impedance

1. In order to disable the internal pull-up on the SDO/SA0 pin, write 90h in *CTRL_REG0 (1Eh)*.

4.1 Soldering information

The LGA package is compliant with the ECOPACK®, RoHS and “Green” standard.
It is qualified for soldering heat resistance according to JEDEC J-STD-020.

Leave “Pin 1 Indicator” unconnected during soldering.

Land pattern and soldering recommendations are available at www.st.com.

5 Digital main blocks

5.1 FIFO

The LIS3DH embeds a 32-level FIFO for each of the three output channels, X, Y and Z.

This allows consistent power saving for the system, since the host processor does not need to continuously poll data from the sensor, but it can wake up only when needed and burst the significant data out from the FIFO.

In order to enable the FIFO buffer, the FIFO_EN bit in *CTRL_REG5 (24h)* must be set to '1'.

This buffer can work according to the following different modes: Bypass mode, FIFO mode, Stream mode and Stream-to-FIFO mode. Each mode is selected by the FM [1:0] bits in *FIFO_CTRL_REG (2Eh)*. Programmable FIFO watermark level, FIFO empty or FIFO overrun events can be enabled to generate dedicated interrupts on the INT1 pin (configuration through *CTRL_REG3 (22h)*).

In the *FIFO_SRC_REG (2Fh)* register the EMPTY bit is equal to '1' when all FIFO samples are ready and FIFO is empty.

In the *FIFO_SRC_REG (2Fh)* register the WTM bit goes to '1' if new data is written in the buffer and *FIFO_SRC_REG (2Fh)* (FSS [4:0]) is greater than or equal to *FIFO_CTRL_REG (2Eh)* (FTH [4:0]). *FIFO_SRC_REG (2Fh)* (WTM) goes to '0' if reading an X, Y, Z data slot from FIFO and *FIFO_SRC_REG (2Fh)* (FSS [4:0]) is less than or equal to *FIFO_CTRL_REG (2Eh)* (FTH [4:0]).

In the *FIFO_SRC_REG (2Fh)* register the OVRN_FIFO bit is equal to '1' if the FIFO slot is overwritten.

5.1.1 Bypass mode

In Bypass mode the FIFO is not operational and for this reason it remains empty. For each channel only the first address is used. The remaining FIFO levels are empty.

Bypass mode must be used in order to reset the FIFO buffer when a different mode is operating (i.e. FIFO mode).

5.1.2 FIFO mode

In FIFO mode, the buffer continues filling data from the X, Y and Z accelerometer channels until it is full (a set of 32 samples stored). When the FIFO is full, it stops collecting data from the input channels and the FIFO content remains unchanged.

An overrun interrupt can be enabled, I1_OVERRUN = '1' in the *CTRL_REG3 (22h)* register, in order to be raised when the FIFO stops collecting data. When the overrun interrupt occurs, the first data has been overwritten and the FIFO stops collecting data from the input channels.

After the last read it is necessary to exit Bypass mode in order to reset the FIFO content. After this reset command, it is possible to restart FIFO mode just by selecting the FIFO mode configuration (FM[1:0] bits) in register *FIFO_CTRL_REG (2Eh)*.

5.1.3 Stream mode

In Stream mode the FIFO continues filling data from the X, Y, and Z accelerometer channels until the buffer is full (a set of 32 samples stored) at which point the FIFO buffer index restarts from the beginning and older data is replaced by the current data. The oldest values continue to be overwritten until a read operation frees the FIFO slots.

An overrun interrupt can be enabled, I1_OVERRUN = '1' in the *CTRL_REG3 (22h)* register, in order to read the entire contents of the FIFO at once. If, in the application, it is mandatory not to lose data and it is not possible to read at least one sample for each axis within one ODR period, a watermark interrupt can be enabled in order to read partially the FIFO and leave memory slots free for incoming data.

Setting the FTH [4:0] bit in the *FIFO_CTRL_REG (2Eh)* register to an N value, the number of X, Y and Z data samples that should be read at the rise of the watermark interrupt is up to (N+1).

5.1.4 Stream-to-FIFO mode

In Stream-to-FIFO mode, data from the X, Y and Z accelerometer channels are collected in a combination of Stream mode and FIFO mode. The FIFO buffer starts operating in Stream mode and switches to FIFO mode when the selected interrupt occurs.

The FIFO operating mode changes according to the INT1 pin value if the TR bit is set to ‘0’ in the *FIFO_CTRL_REG (2Eh)* register or the INT2 pin value if the TR bit is set to‘1’ in the *FIFO_CTRL_REG (2Eh)* register.

When the interrupt pin is selected and the interrupt event is configured on the corresponding pin, the FIFO operates in Stream mode if the pin value is equal to ‘0’ and it operates in FIFO mode if the pin value is equal to ‘1’. Switching modes is dynamically performed according to the pin value.

Stream-to-FIFO can be used in order to analyze the sampling history that generates an interrupt. The standard operation is to read the contents of FIFO when the FIFO mode is triggered and the FIFO buffer is full and stopped.

5.1.5 Retrieving data from FIFO

FIFO data is read from *OUT_X_L (28h), OUT_X_H (29h)*, *OUT_Y_L (2Ah), OUT_Y_H (2Bh)* and *OUT_Z_L (2Ch), OUT_Z_H (2Dh)*. When the FIFO is in Stream, Stream-to-FIFO or FIFO mode, a read operation from the *OUT_X_L (28h), OUT_X_H (29h)*, *OUT_Y_L (2Ah), OUT_Y_H (2Bh)* or *OUT_Z_L (2Ch), OUT_Z_H (2Dh)* registers provides the data stored in the FIFO. Each time data is read from the FIFO, the oldest X, Y and Z data are placed in the *OUT_X_L (28h), OUT_X_H (29h)*, *OUT_Y_L (2Ah), OUT_Y_H (2Bh)* and *OUT_Z_L (2Ch), OUT_Z_H (2Dh)* registers and both single read and read burst operations can be used.

The address to be read is automatically updated by the device and it rolls back to 0x28 when register 0x2D is reached. In order to read all FIFO levels in a multiple byte read, 192 bytes (6 output registers of 32 levels) have to be read.

6 Digital interfaces

The registers embedded inside the LIS3DH may be accessed through both the I^2C and SPI serial interfaces. The latter may be SW configured to operate either in 3-wire or 4-wire interface mode.

The serial interfaces are mapped onto the same pads. To select/exploit the I^2C interface, the CS line must be tied high (i.e. connected to Vdd_IO).

Table 14. Serial interface pin description

Pin name	Pin description
CS	SPI enable I^2C/SPI mode selection: 1: SPI idle mode / I^2C communication enabled 0: SPI communication mode / I^2C disabled
SCL SPC	I^2C serial clock (SCL) SPI serial port clock (SPC)
SDA SDI SDO	I^2C serial data (SDA) SPI serial data input (SDI) 3-wire interface serial data output (SDO)
SA0 SDO	I^2C less significant bit of the device address (SA0) SPI serial data output (SDO)

6.1 I^2C serial interface

The LIS3DH I^2C is a bus slave. The I^2C is employed to write data into registers whose content can also be read back.

The relevant I^2C terminology is given in the table below.

Table 15. I^2C terminology

Term	Description
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by the master

There are two signals associated with the I^2C bus: the serial clock line (SCL) and the Serial DAta line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. Both the lines must be connected to Vdd_IO through external pull-up resistor. When the bus is free both the lines are high.

The I^2C interface is compliant with fast mode (400 kHz) I^2C standards as well as with normal mode.

6.1.1 I²C operation

The transaction on the bus is started through a START (ST) signal. A START condition is defined as a HIGH-to-LOW transition on the data line while the SCL line is held HIGH. After this has been transmitted by the Master, the bus is considered busy. The next byte of data transmitted after the start condition contains the address of the slave in the first 7 bits and the eighth bit tells whether the Master is receiving data from the slave or transmitting data to the slave. When an address is sent, each device in the system compares the first seven bits after a start condition with its address. If they match, the device considers itself addressed by the Master.

The Slave ADdress (SAD) associated to the LIS3DH is 001100xb. The **SDO**/**SA0** pad can be used to modify the less significant bit of the device address. If the SA0 pad is connected to the voltage supply, LSb is ‘1’ (address 0011001b) else if SA0 pad is connected to ground, the LSb value is ‘0’ (address 0011000b). This solution permits to connect and address two different accelerometers to the same I²C lines.

Data transfer with acknowledge is mandatory. The transmitter must release the SDA line during the acknowledge pulse. The receiver must then pull the data line LOW so that it remains stable low during the HIGH period of the acknowledge clock pulse. A receiver which has been addressed is obliged to generate an acknowledge after each byte of data received.

The I²C embedded inside the LIS3DH behaves like a slave device and the following protocol must be adhered to. After the start condition (ST) a slave address is sent, once a slave acknowledge (SAK) has been returned, an 8-bit sub-address (SUB) is transmitted: the 7 LSb represent the actual register address while the MSB enables address auto increment. If the MSb of the SUB field is ‘1’, the SUB (register address) is automatically increased to allow multiple data read/writes.

The slave address is completed with a Read/Write bit. If the bit was ‘1’ (Read), a repeated START (SR) condition must be issued after the two sub-address bytes; if the bit is ‘0’ (Write) the Master transmit to the slave with direction unchanged. *Table 16* explains how the SAD+Read/Write bit pattern is composed, listing all the possible configurations.

Table 16. SAD+Read/Write patterns

Command	SAD[6:1]	SAD[0] = SA0	R/W	SAD+R/W
Read	001100	0	1	00110001 (31h)
Write	001100	0	0	00110000 (30h)
Read	001100	1	1	00110011 (33h)
Write	001100	1	0	00110010 (32h)

Table 17. Transfer when master is writing one byte to slave

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

Table 18. Transfer when master is writing multiple bytes to slave

Master	ST	SAD + W		SUB		DATA		DATA		SP
Slave			SAK		SAK		SAK		SAK	

Table 19. Transfer when master is receiving (reading) one byte of data from slave

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

Table 20. Transfer when master is receiving (reading) multiple bytes of data from slave

Master	ST	SAD+W		SUB		SR	SAD+R			MAK		MAK		NMAK	SP
Slave			SAK		SAK			SAK	DATA		DATA		DATA		

Data are transmitted in byte format (DATA). Each data transfer contains 8 bits. The number of bytes transferred per transfer is unlimited. Data is transferred with the Most Significant bit (MSb) first. If a receiver can't receive another complete byte of data until it has performed some other function, it can hold the clock line, SCL LOW to force the transmitter into a wait state. Data transfer only continues when the receiver is ready for another byte and releases the data line. If a slave receiver doesn't acknowledge the slave address (i.e. it is not able to receive because it is performing some real-time function) the data line must be left HIGH by the slave. The Master can then abort the transfer. A LOW-to-HIGH transition on the SDA line while the SCL line is HIGH is defined as a STOP condition. Each data transfer must be terminated by the generation of a STOP (SP) condition.

In order to read multiple bytes, it is necessary to assert the most significant bit of the sub-address field. In other words, SUB(7) must be equal to 1 while SUB(6-0) represents the address of first register to be read.

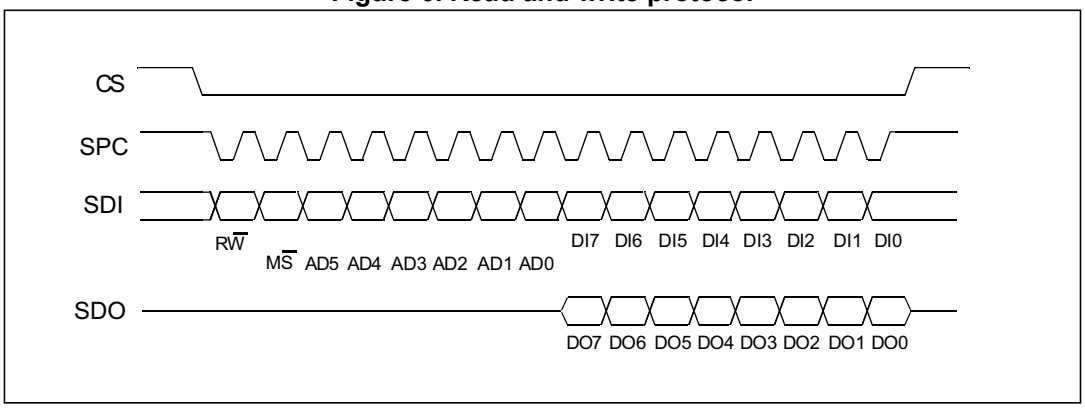
In the presented communication format MAK is Master acknowledge and NMAK is No Master Acknowledge.

6.2 SPI bus interface

The LIS3DH SPI is a bus slave. The SPI allows writing to and reading from the registers of the device.

The serial interface interacts with the application using 4 wires: **CS**, **SPC**, **SDI** and **SDO**.

Figure 6. Read and write protocol



CS is the serial port enable and it is controlled by the SPI master. It goes low at the start of the transmission and goes back high at the end. **SPC** is the serial port clock and it is controlled by the SPI master. It is stopped high when **CS** is high (no transmission). **SDI** and **SDO** are respectively the serial port data input and output. Those lines are driven at the falling edge of **SPC** and should be captured at the rising edge of **SPC**.

Both the read register and write register commands are completed in 16 clock pulses or in multiples of 8 in case of multiple read/write bytes. Bit duration is the time between two falling edges of **SPC**. The first bit (bit 0) starts at the first falling edge of **SPC** after the falling edge of **CS** while the last bit (bit 15, bit 23, ...) starts at the last falling edge of SPC just before the rising edge of **CS**.

bit 0: R$\overline{W}$ bit. When 0, the data DI(7:0) is written into the device. When 1, the data DO(7:0) from the device is read. In latter case, the chip drives **SDO** at the start of bit 8.

bit 1: M$\overline{S}$ bit. When 0, the address remains unchanged in multiple read/write commands. When 1, the address is auto incremented in multiple read/write commands.

bit 2-7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DI(7:0) (write mode). This is the data that is written into the device (MSb first).

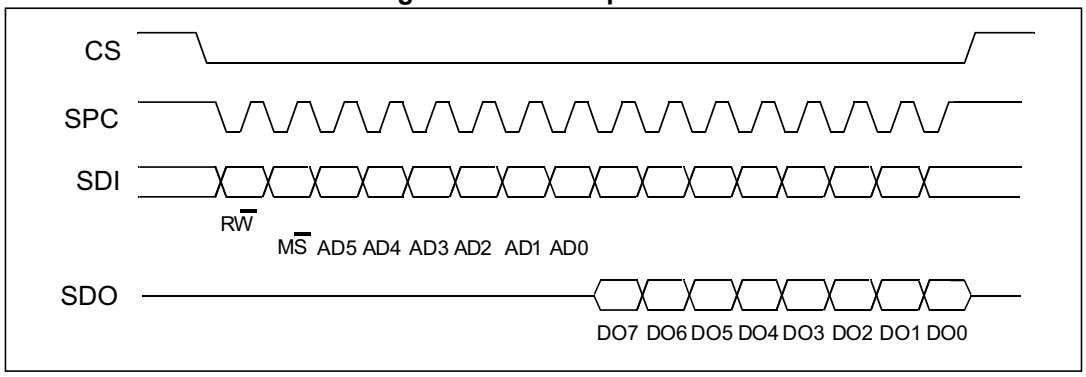
bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

In multiple read/write commands further blocks of 8 clock periods are added. When the M$\overline{S}$ bit is ‘0’, the address used to read/write data remains the same for every block. When the M$\overline{S}$ bit is ‘1’, the address used to read/write data is increased at every block.

The function and the behavior of **SDI** and **SDO** remain unchanged.

6.2.1 SPI read

Figure 7. SPI read protocol



The SPI read command is performed with 16 clock pulses. A multiple byte read command is performed by adding blocks of 8 clock pulses to the previous one.

bit 0: READ bit. The value is 1.

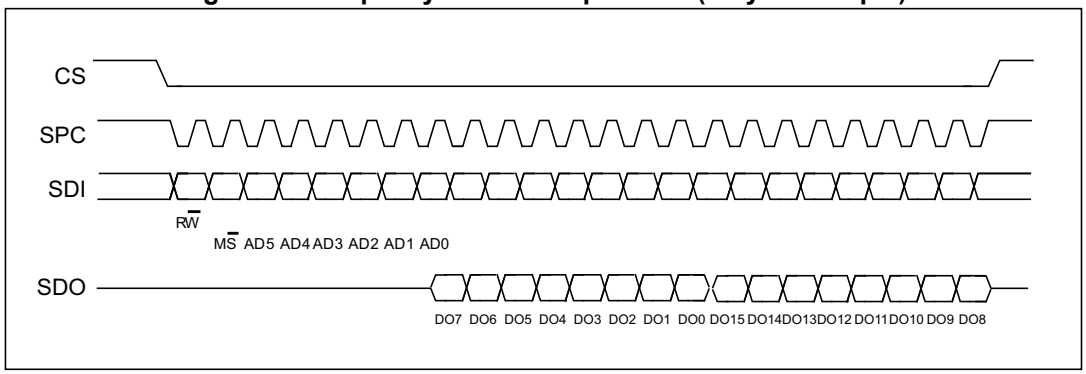
bit 1: MS bit. When 0, does not increment the address; when 1, increments the address in multiple reads.

bit 2-7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

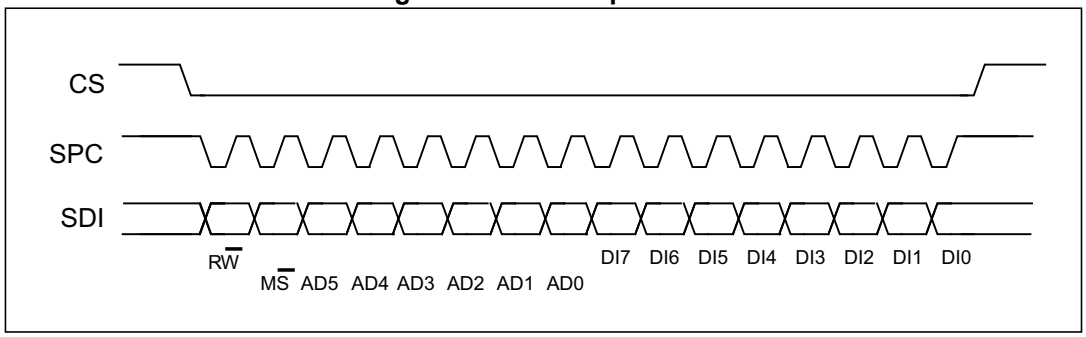
bit 16-... : data DO(...-8). Further data in multiple byte reads.

Figure 8. Multiple byte SPI read protocol (2-byte example)



6.2.2 SPI write

Figure 9. SPI write protocol



The SPI write command is performed with 16 clock pulses. A multiple byte write command is performed by adding blocks of 8 clock pulses to the previous one.

bit 0: WRITE bit. The value is 0.

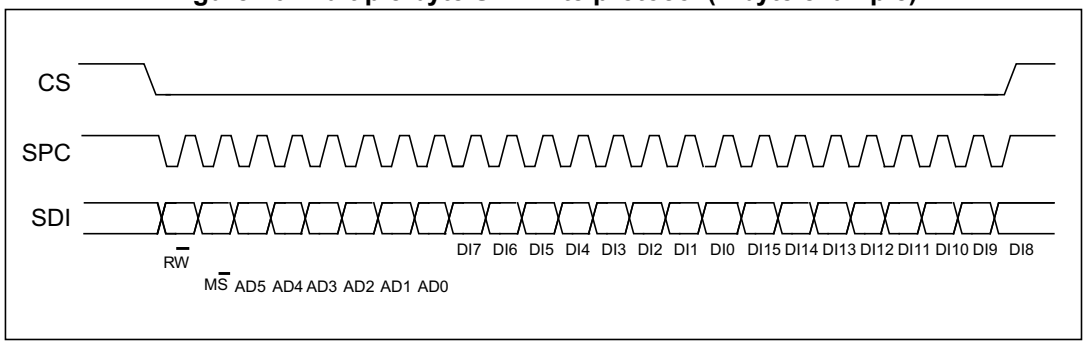
bit 1: MS bit. When 0, does not increment the address; when 1, increments the address in multiple writes.

bit 2 -7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DI(7:0) (write mode). This is the data that is written inside the device (MSb first).

bit 16-... : data DI(...-8). Further data in multiple byte writes.

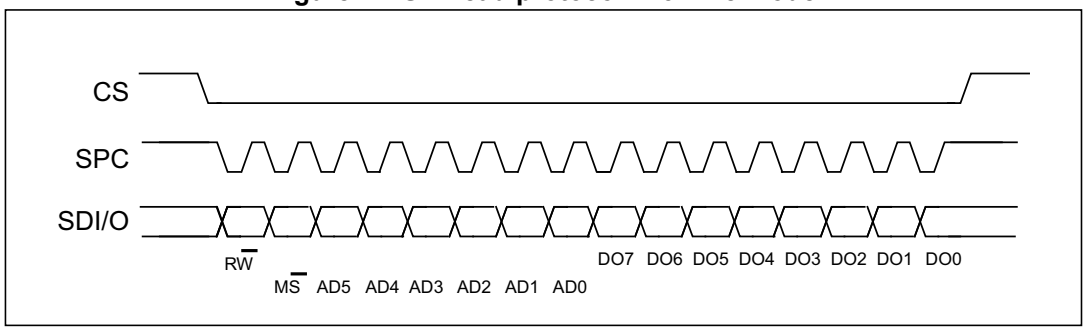
Figure 10. Multiple byte SPI write protocol (2-byte example)



6.2.3 SPI read in 3-wire mode

3-wire mode is entered by setting the bit SIM (SPI serial interface mode selection) to ‘1’ in *CTRL_REG4 (23h)*.

Figure 11. SPI read protocol in 3-wire mode



The SPI read command is performed with 16 clock pulses:

bit 0: READ bit. The value is 1.

bit 1: MS bit. When 0, does not increment the address; when 1, increments the address in multiple reads.

bit 2-7: address AD(5:0). This is the address field of the indexed register.

bit 8-15: data DO(7:0) (read mode). This is the data that is read from the device (MSb first).

The multiple read command is also available in 3-wire mode.

7 Register mapping

The table given below provides a list of the 8-bit registers embedded in the device and the corresponding addresses.

Table 21. Register address map

Name	Type	Register address		Default	Comment
		Hex	Binary		
Reserved (do not modify)		00 - 06			Reserved
STATUS_REG_AUX	r	07	000 0111	Output	
OUT_ADC1_L	r	08	000 1000	Output	
OUT_ADC1_H	r	09	000 1001	Output	
OUT_ADC2_L	r	0A	000 1010	Output	
OUT_ADC2_H	r	0B	000 1011	Output	
OUT_ADC3_L	r	0C	000 1100	Output	
OUT_ADC3_H	r	0D	000 1101	Output	
Reserved (do not modify)		0E			Reserved
WHO_AM_I	r	0F	000 1111	00110011	Dummy register
Reserved (do not modify)		10 - 1D			Reserved
CTRL_REG0	rw	1E	001 1110	00010000	
TEMP_CFG_REG	rw	1F	001 1111	00000000	
CTRL_REG1	rw	20	010 0000	00000111	
CTRL_REG2	rw	21	010 0001	00000000	
CTRL_REG3	rw	22	010 0010	00000000	
CTRL_REG4	rw	23	010 0011	00000000	
CTRL_REG5	rw	24	010 0100	00000000	
CTRL_REG6	rw	25	010 0101	00000000	
REFERENCE	rw	26	010 0110	00000000	
STATUS_REG	r	27	010 0111	Output	
OUT_X_L	r	28	010 1000	Output	
OUT_X_H	r	29	010 1001	Output	
OUT_Y_L	r	2A	010 1010	Output	
OUT_Y_H	r	2B	010 1011	Output	
OUT_Z_L	r	2C	010 1100	Output	
OUT_Z_H	r	2D	010 1101	Output	
FIFO_CTRL_REG	rw	2E	010 1110	00000000	
FIFO_SRC_REG	r	2F	010 1111	Output	

Table 21. Register address map

Name	Type	Register address		Default	Comment
		Hex	Binary		
INT1_CFG	rw	30	011 0000	00000000	
INT1_SRC	r	31	011 0001	Output	
INT1_THS	rw	32	011 0010	00000000	
INT1_DURATION	rw	33	011 0011	00000000	
INT2_CFG	rw	34	011 0100	00000000	
INT2_SRC	r	35	011 0101	Output	
INT2_THS	rw	36	011 0110	00000000	
INT2_DURATION	rw	37	011 0111	00000000	
CLICK_CFG	rw	38	011 1000	00000000	
CLICK_SRC	r	39	011 1001	Output	
CLICK_THS	rw	3A	011 1010	00000000	
TIME_LIMIT	rw	3B	011 1011	00000000	
TIME_LATENCY	rw	3C	011 1100	00000000	
TIME_WINDOW	rw	3D	011 1101	00000000	
ACT_THS	rw	3E	011 1110	00000000	
ACT_DUR	rw	3F	011 1111	00000000	

Registers marked as *Reserved* or not listed in the table above must not be changed. Writing to those registers may cause permanent damage to the device.

The content of the registers that are loaded at boot should not be changed. They contain the factory calibration values. Their content is automatically restored when the device is powered up.

The boot procedure is complete about 5 milliseconds after device power-up.

8 Registers description

8.1 STATUS_REG_AUX (07h)

Table 22. STATUS_REG_AUX register

321OR	3OR	2OR	1OR	321DA	3DA	2DA	1DA

Table 23. STATUS_REG_AUX description

321OR	1, 2 and 3-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: a new set of data has overwritten the previous set)
3OR	3-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the 3-axis has overwritten the previous data)
2OR	2-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the 2-axis has overwritten the previous data)
1OR	1-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the 1-axis has overwritten the previous data)
321DA	1, 2 and 3-axis new data available. Default value: 0 (0: a new set of data is not yet available; 1: a new set of data is available)
3DA	3-axis new data available. Default value: 0 (0: new data for the 3-axis is not yet available; 1: new data for the 3-axis is available)
2DA	2 -axis new data available. Default value: 0 (0: new data for the 2-axis is not yet available; 1: new data for the 2-axis is available)
1DA	1-axis new data available. Default value: 0 (0: new data for the 1-axis is not yet available; 1: new data for the 1-axis is available)

8.2 OUT_ADC1_L (08h), OUT_ADC1_H (09h)

Auxiliary 10-bit ADC channel 1 conversion. For auxiliary ADC setting refer to *Section 3.7: Auxiliary ADC and temperature sensor*.

8.3 OUT_ADC2_L (0Ah), OUT_ADC2_H (0Bh)

Auxiliary 10-bit ADC channel 2 conversion. For auxiliary ADC setting refer to *Section 3.7: Auxiliary ADC and temperature sensor*.

8.4 OUT_ADC3_L (0Ch), OUT_ADC3_H (0Dh)

Auxiliary 10-bit ADC channel 3 conversion or temperature sensor data output. Refer to *Section 3.7: Auxiliary ADC and temperature sensor*.

8.5 WHO_AM_I (0Fh)

Table 24. WHO_AM_I register

0	0	1	1	0	0	1	1

Device identification register.

8.6 CTRL_REG0 (1Eh)

Table 25. CTRL_REG0 register

SDO_PU_DISC	0(1)	0(1)	1(2)	0(1)	0(1)	0(1)	0(1)

1. This bit must be set to 0 for correct operation of the device.
2. This bit must be set to 1 for correct operation of the device.

Table 26. CTRL_REG0 description

SDO_PU_DISC	Disconnect SDO/SA0 pull-up. Default value: 00010000 (0: pull-up connected to SDO/SA0 pin; 1: pull-up disconnected to SDO/SA0 pin)

Note: Leave bits 0 through 6 at the default value in order to ensure correct operation of the device.

8.7 TEMP_CFG_REG (1Fh)

Table 27. TEMP_CFG_REG register

ADC_EN	TEMP_EN	0	0	0	0	0	0

Table 28. TEMP_CFG_REG description

TEMP_EN	Temperature sensor (T) enable. Default value: 0 (0: T disabled; 1: T enabled)
ADC_EN	ADC enable. Default value: 0 (0: ADC disabled; 1: ADC enabled)

8.8 CTRL_REG1 (20h)

Table 29. CTRL_REG1 register

ODR3	ODR2	ODR1	ODR0	LPen	Zen	Yen	Xen

Table 30. CTRL_REG1 description

ODR[3:0]	Data rate selection. Default value: 0000 (0000: power-down mode; others: Refer to *Table 31: Data rate configuration*)
LPen	Low-power mode enable. Default value: 0 (0: high-resolution mode / normal mode, 1: low-power mode) (Refer to section *Section 3.2.1: High-resolution, normal mode, low-power mode*)
Zen	Z-axis enable. Default value: 1 (0: Z-axis disabled; 1: Z-axis enabled)
Yen	Y-axis enable. Default value: 1 (0: Y-axis disabled; 1: Y-axis enabled)
Xen	X-axis enable. Default value: 1 (0: X-axis disabled; 1: X-axis enabled)

ODR[3:0] is used to set the power mode and ODR selection. The following table indicates the frequency of each combination of ODR[3:0].

Table 31. Data rate configuration

ODR3	ODR2	ODR1	ODR0	Power mode selection
0	0	0	0	Power-down mode
0	0	0	1	HR / Normal / Low-power mode (1 Hz)
0	0	1	0	HR / Normal / Low-power mode (10 Hz)
0	0	1	1	HR / Normal / Low-power mode (25 Hz)
0	1	0	0	HR / Normal / Low-power mode (50 Hz)
0	1	0	1	HR / Normal / Low-power mode (100 Hz)
0	1	1	0	HR / Normal / Low-power mode (200 Hz)
0	1	1	1	HR / Normal / Low-power mode (400 Hz)
1	0	0	0	Low power mode (1.60 kHz)
1	0	0	1	HR / normal (1.344 kHz); Low-power mode (5.376 kHz)

8.9 CTRL_REG2 (21h)

Table 32. CTRL_REG2 register

HPM1	HPM0	HPCF2	HPCF1	FDS	HPCLICK	HP_IA2	HP_IA1

Table 33. CTRL_REG2 description

HPM[1:0]	High-pass filter mode selection. Default value: 00 Refer to *Table 34: High-pass filter mode configuration*
HPCF[2:1]	High-pass filter cutoff frequency selection
FDS	Filtered data selection. Default value: 0 (0: internal filter bypassed; 1: data from internal filter sent to output register and FIFO)
HPCLICK	High-pass filter enabled for CLICK function. (0: filter bypassed; 1: filter enabled)
HP_IA2	High-pass filter enabled for AOI function on interrupt 2, (0: filter bypassed; 1: filter enabled)
HP_IA1	High-pass filter enabled for AOI function on interrupt 1, (0: filter bypassed; 1: filter enabled)

Table 34. High-pass filter mode configuration

HPM1	HPM0	High-pass filter mode
0	0	Normal mode (reset by reading *REFERENCE (26h)*)
0	1	Reference signal for filtering
1	0	Normal mode
1	1	Autoreset on interrupt event

8.10 CTRL_REG3 (22h)

Table 35. CTRL_REG3 register

I1_CLICK	I1_IA1	I1_IA2	I1_ZYXDA	I1_321DA	I1_WTM	I1_OVERRUN	--

Table 36. CTRL_REG3 description

I1_CLICK	Click interrupt on INT1. Default value: 0 (0: disable; 1: enable)
I1_IA1	IA1 interrupt on INT1. Default value: 0 (0: disable; 1: enable)
I1_IA2	IA2 interrupt on INT1. Default value: 0 (0: disable; 1: enable)
I1_ZYXDA	ZYXDA interrupt on INT1. Default value: 0 (0: disable; 1: enable)
I1_321DA	321DA interrupt on INT1. Default value: 0 (0: disable; 1: enable)
I1_WTM	FIFO watermark interrupt on INT1. Default value: 0 (0: disable; 1: enable)
I1_OVERRUN	FIFO overrun interrupt on INT1. Default value: 0 (0: disable; 1: enable)

8.11 CTRL_REG4 (23h)

Table 37. CTRL_REG4 register

BDU	BLE(1)	FS1	FS0	HR	ST1	ST0	SIM

1. The BLE function can be activated only in high-resolution mode.

Table 38. CTRL_REG4 description

BDU	Block data update. Default value: 0 (0: continuous update; 1: output registers not updated until MSB and LSB reading)
BLE	Big/little endian data selection. Default value 0. (0: Data LSB @ lower address; 1: Data MSB @ lower address)
FS[1:0]	Full-scale selection. default value: 00 (00: ±2 *g*; 01: ±4 *g*; 10: ±8 *g*; 11: ±16 *g*)
HR	High-resolution output mode: Default value: 0 (0: high-resolution disabled; 1: high-resolution enabled)
ST[1:0]	Self-test enable. Default value: 00 (00: self-test disabled; other: See *Table 39*)
SIM	SPI serial interface mode selection. Default value: 0 (0: 4-wire interface; 1: 3-wire interface)

Table 39. Self-test mode configuration

ST1	ST0	Self test mode
0	0	Normal mode
0	1	Self-test 0
1	0	Self-test 1
1	1	--

8.12 CTRL_REG5 (24h)

Table 40. CTRL_REG5 register

BOOT	FIFO_EN	--	--	LIR_INT1	D4D_INT1	LIR_INT2	D4D_INT2

Table 41. CTRL_REG5 description

BOOT	Reboot memory content. Default value: 0 (0: normal mode; 1: reboot memory content)
FIFO_EN	FIFO enable. Default value: 0 (0: FIFO disable; 1: FIFO enable)
LIR_INT1	Latch interrupt request on INT1_SRC register, with *INT1_SRC (31h)* register cleared by reading *INT1_SRC (31h)* itself. Default value: 0. (0: interrupt request not latched; 1: interrupt request latched)
D4D_INT1	4D enable: 4D detection is enabled on INT1 when 6D bit on INT1_CFG is set to 1.
LIR_INT2	Latch interrupt request on *INT2_SRC (35h)* register, with *INT2_SRC (35h)* register cleared by reading *INT2_SRC (35h)* itself. Default value: 0 (0: interrupt request not latched; 1: interrupt request latched)
D4D_INT2	4D enable: 4D detection is enabled on INT2 pin when 6D bit on *INT2_CFG (34h)* is set to 1.

8.13 CTRL_REG6 (25h)

Table 42. CTRL_REG6 register

I2_CLICK	I2_IA1	I2_IA2	I2_BOOT	I2_ACT	- -	INT_POLARITY	-

Table 43. CTRL_REG6 description

I2_CLICK	Click interrupt on INT2 pin. Default value: 0 (0: disabled; 1: enabled)
I2_IA1	Enable interrupt 1 function on INT2 pin. Default value: 0 (0: function disabled; 1: function enabled)
I2_IA2	Enable interrupt 2 function on INT2 pin. Default value: 0 (0: function disabled; 1: function enabled)
I2_BOOT	Enable boot on INT2 pin. Default value: 0 (0: disabled; 1:enabled)
I2_ACT	Enable activity interrupt on INT2 pin. Default value: 0 (0: disabled; 1:enabled)
INT_POLARITY	INT1 and INT2 pin polarity. Default value: 0 (0: active-high; 1: active-low)

8.14 REFERENCE (26h)

Table 44. REFERENCE register

Ref7	Ref6	Ref5	Ref4	Ref3	Ref2	Ref1	Ref0

Table 45. REFERENCE register description

Ref[7:0]	Reference value for Interrupt generation. Default value: 0000 0000

8.15 STATUS_REG (27h)

Table 46. STATUS register

ZYXOR	ZOR	YOR	XOR	ZYXDA	ZDA	YDA	XDA

Table 47. STATUS register description

ZYXOR	X, Y and Z-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: a new set of data has overwritten the previous set)
ZOR	Z-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: a new data for the Z-axis has overwritten the previous data)
YOR	Y-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the Y-axis has overwritten the previous data)
XOR	X-axis data overrun. Default value: 0 (0: no overrun has occurred; 1: new data for the X-axis has overwritten the previous data)
ZYXDA	X, Y and Z-axis new data available. Default value: 0 (0: a new set of data is not yet available; 1: a new set of data is available)
ZDA	Z-axis new data available. Default value: 0 (0: new data for the Z-axis is not yet available; 1: new data for the Z-axis is available)
YDA	Y-axis new data available. Default value: 0 (0: new data for the Y-axis is not yet available; 1: new data for the Y-axis is available)

8.16 OUT_X_L (28h), OUT_X_H (29h)

X-axis acceleration data. The value is expressed as two's complement left-justified.
Please refer to *Section 3.2.1: High-resolution, normal mode, low-power mode*.

8.17 OUT_Y_L (2Ah), OUT_Y_H (2Bh)

Y-axis acceleration data. The value is expressed as two's complement left-justified.
Please refer to *Section 3.2.1: High-resolution, normal mode, low-power mode*.

8.18 OUT_Z_L (2Ch), OUT_Z_H (2Dh)

Z-axis acceleration data. The value is expressed as two's complement left-justified.
Please refer to *Section 3.2.1: High-resolution, normal mode, low-power mode*.

8.19 FIFO_CTRL_REG (2Eh)

Table 48. REFERENCE register

FM1	FM0	TR	FTH4	FTH3	FTH2	FTH1	FTH0

Table 49. REFERENCE register description

FM[1:0]	FIFO mode selection. Default value: 00 (see *Table 50*)
TR	Trigger selection. Default value: 0 (0: trigger event allows triggering signal on INT1 1: trigger event allows triggering signal on INT2)
FTH[4:0]	Default value: 00000

Table 50. FIFO mode configuration

FM1	FM0	Self test mode
0	0	Bypass mode
0	1	FIFO mode
1	0	Stream mode
1	1	Stream-to-FIFO

8.20 FIFO_SRC_REG (2Fh)

Table 51. FIFO_SRC_REG register

WTM	OVRN_FIFO	EMPTY	FSS4	FSS3	FSS2	FSS1	FSS0

Table 52. FIFO_SRC_REG description

WTM	WTM bit is set high when FIFO content exceeds watermark level
OVRN_FIFO	OVRN bit is set high when FIFO buffer is full; this means that the FIFO buffer contains 32 unread samples. At the following ODR a new sample set replaces the oldest FIFO value. The OVRN bit is set to 0 when the first sample set has been read
EMPTY	EMPTY flag is set high when all FIFO samples have been read and FIFO is empty
FSS [4:0]	FSS [4:0] field always contains the current number of unread samples stored in the FIFO buffer. When FIFO is enabled, this value increases at ODR frequency until the buffer is full, whereas, it decreases every time one sample set is retrieved from FIFO.

8.21 INT1_CFG (30h)

Table 53. INT1_CFG register

AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE

Table 54. INT1_CFG description

AOI	And/Or combination of Interrupt events. Default value: 0 Refer to *Table 55: Interrupt mode*
6D	6 direction detection function enabled. Default value: 0Refer to *Table 55: Interrupt mode*
ZHIE	Enable interrupt generation on Z high event or on Direction recognition. Default value: 0 (0: disable interrupt request;1: enable interrupt request)
ZLIE	Enable interrupt generation on Z low event or on Direction recognition. Default value: 0 (0: disable interrupt request;1: enable interrupt request)
YHIE	Enable interrupt generation on Y high event or on Direction recognition. Default value: 0 (0: disable interrupt request; 1: enable interrupt request.)
YLIE	Enable interrupt generation on Y low event or on Direction recognition. Default value: 0 (0: disable interrupt request; 1: enable interrupt request.)
XHIE	Enable interrupt generation on X high event or on Direction recognition. Default value: 0 (0: disable interrupt request; 1: enable interrupt request.)
XLIE	Enable interrupt generation on X low event or on Direction recognition. Default value: 0 (0: disable interrupt request; 1: enable interrupt request.)

Content of this register is loaded at boot.

Write operation at this address is possible only after system boot.

Table 55. Interrupt mode

AOI	6D	Interrupt mode
0	0	OR combination of interrupt events
0	1	6-direction movement recognition
1	0	AND combination of interrupt events
1	1	6-direction position recognition

Difference between AOI-6D = ‘01’ and AOI-6D = ‘11’.

AOI-6D = ‘01’ is movement recognition. An interrupt is generated when the orientation moves from an unknown zone to known zone. The interrupt signal remains for a duration ODR.

AOI-6D = ‘11’ is direction recognition. An interrupt is generated when the orientation is inside a known zone. The interrupt signal remains until the orientation is inside the zone.

8.22 INT1_SRC (31h)

Table 56. INT1_SRC register

0	IA	ZH	ZL	YH	YL	XH	XL

Table 57. INT1_SRC description

IA	Interrupt active. Default value: 0 (0: no interrupt has been generated; 1: one or more interrupts have been generated)
ZH	Z high. Default value: 0 (0: no interrupt, 1: Z high event has occurred)
ZL	Z low. Default value: 0 (0: no interrupt; 1: Z low event has occurred)
YH	Y high. Default value: 0 (0: no interrupt, 1: Y high event has occurred)
YL	Y low. Default value: 0 (0: no interrupt, 1: Y low event has occurred)
XH	X high. Default value: 0 (0: no interrupt, 1: X high event has occurred)
XL	X low. Default value: 0 (0: no interrupt, 1: X low event has occurred)

Interrupt 1 source register. Read-only register.

Reading at this address clears the *INT1_SRC (31h)* IA bit (and the interrupt signal on the INT 1 pin) and allows the refresh of data in *INT1_SRC (31h)* if the latched option was chosen.

8.23 INT1_THS (32h)

Table 58. INT1_THS register

0	THS6	THS5	THS4	THS3	THS2	THS1	THS0

Table 59. INT1_THS description

THS[6:0]	Interrupt 1 threshold. Default value: 000 0000 1 LSb = 16 m*g* @ FS = ±2 *g* 1 LSb = 32 m*g* @ FS = ±4 *g* 1 LSb = 62 m*g* @ FS = ±8 *g* 1 LSb = 186 m*g* @ FS = ±16 *g*

8.24 INT1_DURATION (33h)

Table 60. INT1_DURATION register

0	D6	D5	D4	D3	D2	D1	D0

Table 61. INT1_DURATION description

D[6:0]	Duration value. Default value: 000 0000 1 LSb = 1/ODR

The **D[6:0]** bits set the minimum duration of the Interrupt 2 event to be recognized. Duration steps and maximum values depend on the ODR chosen.

Duration time is measured in N/ODR, where N is the content of the duration register.

8.25 INT2_CFG (34h)

Table 62. INT2_CFG register

AOI	6D	ZHIE	ZLIE	YHIE	YLIE	XHIE	XLIE

Table 63. INT2_CFG description

AOI	AND/OR combination of interrupt events. Default value: 0 (see *Table 64*)
6D	6-direction detection function enabled. Default value: 0. Refer to *Table 64.*
ZHIE	Enable interrupt generation on Z high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
ZLIE	Enable interrupt generation on Z low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)
YHIE	Enable interrupt generation on Y high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
YLIE	Enable interrupt generation on Y low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)
XHIE	Enable interrupt generation on X high event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
XLIE	Enable interrupt generation on X low event. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value lower than preset threshold)

The content of this register is loaded at boot.

A write operation to this address is possible only after system boot.

Table 64. Interrupt mode

AOI	6D	Interrupt mode
0	0	OR combination of interrupt events
0	1	6-direction movement recognition
1	0	AND combination of interrupt events
1	1	6-direction position recognition

The difference between AOI-6D = ‘01’ and AOI-6D = ‘11’.

AOI-6D = ‘01’ is movement recognition. An interrupt is generated when the orientation moves from an unknown zone to a known zone. The interrupt signal remains for a duration ODR.

AOI-6D = ‘11’ is direction recognition. An interrupt is generated when the orientation is inside a known zone. The interrupt signal remains while the orientation is inside the zone.

8.26 INT2_SRC (35h)

Table 65. INT2_SRC register

0	IA	ZH	ZL	YH	YL	XH	XL

Table 66. INT2_SRC description

IA	Interrupt active. Default value: 0 (0: no interrupt has been generated; 1: one or more interrupts have been generated)
ZH	Z high. Default value: 0 (0: no interrupt, 1: Z high event has occurred)
ZL	Z low. Default value: 0 (0: no interrupt; 1: Z low event has occurred)
YH	Y high. Default value: 0 (0: no interrupt, 1: Y high event has occurred)
YL	Y low. Default value: 0 (0: no interrupt, 1: Y low event has occurred)
XH	X high. Default value: 0 (0: no interrupt, 1: X high event has occurred)
XL	X low. Default value: 0 (0: no interrupt, 1: X low event has occurred)

Interrupt 2 source register. Read-only register.

Reading at this address clears the *INT2_SRC (35h)* IA bit (and the interrupt signal on the INT2 pin) and allows the refresh of data in the *INT2_SRC (35h)* register if the latched option was chosen.

8.27 INT2_THS (36h)

Table 67. INT2_THS register

0	THS6	THS5	THS4	THS3	THS2	THS1	THS0

Table 68. INT2_THS description

THS[6:0]	Interrupt 2 threshold. Default value: 000 0000 1 LSb = 16 m*g* @ FS = ±2 *g* 1 LSb = 32 m*g* @ FS = ±4 *g* 1 LSb = 62 m*g* @ FS = ±8 *g* 1 LSb = 186 m*g* @ FS = ±16 *g*

8.28 INT2_DURATION (37h)

Table 69. INT2_DURATION register

0	D6	D5	D4	D3	D2	D1	D0

Table 70. INT2_DURATION description

D[6:0]	Duration value. Default value: 000 0000 1 LSb = 1/ODR(1)

1. Duration time is measured in N/ODR, where N is the content of the duration register.

The **D[6:0]** bits set the minimum duration of the Interrupt 2 event to be recognized. Duration time steps and maximum values depend on the ODR chosen.

8.29 CLICK_CFG (38h)

Table 71. CLICK_CFG register

--	--	zd	zs	yd	ys	xd	xs

Table 72. CLICK_CFG description

ZD	Enable interrupt double click on Z-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
ZS	Enable interrupt single click on Z-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
YD	Enable interrupt double click on Y-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
YS	Enable interrupt single click on Y-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
XD	Enable interrupt double click on X-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)
XS	Enable interrupt single click on X-axis. Default value: 0 (0: disable interrupt request; 1: enable interrupt request on measured accel. value higher than preset threshold)

8.30 CLICK_SRC (39h)

Table 73. CLICK_SRC register

	IA	DCLICK	SCLICK	Sign	Z	Y	X

Table 74. CLICK_SRC description

IA	Interrupt active. Default value: 0 (0: no interrupt has been generated; 1: one or more interrupts have been generated)
DCLICK	Double-click enable. Default value: 0 (0:double-click detection disabled, 1: double-click detection enabled)
SCLICK	Single-click enable. Default value: 0 (0: Single-click detection disabled, 1: single-click detection enabled)
Sign	Click sign. (0: positive detection, 1: negative detection)
Z	Z click detection. Default value: 0 (0: no interrupt, 1: Z high event has occurred)
Y	Y click detection. Default value: 0 (0: no interrupt, 1: Y high event has occurred)
X	X click detection. Default value: 0 (0: no interrupt, 1: X high event has occurred)

8.31 CLICK_THS (3Ah)

Table 75. CLICK_THS register

LIR_Click	Ths6	Ths5	Ths4	Ths3	Ths2	Ths1	Ths0

Table 76. CLICK_SRC description

LIR_Click	If the LIR_Click bit is not set, the interrupt is kept high for the duration of the latency window. If the LIR_Click bit is set, the interrupt is kept high until the *CLICK_SRC (39h)* register is read.
Ths[6:0]	Click threshold. Default value: 000 0000

8.32 TIME_LIMIT (3Bh)

Table 77. TIME_LIMIT register

-	TLI6	TLI5	TLI4	TLI3	TLI2	TLI1	TLI0

Table 78. TIME_LIMIT description

TLI[6:0]	Click time limit. Default value: 000 0000

8.33 TIME_LATENCY (3Ch)

Table 79. TIME_LATENCY register

TLA7	TLA6	TLA5	TLA4	TLA3	TLA2	TLA1	TLA0

Table 80. TIME_LATENCY description

TLA[7:0]	Click time latency. Default value: 0000 0000

8.34 TIME WINDOW (3Dh)

Table 81. TIME_WINDOW register

TW7	TW6	TW5	TW4	TW3	TW2	TW1	TW0

Table 82. TIME_WINDOW description

TW[7:0]	Click time window

8.35 ACT_THS (3Eh)

Table 83. ACT_THS register

--	Acth6	Acth5	Acth4	Acth3	Acth2	Acth1	Acth0

Table 84. ACT_THS description

Acth[6:0]	Sleep-to-wake, return-to-sleep activation threshold in low-power mode 1 LSb = 16 mg @ FS = ±2 *g* 1 LSb = 32 mg @ FS = ±4 *g* 1 LSb = 62 mg @ FS = ±8 *g* 1 LSb = 186 mg @ FS = ±16 *g*

8.36 ACT_DUR (3Fh)

Table 85. ACT_DUR register

ActD7	ActD6	ActD5	ActD4	ActD3	ActD2	ActD1	ActD0

Table 86. ACT_DUR description

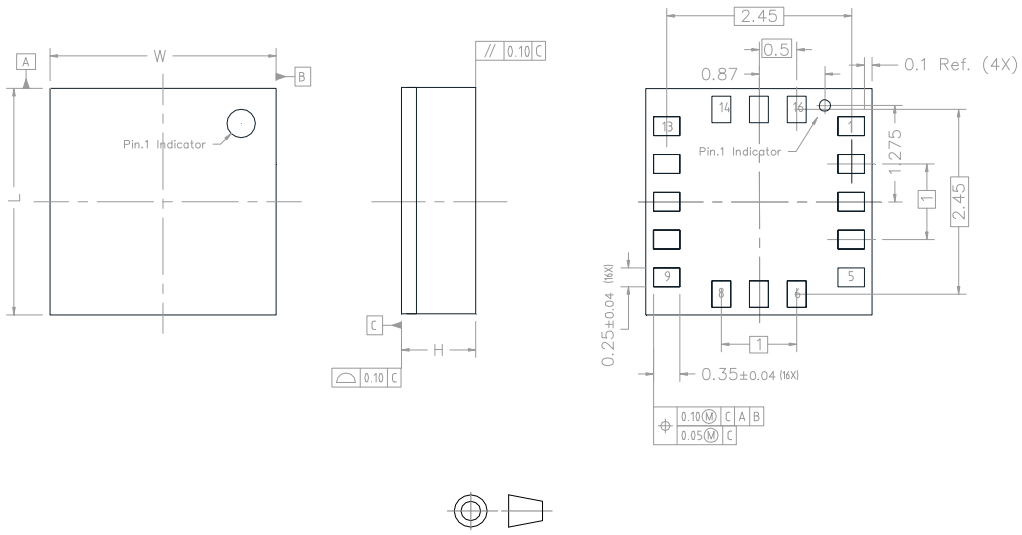
ActD[7:0]	Sleep-to-wake, return-to-sleep duration 1 LSb = (8*1[LSb]+1)/ODR

9 Package information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: www.st.com.
ECOPACK is an ST trademark.

9.1 LGA-16 package information

Figure 12. LGA-16 package outline and mechanical dimensions



Dimensions are in millimeter unless otherwise specified
General Tolerance is +/-0.1mm unless otherwise specified

OUTER DIMENSIONS

ITEM	DIMENSION [mm]	TOLERANCE [mm]
Length [L]	3	±0.15
Width [W]	3	±0.15
Height [H]	1 max	/

7983231_13

9.2 LGA-16 packing information

Figure 13. Carrier tape information for LGA-16 package

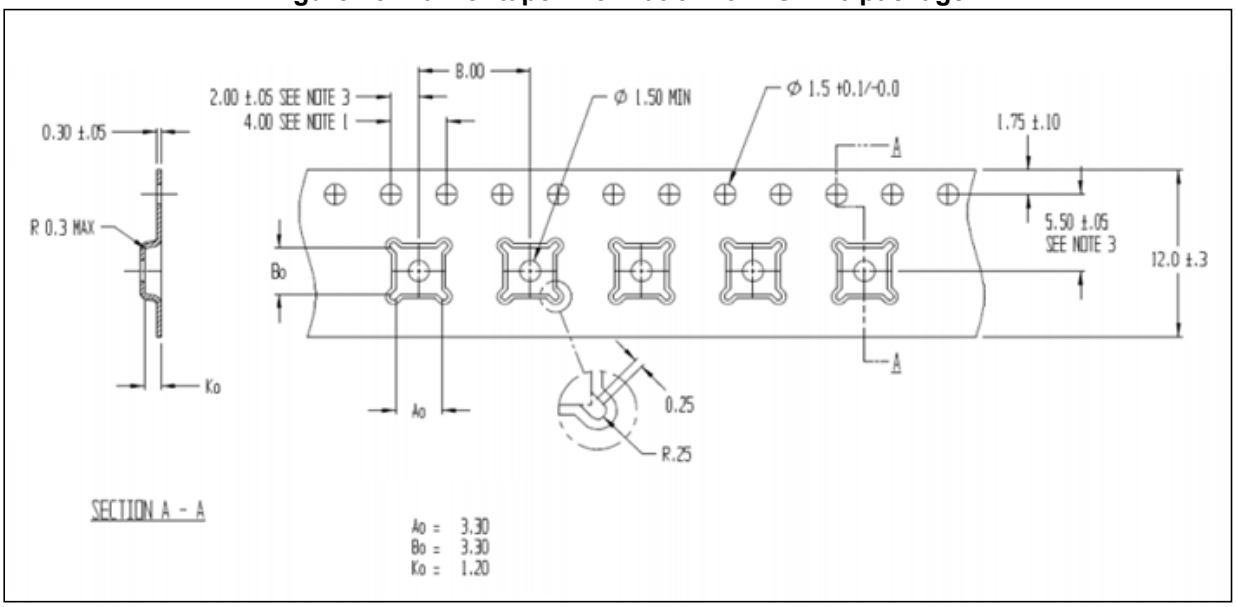


Figure 14. LGA-16 package orientation in carrier tape

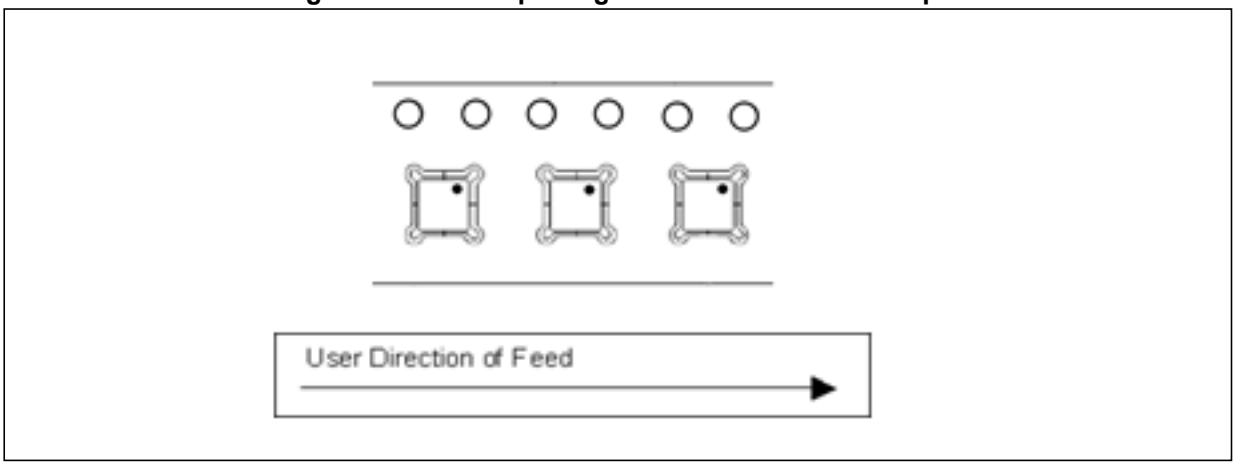


Figure 15. Reel information for carrier tape of LGA-16 package

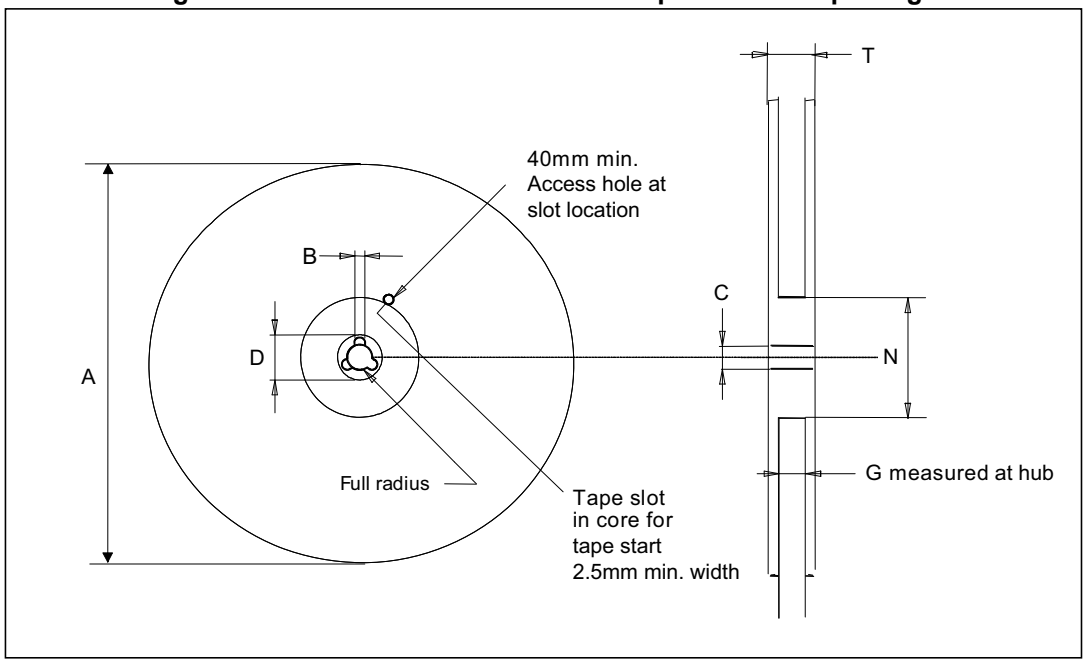


Table 87. Reel dimensions for carrier tape of LGA-16 package

Reel dimensions (mm)	
A (max)	330
B (min)	1.5
C	13 ±0.25
D (min)	20.2
N (min)	60
G	12.4 +2/-0
T (max)	18.4

10 Revision history

Table 88. Document revision history

Date	Revision	Changes
21-May-2010	1	Initial release
12-Dec-2016	2	Updated *Table 1: Device summary* Updated *Features* and *Figure 1: Block diagram* Updated *Table 2: Pin description* and *Table 14: Serial interface pin description* Added *Table 3: Internal pull-up values (typ.) for SDO/SA0 pin* Updated *Table 9: Absolute maximum ratings* Updated *Section 3.7: Auxiliary ADC and temperature sensor* Updated *Section 4: Application hints* Updated *Section 5: Digital main blocks* Updated *Section 7: Register mapping* and *Section 8: Registers description* Updated *Section 9.1: LGA-16 package information* Added *Section 9.2: LGA-16 packing information* Minor textual updates