PmodIA™ Reference Manual

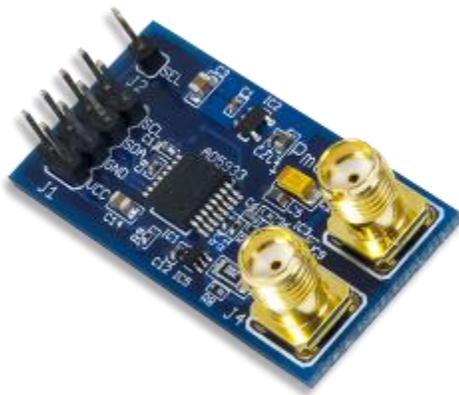**Revised April 15, 2016**
This manual applies to the PmodIA rev. A

Overview

The PmodIA is an impedance analyzer built around the Analog Devices AD5933 12-bit Impedance Converter Network Analyzer.

The PmodIA.

Features include:

- Impedance analyzer with 12-bit impedance converter
- Measure impedance values ranging from 100Ω to 10 MΩ.
- Programmable frequency sweep
- Programmable gain amplifier
- Optional external clock generation
- Small PCB size for flexible designs 1.6 in × 0.8 in (4.1 cm × 2.0 cm)
- 2×4-pin port with I²C interface
- Follows Digilent Interface Specification
- Library and example code available in resource center

1 Functional Description

The PmodIA utilizes Analog Devices AD5933 with its on-board frequency generator and analog-to-digital converter (ADC) to be able to excite an external unknown impedance at a known frequency. This known frequency is sent out through one of the SMA connectors. The frequency response is captured by the other SMA connector and sent to the ADC and a discrete Fourier transform (DFT) is performed on sampled data, storing the real and imaginary parts of the solution in the on-chip data registers. The magnitude of the unknown impedance as well as the relative phase of the impedance at each point in the generated frequency sweep can be calculated from these two data words.

1.1 I²C Interface

The PmodIA acts as a slave device using I²C communication protocol. The I²C interface standard uses two signal lines. These are I²C data and I²C clock. These signals map to the serial data (SDA) and serial clock (SCL) respectively on the PmodIA. (See Table 1.) The following instructions explain how to read and write to the device.

You must consider two protocols when writing to the PmodIA: the write byte/command byte and the block write. Writing a single byte from the master to the slave requires the master to initiate a start condition and send the 7-bit slave address. You must hold the read/write bit low to write to the slave device successfully. The PmodIA should set the slave address as 0001101 (0x0D) upon startup. After the slave acknowledges its address, the master must send the address of the register it wants to write to. Once the slave acknowledges receipt of this address, the master will send a single data byte that the slave should acknowledge with a return bit. The master should then issue a stop condition.

You can also use this protocol to set a pointer for a register address. After the master sends the slave address and write bit, and the slave responds with an acknowledge bit, the master sends a pointer command byte (10110000, or, 0xB0). The slave will assert an acknowledge bit and then the master will send the address of the register to point to in memory. The next time the device reads from or writes data to a register, it will occur at this address.

Note: The pointer must be set prior to using block write or block read protocols.

You can perform a block write protocol in a similar fashion to setting a pointer. Send the block write command (10100000, or, 0xA0) in place of the pointer command, and the number of bytes being sent (represented as a byte) will take the place of the register address with subsequent data bytes being zero indexed. Use the same two protocols when reading data from the PmodIA: receive byte and block read.

Connector J1 – I²C Communications		
Pin	**Signal**	**Description**
1, 2	SCL	I²C clock
3, 4	SDA	I²C data
5, 6	GND	Power supply Ground
7, 8	VCC	Power supply (3.3V/5V)

Table 1. Interface Connector Signal Description

1.2 Clock Source

The PmodIA has an internal oscillator that generates a 16.776 MHz clock to run the device. You can use an external clock by loading IC4 on the PmodIA and setting bit 3 in the control register (register address 0x80 and 0x81).

The PmodIA schematic provides a list of recommended oscillators. The schematic is available from the PmodIA product page at www.digilentinc.com.

1.3   Setting up a Frequency Sweep

The electrical impedance, Z, of a circuit can vary over a range of frequencies. The PmodIA allows you to easily set up a frequency sweep to find the impedance characteristics of a circuit.

First, you must set up an I²C interface between the host board and the PmodIA. The PmodIA requires three pieces of information to perform a frequency sweep: a starting frequency, the number of steps in the sweep, and the frequency increment after each step. The starting frequency and the increment per step parameters are stored as 24-bit words. The number of steps parameter is stored as a 9-bit word.

You can program the peak-to-peak voltage of the output frequency in the sweep by setting bits 10 and 9 in the control register. The peak to peak voltage needs to be set appropriately in relation to the impedance test. This is to avoid the internal op-amps from trying to deliver an output voltage or current beyond their maximum capability. It is recommended that when using the 20-ohm feedback resistor to set the peak to peak voltage to either the 200mV or 400mV and when using the 100K-ohm feedback resistor, set the peak to peak voltage at 1V.

Once the circuit has been excited, it takes some time to reach its steady state. You can program a settling time for each point in the frequency sweep by writing a value to register addresses 0x8A and 0x8B. This value represents the number of output frequency periods that the analog-to-digital converter will ignore before it starts sampling the frequency response. (See Table 2 for a list of registers and their corresponding parameters.)

Register Address	Parameter
0x80, 0x81	Control register (Bit-10 and Bit-9 set peak-to-peak voltage for the output frequency).
0x82, 0x83, 0x84	Start frequency (Hz)
0x85, 0x86, 0x87	Increment per step (Hz)
0x88, 0x89	Number of steps in sweep
0x8A, 0x8B	Settling time (Number of output frequency periods)

Table 2. Frequency sweep parameter storage registers.

You can calculate the 24-bit word to store at the register addresses for the start frequency and the increment per step parameters using the start frequency code and frequency increment code equations below. You can also find these equations and more information in the AD5933 data sheet.

$$\textbf{Start Frequency Code} := \left(\frac{Required\ Output\ Start\ Frequency}{\left(\frac{MCLK}{4} \right)} \right) * 2^{27}$$

$$\textbf{Frequency Increment Code} := \left(\frac{Required\ Frequency\ Increment}{\left(\frac{MCLK}{4} \right)} \right) * 2^{27}$$

Once you have set these parameters, perform the following steps to start the frequency sweep (paraphrased from the AD5933 data sheet):

1)   Enter standby mode by sending the standby command to the control register.

2) Enter the initialize mode by sending an initialize with start frequency command to the control register. This allows the circuit being measured to reach its steady state.
3) Start the frequency sweep by sending the start frequency sweep command to the control register.

1.4   Impedance Calculations

The analog-to-digital converter samples the frequency response from unknown impedances at up to 1MSPS with 12-bit resolution for every point in the frequency sweep. Before storing the measurements, the PmodIA performs a Discrete Fourier Transform (DFT) on the sampled data (1,024 samples for each frequency step). Two registers store the DFT result: the Real Register, and the Imaginary Register.

Electrical impedance contains both real and imaginary numbers. In Cartesian form, you can express impedance with the equation:

$$Z = \ Real + j * Imaginary$$

Where $Real$ is the real component, $Imaginary$ is the imaginary component, and j is an imaginary number (equivalent to $i = \ \sqrt{-1}$, in mathematics). You can also represent impedance in polar form:

$$Impedance = \ |Z|\angle\theta$$

Where |Z| is the magnitude and $\angle \ \theta$ is the phase angle:

$$|Z| = \ \sqrt{Real^2 + Imaginary^2}$$

$$\angle\theta = \ tan^{-1}\frac{Imaginary}{Real}$$

The PmodIA does not perform any calculations. After each DFT, the master device must read the values in the Real and Imaginary registers.

In order to calculate the true impedance, you must take into account the gain. You can find an example gain factor calculation in the AD9533 data sheet.

1.5   Temperature Readings

The PmodIA has a self-contained, 13-bit temperature sensor to monitor device temperature. Please refer to the AD5933 data sheet for more information on controlling this module.

1.6   Register Addresses

The AD5933 data sheet has a complete table of register addresses.

2   Physical Dimensions

The pins on the pin header are spaced 100 mil apart. The PCB is 1.6 inches long on the sides parallel to the pins on the pin header and 0.8 inches long on the sides perpendicular to the pin header.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Data Conversion IC Development Tools *category:*

Click to view products by Digilent *manufacturer:*

Other Similar products are found below :

EVAL-AD5063EBZ   EVAL-AD5422LFEBZ   EVAL-AD7265EDZ   EVAL-AD7641EDZ   EVAL-AD7674EDZ   EVAL-AD7719EBZ   EVAL-AD7767-1EDZ   EVAL-AD7995EBZ   AD9114-DPG2-EBZ   AD9211-200EBZ   AD9251-20EBZ   AD9251-65EBZ   AD9255-125EBZ   AD9284-250EBZ   AD9613-170EBZ   AD9627-125EBZ   AD9629-20EBZ   AD9709-EBZ   AD9716-DPG2-EBZ   AD9737A-EBZ   AD9787-DPG2-EBZ   AD9993-EBZ   DAC8555EVM   ADS5482EVM   ADS8372EVM   EVAL-AD5061EBZ   EVAL-AD5062EBZ   EVAL-AD5443-DBRDZ   EVAL-AD5570SDZ   EVAL-AD7450ASDZ   EVAL-AD7677EDZ   EVAL-AD7992EBZ   EVAL-AD7994EBZ   AD9119-MIX-EBZ   AD9148-M5375-EBZ   AD9204-80EBZ   AD9233-125EBZ   AD9265-105EBZ   AD9265-80EBZ   AD9608-125EBZ   AD9629-80EBZ   AD9648-125EBZ   AD9649-20EBZ   AD9650-80EBZ   AD9765-EBZ   AD9767-EBZ   ADS8322EVM   LM96080EB/NOPB   EVAL-AD5445SDZ   EVAL-AD5660EBZ