

Digilent PmodIA™ Impedance Analyzer Reference Manual

Revision: November 01, 2012

Note: This document applies to REV A of the board



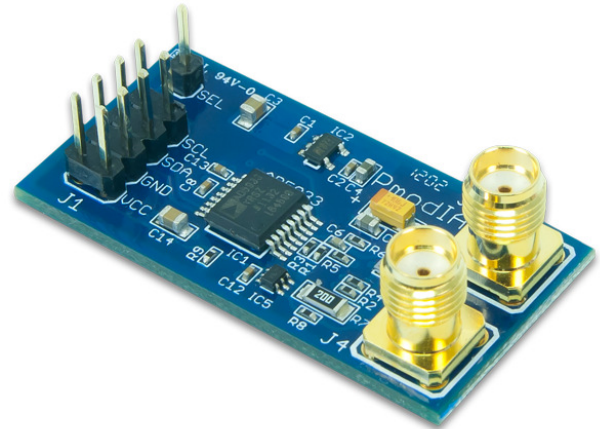
1300 Henley Court | Pullman, WA 99163
(509) 334 6306 Voice and Fax

Overview

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

Features include:

- I²C communication interface
- capable of measuring impedances ranging from 100Ω to 10 MΩ.
- programmable frequency sweep: start frequency, step increment, and number of steps.
- programmable gain amplifier
- external clock generation optional



clock (SCL) respectively on the PmodIA. (See Table 1.) The following instructions explain how to read and write to the device.

Functional Description

The PmodIA uses an 8-pin connector that allows for I²C communication. There is also a 1-pin connector that allows you to select either a high or low impedance measurement.

The PmodIA measures impedance by emitting an AC voltage at a known frequency and sampling the frequency response to identify the unknown impedance value. You can access the voltage output via the SMA connector J3. SMA connector J4 captures the response.

Please see the AD5933 data sheet, available from www.analog.com, for more detailed information on controlling the analyzer.

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

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

Table 1. Interface Connector Signal Description

Clock Source

The PmodIA has an internal oscillator that generates a 16.776MHz 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.

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.

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.)

Frequency Sweep Parameter Storage Registers

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)

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.

Start Frequency Code:

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

Frequency Increment Code:

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

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.

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:

$$Z = \text{Real} + j * \text{Imaginary}$$

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:

$$\text{Impedance} = |Z| \angle \theta$$

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

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

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

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.

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.

Register Addresses

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