

## Introduction

The PmodAD2 is an analog to digital converter peripheral module with up to 4 channels and 12-bit resolution powered by the Analog Devices AD7991 circuit. Digilent provides an Arduino driver library for this device.

This document provides an overview of the operation of this driver library and describes the functions that define its programming interface.

## Overview

The PmodDA2 device has one Analog Devices AD7991 chip that offers up to 4 channels of 12-bit analog to digital converter.

For more information about the hardware interface of the PmodDA2, refer to the PmodDA2. Reference Manual available for download from the Digilent web site ([www.digilentinc.com](http://www.digilentinc.com)).

The converter is a serial device that is accessed (read) via the I2C interface.

By sending a configuration byte, a request is made for simultaneous conversion of 1, 2, 3 or 4 channels. The device allows using the channel 3 as reference input, in this mode offering just three channels to be converted.

The value read from converter is represented on 12 bits (from 0 to 0x0FFF) and is considered the “*IntegerValue*”. The library also defines the “*PhysicalValue*”, which is a floating point value that corresponds to the “*IntegerValue*” and to a reference value that defines the reference voltage (or any other measured item) (associated to the maximum integer value).

$$\text{PhysicalValue} = \text{IntegerValue} * (\text{ReferenceValue} / (2^{12} - 1))$$

By default, the reference value corresponds to reference voltage of 3.3V in this case.

## Library Operation

### Library Interface

The header file ADCI2C.h defines the interfaces for the ADCI2C driver. The library is accessed via the methods and constants defined for the ADCI2C object class. In order to use this library, the user has to instantiate one library object.

## I2C Initialization Function

In order to communicate with the PmodDA2, the I2C interface has to be initialized. Before calling any other library functions, the `begin()` function must be called. This function initializes and configures the I2C port used by the library.

### AD Converter Functions Overview

In order to read values from the AD converter, the library provides one basic function that performs fast acquisition without validating the user parameters: `RawReadChannels`. Additionally, the library provides two types of functions: `GetIntegerValue()` and `GetPhysicalValue()`, for 1, 2, 3 and 4 channels. Read the Overview chapter for a description of these values.

Because the AD7991 device allows selecting which channels are being converted, this versatility is implemented in the library in the following way: for each type (integer and physical), there are four functions, each providing a different number of output parameters to receive the read values:

- 1Ch suffix (`GetIntegerValue1Ch` and `GetPhysicalValue1Ch`)
  - o Allows converting one channel, according to configuration byte.
  - o The user must specify one output parameter.
- 2Ch suffix (`GetIntegerValues2Ch` and `GetPhysicalValues2Ch`)
  - o Allows converting two channels, according to configuration byte.
  - o The user must specify two output parameters.
- 3Ch suffix (`GetIntegerValues3Ch` and `GetPhysicalValues3Ch`)
  - o Allows converting three channels, according to configuration byte.
  - o The user must specify three output parameters.
- 4Ch suffix (`GetIntegerValues4Ch` and `GetPhysicalValues4Ch`)
  - o Allows converting all four channels, according to configuration byte.
  - o The user must specify four output parameters.

Selecting the function to be used is done in the following way:

- If the user is interested in performance (speed of the execution), he can use `RawReadChannels`. He must specify the number of channels to be read, the configuration options and an array where the values will be stored.
- The user may obtain the integer values using the `GetIntegerValues` function corresponding to the number of channels he intends to use and providing the specific number of output parameters (one for each channel).
- The user may obtain the physical values using the `GetPhysicalValues` function corresponding to the number of channels he intends to use and providing the specific number of output parameters (one for each channel).

When indicating the configuration byte, the user can use logical expressions such as `ADCI2C_MSK_CH0 | ADCI2C_MSK_CH2`. It needs to be mentioned that the order in which channels are written in this expression has no effect. What matters is the channel number.

## ADCI2C Library Functions

### **void begin()**

This function initializes the I2C interface #1 that is used to communicate with the PmodAD2.

### **void RawReadChannels(uint8\_t bConfig, uint8\_t bCntChannels, uint16\_t \*rgwValues)**

#### *Parameters:*

- uint8\_t bConfig - the configuration settings, specifying which channels are selected for conversion and other options
  - it can be one or more (OR-ed) options from the list:
    - ADCI2C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADCI2C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADCI2C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADCI2C\_MSK\_CH3 - channel 3 is selected for conversion
    - ADCI2C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin of the module
    - ADCI2C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
    - ADCI2C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
    - ADCI2C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)
- uint8\_t bCntChannels - the number of channels to be converted. The value has to correspond to the channels specified in the bConfig parameter.
- uint16\_t \*rgwValues - the array where the 16-bit value/s will be placed. It must be allocated according to bCntChannels value.

This function will read from I2C the values of the specified channels and store the received values in the buffer. Each value contains channel number on bits 13, 12 (see AD7991 documentation).

The order in which the channels values are stored is set according to the channel number, e.g. when using bConfig = ADCI2C\_MSK\_CH3 | ADCI2C\_MSK\_CH1, it will store channel 1 value first, then channel 3.

When calling the function, it is important that bCntChannels contains the number of channels selected in bConfig.

This is the lowest level function that can be used to read channels values, and it offers the best performance.

### **void GetIntegerValue1Ch(uint16\_t &wVal0, uint8\_t bConfig);**

#### *Parameters:*

- uint16\_t &wVal0 - parameter used to output the integer value read from channel

- uint8\_t bConfig - the configuration options, specifying which channels are selected for conversion and other options
  - It can be one or more (OR-ed) options from the list:
    - ADCI2C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADCI2C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADCI2C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADCI2C\_MSK\_CH3 - channel 3 is selected for conversion
    - ADCI2C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin of the module
    - ADCI2C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
    - ADCI2C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
    - ADCI2C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)
  - Default value: channel 0 is selected (ADCI2C\_MSK\_CH0)

This function converts one analog input channel according to the configuration byte. The user may select any channel, using bConfig parameter.

The configuration byte bConfig is set accordingly, so that it selects exactly one channel.

If more channels are selected, only the first channel is maintained selected. If no channels are selected, then the Channel 0 is selected.

If bConfig parameter is not provided, the Channel 0 is selected and no other configuration option is used.

If bConfig parameter is provided, along with it the user selects the channel to be converted.

**void GetPhysicalValue1Ch(float &fVal0, uint8\_t bConfig, float dReference);**

*Parameters:*

- float &fVal0 - parameter used to output the physical value corresponding to the channel
- uint8\_t bConfig - the configuration options, specifying which channels are selected for conversion and other options
  - it can be one or more (OR-ed) options from the list:
    - ADCI2C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADCI2C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADCI2C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADCI2C\_MSK\_CH3 - channel 3 is selected for conversion
    - ADCI2C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin
    - ADCI2C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
    - ADCI2C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
    - ADCI2C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)

- Default value: channel 0 is selected (ADC12C\_MSK\_CH0)
- float dReference - the value corresponding to the maximum converted value.
  - Default value: 3.3

This function converts one analog input channel according to the configuration byte. The user may select any channel, using bConfig parameter.

The configuration byte bConfig is set accordingly so that it selects exactly one channel.

If more channels are selected, only the first channel is maintained selected. If no channels are selected, then the Channel 0 is selected.

The function computes the physical value corresponding to the read (integer) value for the selected channel and to the reference value.

If bConfig parameter is not provided, the Channel 0 is selected and no other configuration option is used.

If bConfig parameter is provided, the user must select the channel to be converted.

If dReference parameter is not provided, 3.3 value is used as reference value.

**void GetIntegerValues2Ch(uint16\_t &wVal0, uint16\_t &wVal1, uint8\_t bConfig)**

*Parameters:*

- uint16\_t &wVal0 - parameter used to output the integer value read from the first channel (order of the channels is based on the channel number)
- uint16\_t &wVal1 - parameter used to output the integer value read from the second channel (order of the channels is based on the channel number)
- uint8\_t bConfig - the configuration options, specifying which channels are selected for conversion and other options
  - it can be one or more (OR-ed) options from the list:
    - ADC12C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADC12C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADC12C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADC12C\_MSK\_CH3 - channel 3 is selected for conversion
    - ADC12C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin
    - ADC12C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
    - ADC12C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
    - ADC12C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)
  - or default value: channels 0 and 1 are selected (ADC12C\_MSK\_CH0 | ADC12C\_MSK\_CH1)

This function converts two analog input channels according to the configuration byte. The user may select any combination of two channels, using the bConfig parameter.

The configuration byte bConfig is set accordingly so that it selects exactly two channels.

If more channels are selected, only the first two channels are maintained selected. If less than two channels are selected, then the first two channels (0 and 1) are selected by default.

If bConfig parameter is not provided, the first two channels are selected and no other configuration option is used.

If bConfig parameter is provided, the user must select the channels to be converted.

The order of the channels is set using the channel number, regardless of the order of the channels in the logical expression of bConfig parameter.

**void GetPhysicalValues2Ch(float &fVal0, float &fVal1, uint8\_t bConfig, float dReference)**

*Parameters:*

- float &fVal0 - parameter used to output the physical value corresponding to the first channel (order of the channels is based on the channel number)
- float &fVal1 - parameter used to output the physical value corresponding to the second channel (order of the channels is based on the channel number)
- uint8\_t bConfig - the configuration options, specifying which channels are selected for conversion and other options
  - it can be one or more (OR-ed) options from the list:
    - ADC12C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADC12C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADC12C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADC12C\_MSK\_CH3 - channel 3 is selected for conversion
    - ADC12C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin
    - ADC12C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
    - ADC12C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
    - ADC12C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)
  - Default value: channels 0 and 1 are selected (ADC12C\_MSK\_CH0 | ADC12C\_MSK\_CH1)
- float dReference - the value corresponding to the maximum converted value.
  - Default value: 3.3

This function converts two analog input channels according to the configuration byte. User may select any combination of two channels, using the bConfig parameter.

The configuration byte bConfig is set accordingly so that it selects exactly two channels.

If more channels are selected, only the first two channels are kept selected. If less than two channels are selected, then the first two channels (0 and 1) are selected.

For each channel, the function computes the physical values corresponding to the read (integer) values and to the reference value.

If bConfig parameter is not provided, the first two channels are selected and no other configuration option is used.

If bConfig parameter is provided, the user must select the channels to be converted.

The order of the channels is set using the channel number, regardless of the order of the channels in the logical expression of bConfig parameter.



If dReference parameter is not provided, 3.3 value is used as reference value.

**void GetIntegerValues3Ch(uint16\_t &wVal0, uint16\_t &wVal1, uint16\_t &wVal2, uint8\_t bConfig)**

*Parameters:*

- uint16\_t &wVal0 - parameter used to output the integer value read from the first channel (order of the channels is based on the channel number)
- uint16\_t &wVal1 - parameter used to output the integer value read from the second channel (order of the channels is based on the channel number)
- uint16\_t &wVal2 - parameter used to output the integer value read from the third channel (order of the channels is based on the channel number)
- uint8\_t bConfig - the configuration options, specifying which channels are selected for conversion and other options
  - it can be one or more (OR-ed) options from the list:
    - ADC12C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADC12C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADC12C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADC12C\_MSK\_CH3 - channel 3 is selected for conversion
    - ADC12C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin
    - ADC12C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
    - ADC12C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
    - ADC12C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)
  - Default value: channels 0, 1 and 2 are selected (ADC12C\_MSK\_CH0 | ADC12C\_MSK\_CH1 | ADC12C\_MSK\_CH2)

This function converts three analog input channels according to the configuration byte. The user can select any combination of three channels, using bConfig parameter.

The configuration byte bConfig is set accordingly so that it selects exactly three channels.

If more channels are selected, only the first three channels are maintained selected. If less than three channels are selected, then the first three channels (0, 1 and 2) are selected.

If bConfig parameter is not provided, the first three channels are selected and no other configuration option is used.

If bConfig parameter is provided, the user must select the channels to be converted.

The order of the channels is set using the channel number, regardless of the order of the channels in the logical expression of bConfig parameter.

**void GetPhysicalValues3Ch(float &fVal0, float &fVal1, float &fVal2, uint8\_t bConfig, float dReference)**

*Parameters:*

- float &fVal0 - parameter used to output the physical value corresponding to the first channel (order of the channels is based on the channel number)

- float &fVal1 - parameter used to output the physical value corresponding to the second channel (order of the channels is based on the channel number)
- float &fVal2 - parameter used to output the physical value corresponding to the third channel (order of the channels is based on the channel number)
- uint8\_t bConfig - the configuration options, specifying which channels are selected for conversion and other options
  - it can be one or more (OR-ed) options from the list:
    - ADC12C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADC12C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADC12C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADC12C\_MSK\_CH3 - channel 3 is selected for conversion
    - ADC12C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin
    - ADC12C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
    - ADC12C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
    - ADC12C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)
  - Default value: channels 0, 1 and 2 are selected (ADC12C\_MSK\_CH0 | ADC12C\_MSK\_CH1 | ADC12C\_MSK\_CH2)
- float dReference - the value corresponding to the maximum converted value.
  - Default value: 3.3

This function converts three analog input channels according to the configuration byte. The user may select any combination of three channels, using bConfig parameter.

The configuration byte bConfig is adjusted so that it selects exactly three channels.

If more channels are selected, only the first three are maintained selected. If less than three channels are selected, then the first three channels (0, 1 and 2) are selected.

For each channel, the function computes the physical values corresponding to the read (integer) values and to the reference value.

If bConfig parameter is not provided, the first three channels are selected and no other configuration option is used.

If bConfig parameter is provided, the user must select the channels to be converted.

If dReference parameter is not provided, 3.3 value is used as reference value.

**void GetIntegerValues4Ch(uint16\_t &wVal0, uint16\_t &wVal1, uint16\_t &wVal2, uint16\_t &wVal3, uint8\_t bConfig)**

*Parameters:*

- uint16\_t &wVal0 - parameter used to output the integer value read from the first channel (order of the channels is based on the channel number)
- uint16\_t &wVal1 - parameter used to output the integer value read from the second channel (order of the channels is based on the channel number)
- parameter used to output the integer value read from the third channel (order of the channels is based on the channel number)



- uint16\_t &wVal3 - parameter used to output the integer value read from the fourth channel (order of the channels is based on the channel number)
- uint8\_t bConfig - the configuration options, specifying which channels are selected for conversion and other options
  - it can be one or more (OR-ed) options from the list:
    - ADCI2C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADCI2C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADCI2C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADCI2C\_MSK\_CH3 - channel 3 is selected for conversion
    - ADCI2C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin
    - ADCI2C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
    - ADCI2C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
    - ADCI2C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)
  - Default value: all channels selected (ADCI2C\_MSK\_CH0 | ADCI2C\_MSK\_CH1 | ADCI2C\_MSK\_CH2 | ADCI2C\_MSK\_CH3)

This function converts the four analog input channels according to the configuration byte. The order of the channels is set using channel number, regardless of the order of the channels in the logical expression of bConfig parameter.

If less than four channels are selected in bConfig, then all the four channels are selected. If bConfig parameter is not provided, all the four channels are selected and no other configuration option is used.

**void GetPhysicalValues4Ch(float &fVal0, float &fVal1, float &fVal2, float &fVal3, uint8\_t bConfig, float dReference)**

*Parameters:*

- float &fVal0 - parameter used to output the physical value corresponding to the first channel (order of the channels is based on the channel number)
- float &fVal1 - parameter used to output the physical value corresponding to the second channel (order of the channels is based on the channel number)
- float &fVal2 - parameter used to output the physical value corresponding to the third channel (order of the channels is based on the channel number)
- float &fVal3 - parameter used to output the physical value corresponding to the fourth channel (order of the channels is based on the channel number)
- uint8\_t bConfig - the configuration options, specifying which channels are selected for conversion and other options
  - it can be one or more (OR-ed) options from the list:
    - ADCI2C\_MSK\_CH0 - channel 0 is selected for conversion
    - ADCI2C\_MSK\_CH1 - channel 1 is selected for conversion
    - ADCI2C\_MSK\_CH2 - channel 2 is selected for conversion
    - ADCI2C\_MSK\_CH3 - channel 3 is selected for conversion

- ADC12C\_MSK\_REF\_SEL - if this bit is set to 1, an external reference must be used and applied to the VIN3/VREF pin
- ADC12C\_MSK\_FLTR - filtering on SDA and SCL lines is enabled (0) or bypassed (1).
- ADC12C\_MSK\_BIT\_TRIAL\_DELAY - value 1 to turn off bit trial delay (see AD7991 documentation)
- ADC12C\_MSK\_SAMPLE\_DELAY - value 1 to turn off sample delay (see AD7991 documentation)
- Default value: all channels selected (ADC12C\_MSK\_CH0 | ADC12C\_MSK\_CH1 | ADC12C\_MSK\_CH2 | ADC12C\_MSK\_CH3)
  - float dReference - the value corresponding to the maximum converted value.
    - Default value: 3.3

This function converts the four analog input channels according to the configuration byte. For each channel, the function computes the physical values corresponding to the read (integer) values and to the reference value.

The order of the channels is set using channel number, regardless of the order of the channels in the logical expression of bConfig parameter.

If less than four channels are selected in bConfig, then all four are selected.

If bConfig parameter is not provided, all the four channels are selected and no other configuration option is used.

If dReference parameter is not provided, 3.3 value is used as reference value.

## Library usage

This section of the document describes the way the library is used:

- The PmodAD2 should be connected to the I2C #1 interface pins.
- The analog lines corresponding to channels to be converted are connected to the input lines of PmodAD2.
- Copy the library files according to the README.txt file.
- In the sketch, include the ADC12C library header file  
`#include <ADC12C.h>`
- In the sketch, include the I2C library header file. It is needed in order to access the I2C functionality.  
`#include <Wire.h>`
- In the sketch, instantiate one library object called, for example, myADC12C  
`ADC12C myADC12C;`
- In the sketch, use library functions by calls such as:  
`myADC12C.GetIntegerValues3Ch(w0, w1, w2);`

## Simple Demo

In order to run this demo, place the library and sketch and connect the wires corresponding to the analog values as explained in the Library usage section.

This demo performs the following operations:

- In the setup() function:
  - o Initializes the ADC12C library.
- In the loop() function:
  - o Reads the physical values corresponding to channels 0 and 1 of AD converter and displays them on the serial monitor.