SLDU022 User's Manual

PGA300

Pressure and Temperature Sensor Signal Conditioner

Preliminary

Copyright © 2016, Texas Instruments Incorporated

Preliminary

Revision History

Revision	Date	Descriptions/Comments
1.0		
2.0	31 -Jan - 2014	
2.2	15 - Feb -2014	
2.3	17 - Apr - 2014	
2.4	30 - Sep - 2014	
2.5	07 - Dec - 2014	
2.6	27 - Jan - 2015	LA GUI Style Guide Updates
3.0	25 - Mar - 2015	LabVIEW 2012 Migration, Aesthetic and Functional Enhancements
3.1	20 - APR - 2015	Icon Changes
3.2	11-May-2016	Guided Calibration and Accuracy verification test

The PGA300 User's Manual provides a general overview of the PGA300 Evaluation Module (EVM) GUI which includes,

- Description of the features of the GUI
- Functions to be considered while using the GUI
- Software Installations required

CONTENTS

Preliminary

Preliminary

TABLE OF FIGURES

Preliminary

Preliminary

Preliminary

Preliminary

Preliminary

1. GUI Software Installation

The PGA300 GUI allows the users to communicate to the PGA300 EVM.

The following section explains the location and the procedure for installing the software properly.

Ensure that no USB connections are made to the EVM until the installation is completed.

1.1. System Requirements

- Supported OS – Windows XP or higher
- Recommended RAM memory - 4GB or higher
- Recommended CPU Operating Speed – 3.3 GHz or higher

1.2. Installation Procedure

The following procedure will help you install the PGA300 GUI

1. Double click on the **Setup.exe** from the Volume folder as shown in Fig.1.

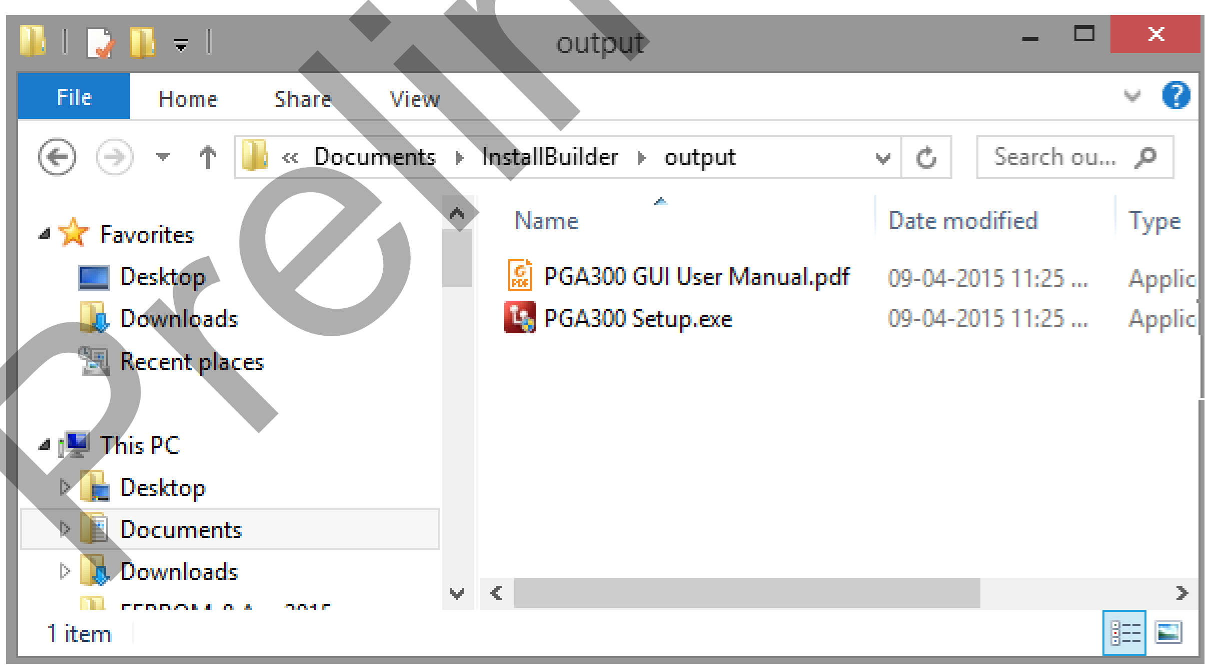


Fig 1: Setup.exe from Volume folder

Preliminary

A screen shown in *Fig 2: Installation Initialization* will appear.

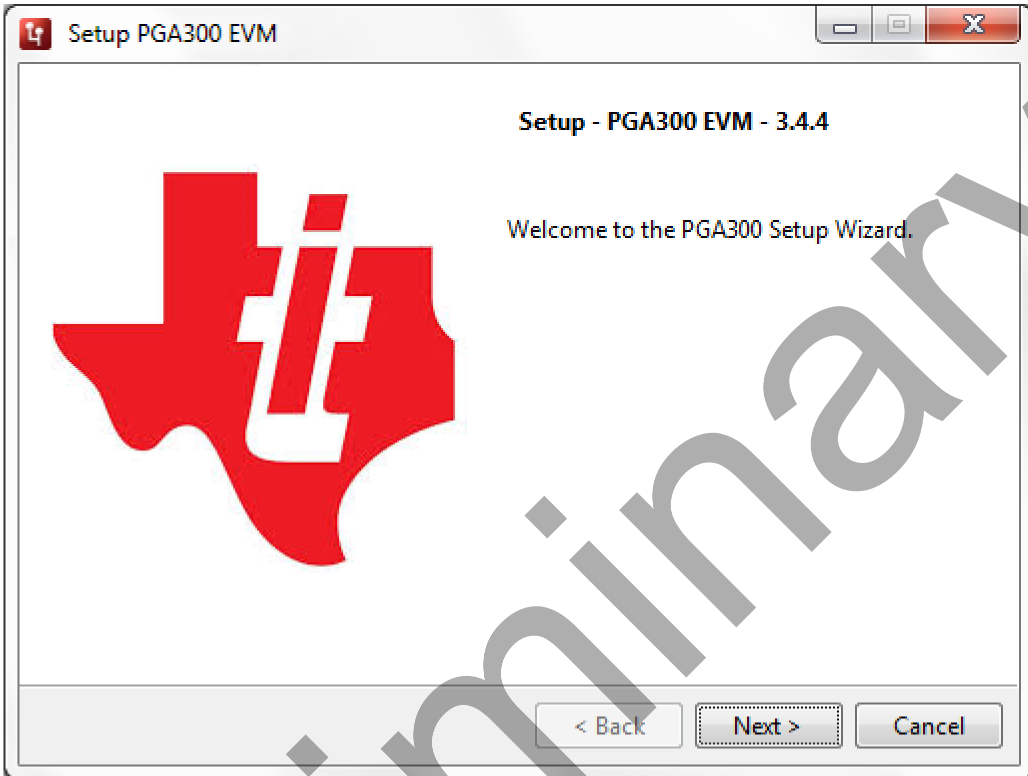


Fig 2: Installation Initialization

Preliminary

2. The License Agreement for PGA300 GUI will appear as shown below *in Fig 3: License Agreement - GUI*. Please read through the agreement carefully and enable the "I Accept the License Agreement" radio button and press the **Next»** button

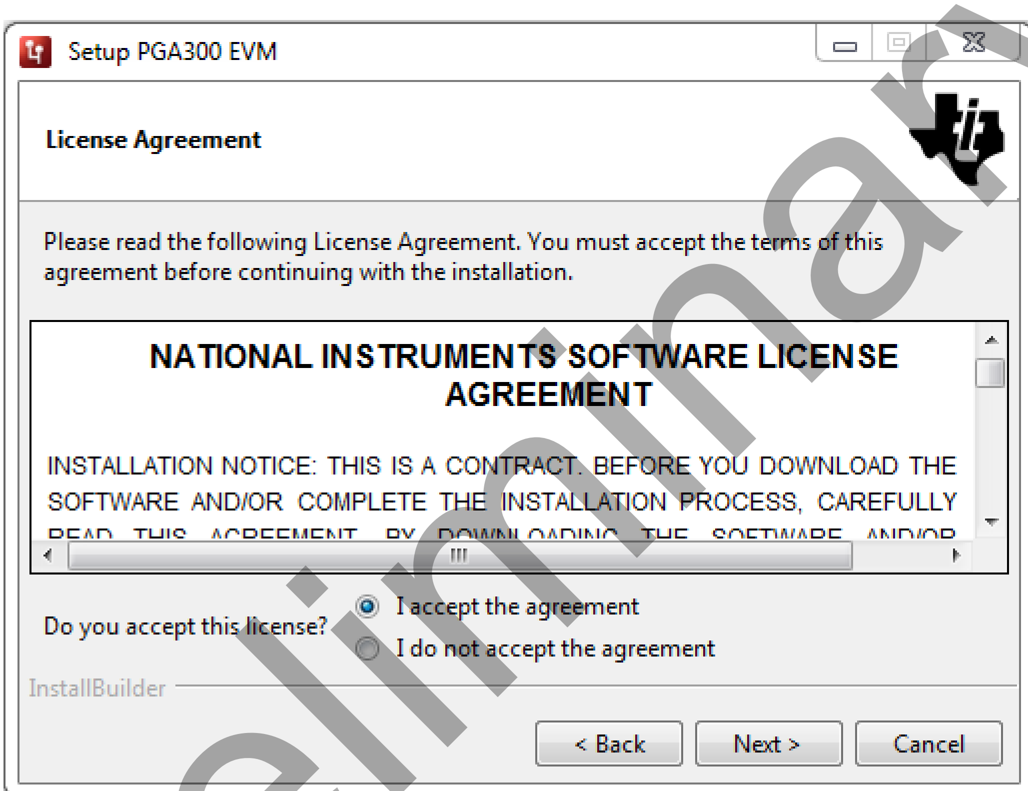


Fig 3: License Agreement - GUI

Preliminary

The License Agreement for Python 2.7 will appear as shown below *in Fig 4: License Agreement - Python*. Please read through the agreement carefully and enable the "I accept the agreement" radio button and press the **Next»** button

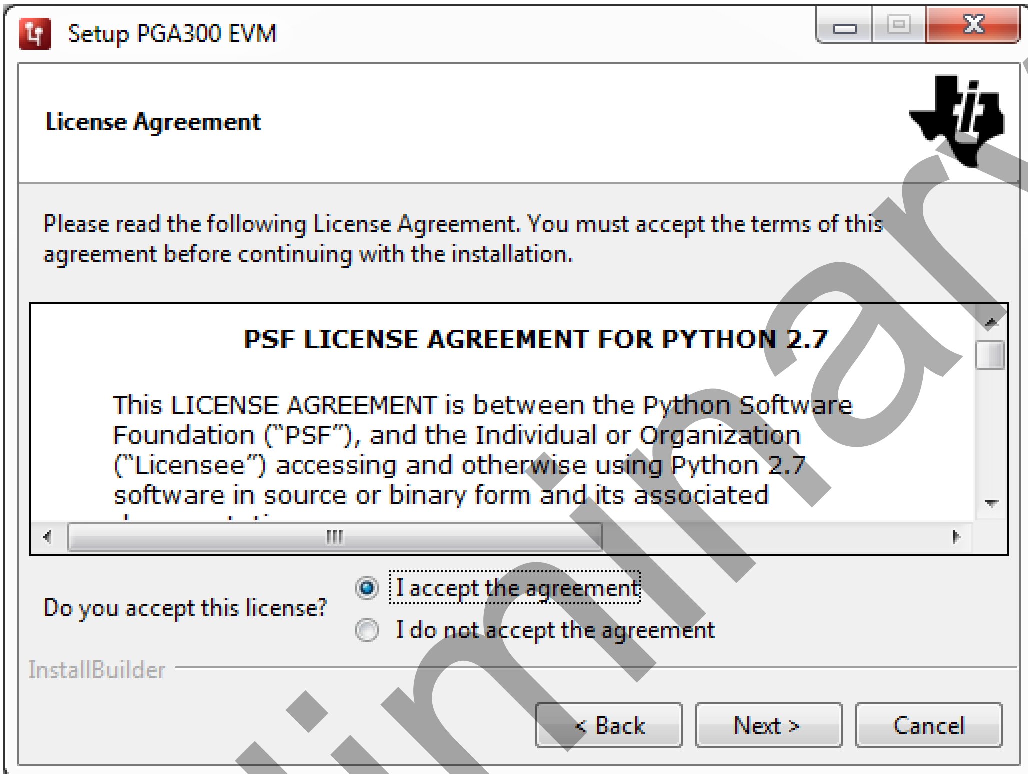


Fig 4: License Agreement - Python

It is highly recommended to keep the default values as provided in the installer.

Preliminary

3. Set the destination directories for the PGA300 GUI installation and press the **Next»** button as shown in *Fig 5: Destination Directory* .

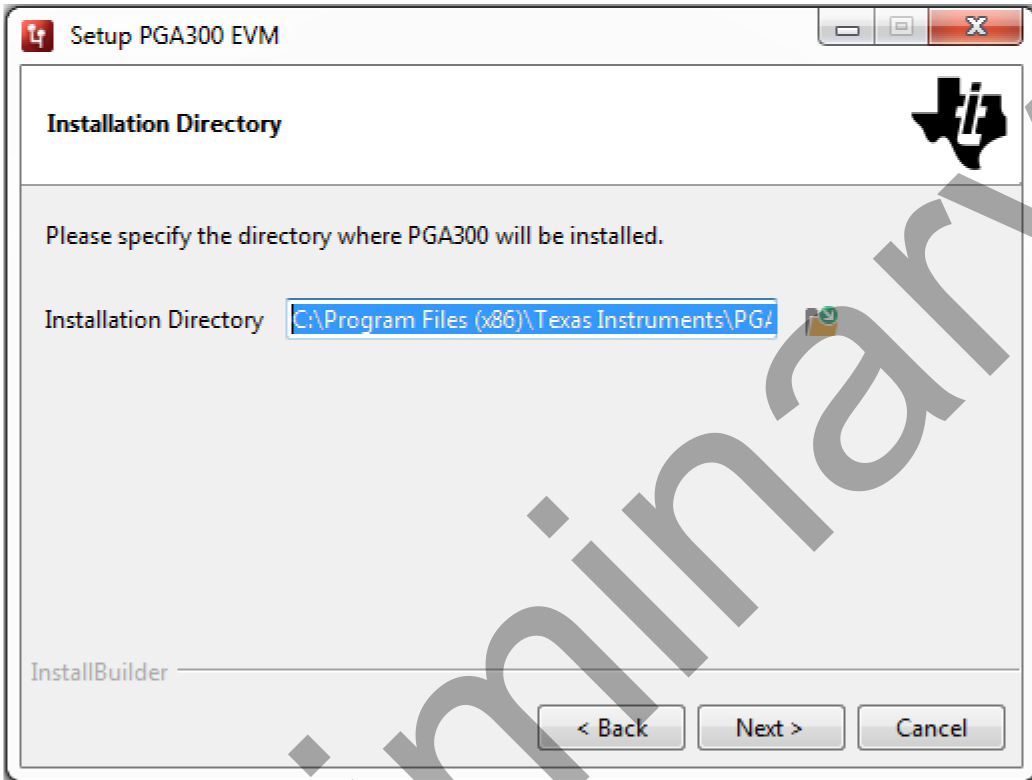


Fig 5: Destination Directory

Preliminary

4. A screen as shown in *Fig 6: Start Installation* will appear. Click **Next»** to begin installation.

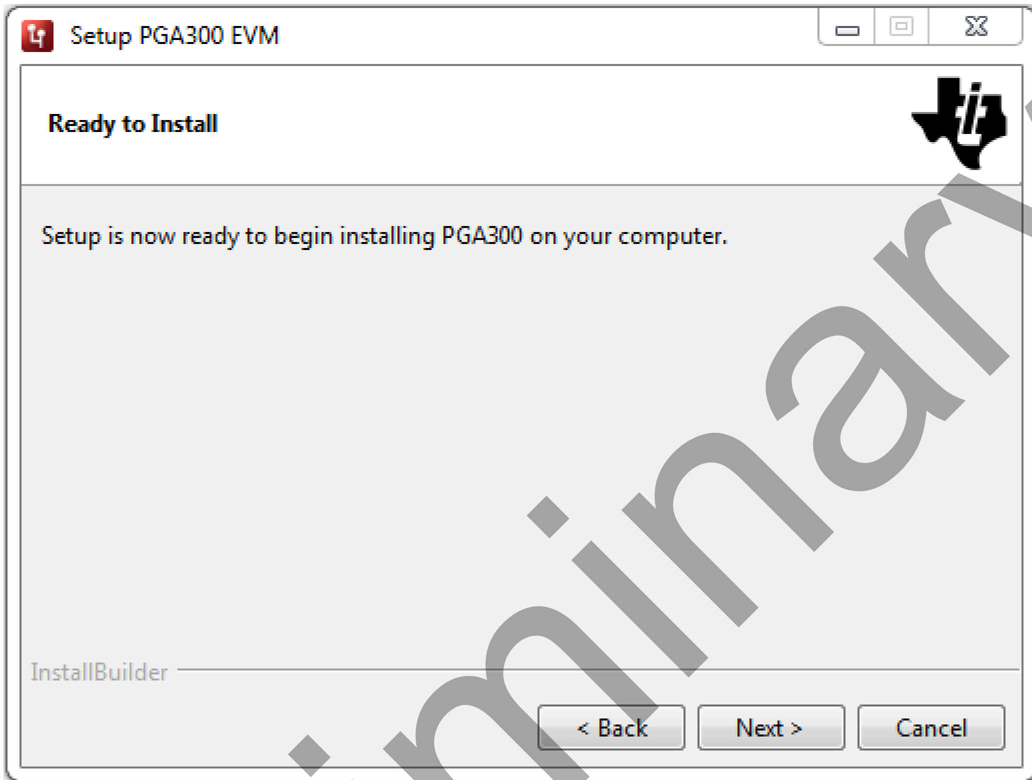


Fig 6: Start Installation

Preliminary

5. The installer will begin self-extraction and proceed with the installation as shown in *Fig 7: Installation in Progress*.

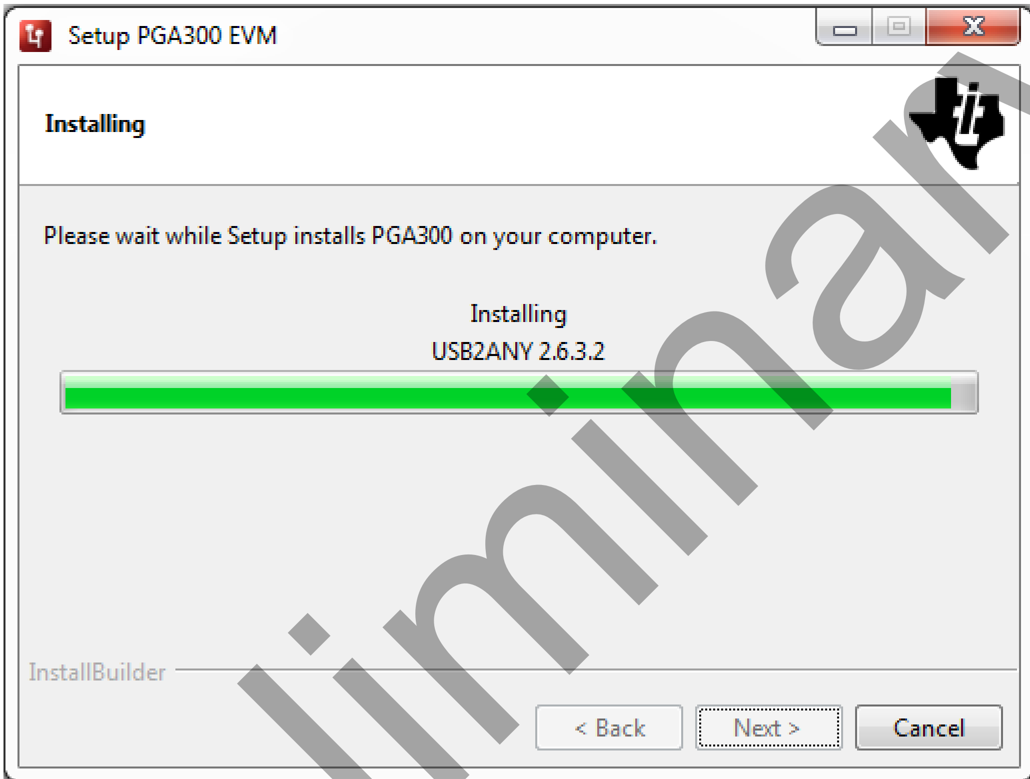


Fig 7: Installation in Progress

Preliminary

6. Towards the end of PG300 GUI installation, Python installation will start. Select the required option and click on **Next>>** button

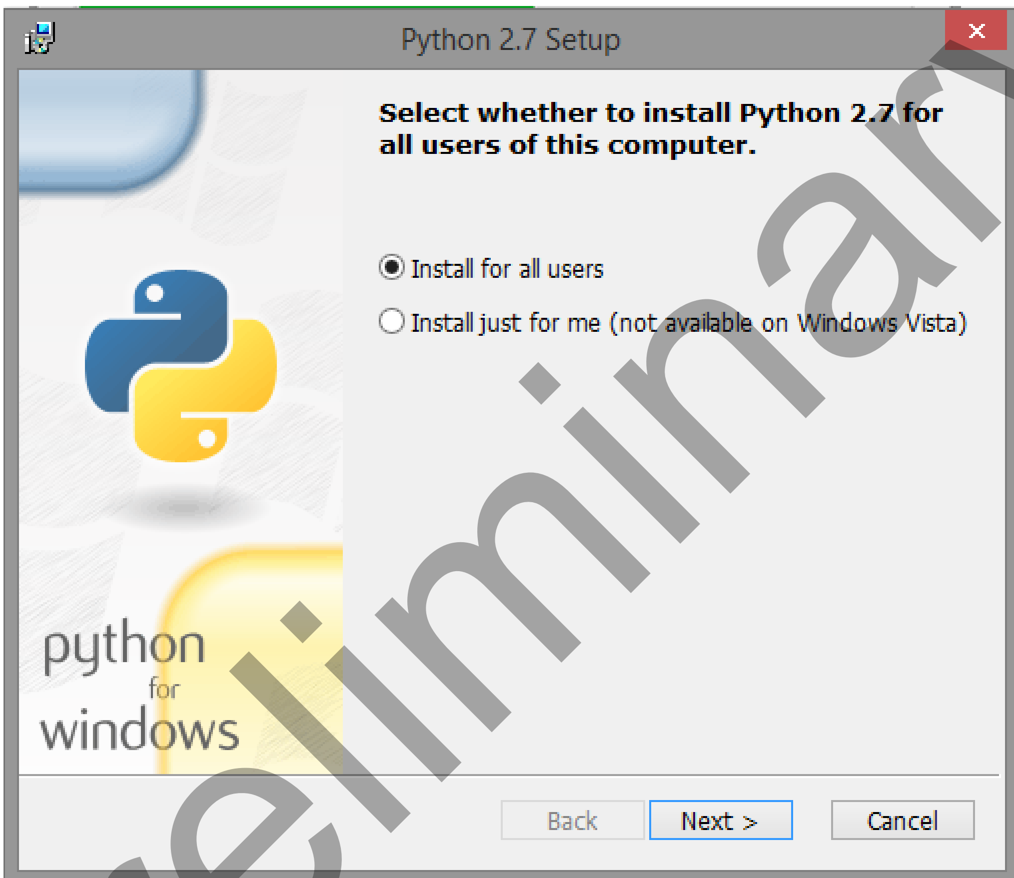


Fig 8: Python Installation

Preliminary

7. Select the installation folder for Python and click **Next>>** button

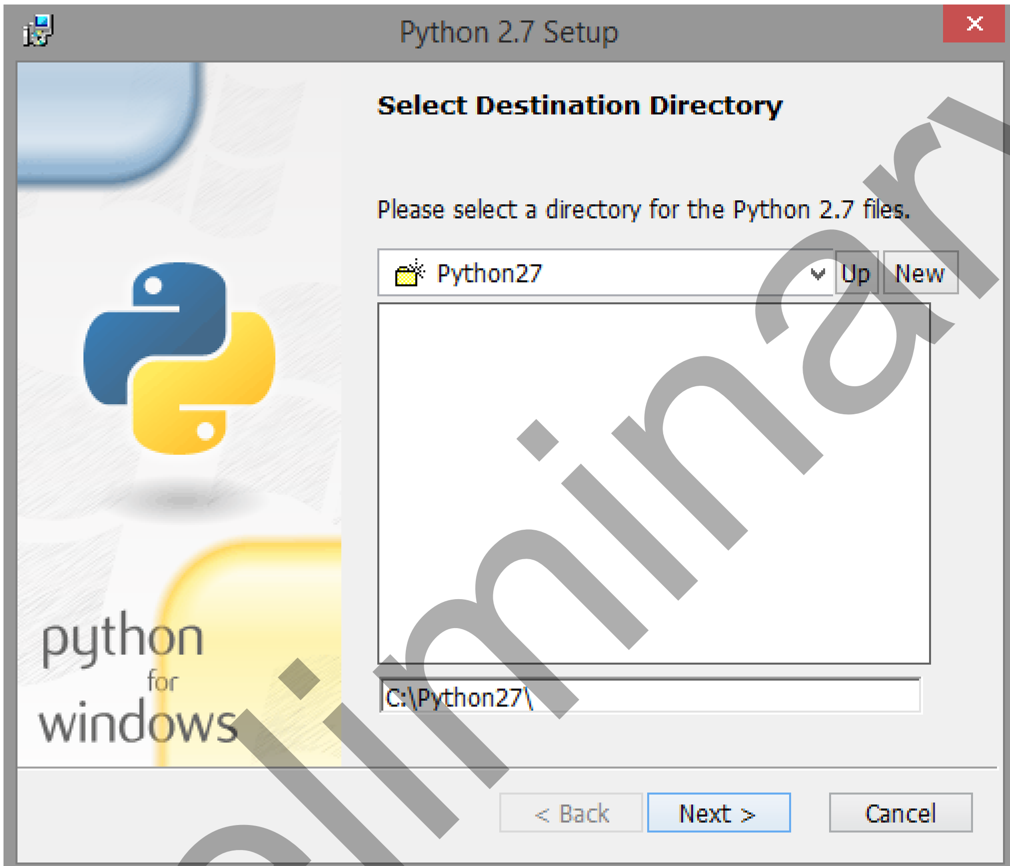


Fig 9: Python Installation Directory

Preliminary

8. A dialog as shown in *Fig 10: Python Customization* will appear. Click **Next>>** button from the dialog

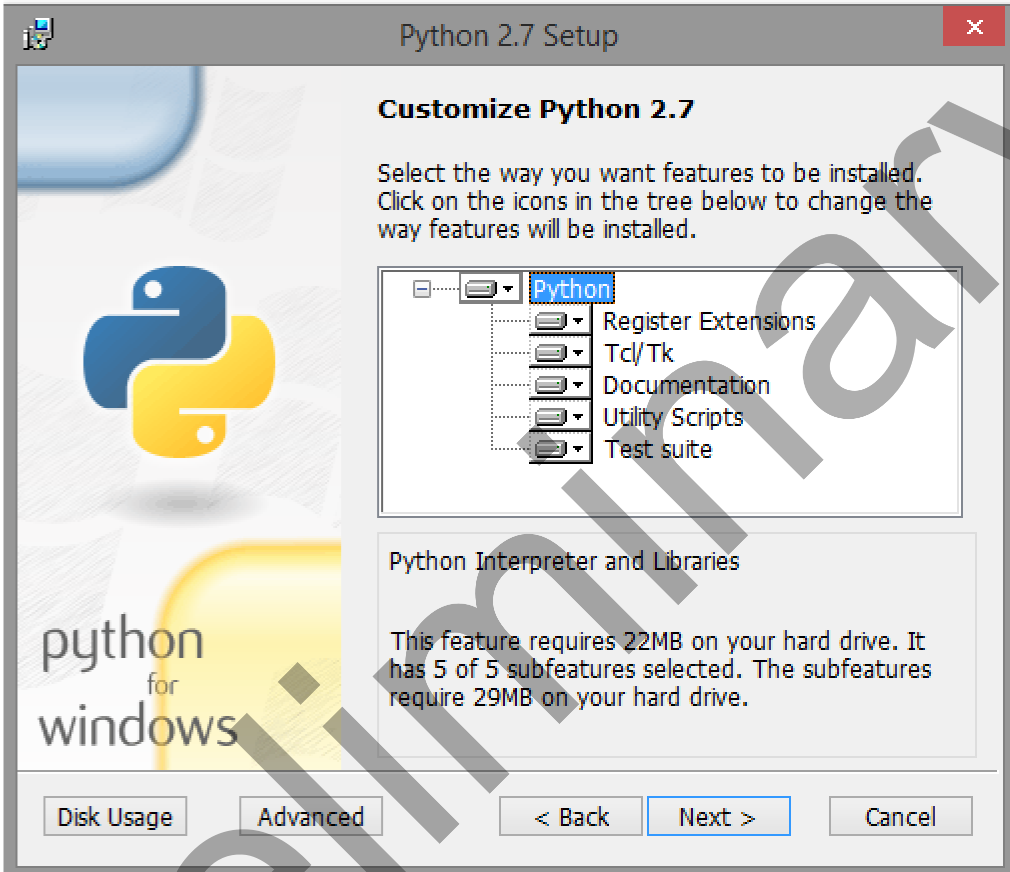


Fig 10: Python Customization

Preliminary

9. Python installation will start. The progress will be shown as *in Fig 11: Python Installation Progress*

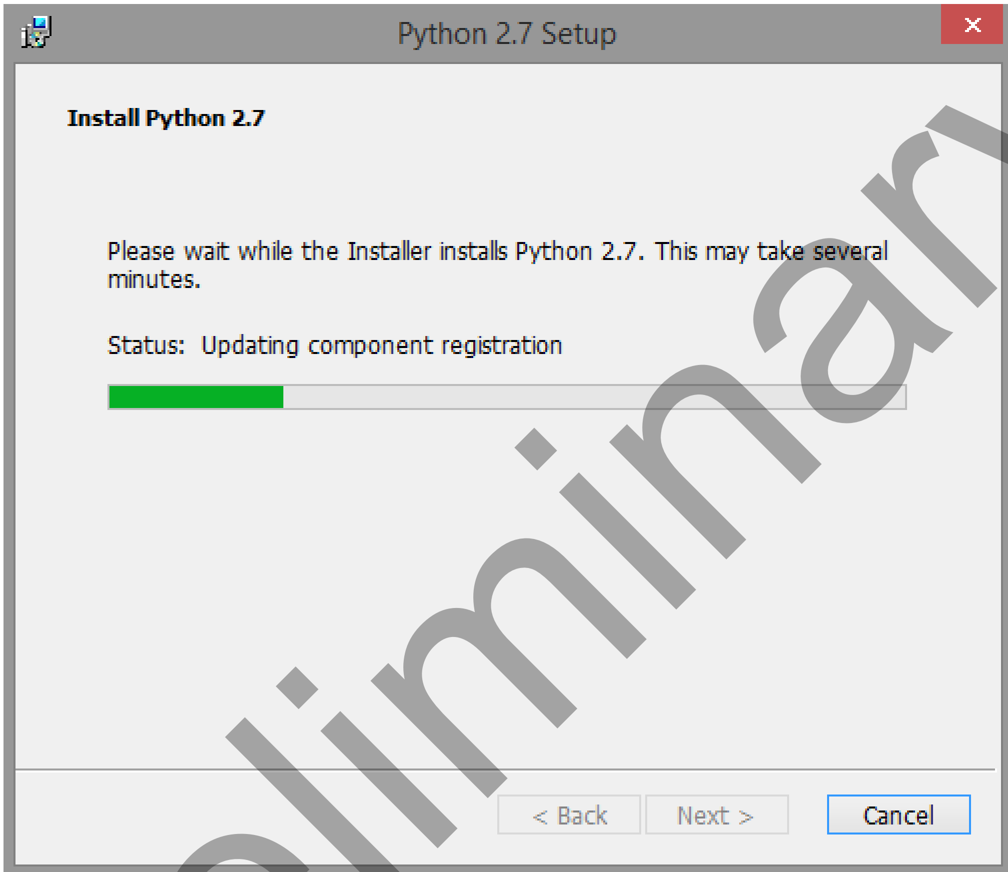


Fig 11: Python Installation Progress

Preliminary

10. After Python installation is finished, click on **Finish** button

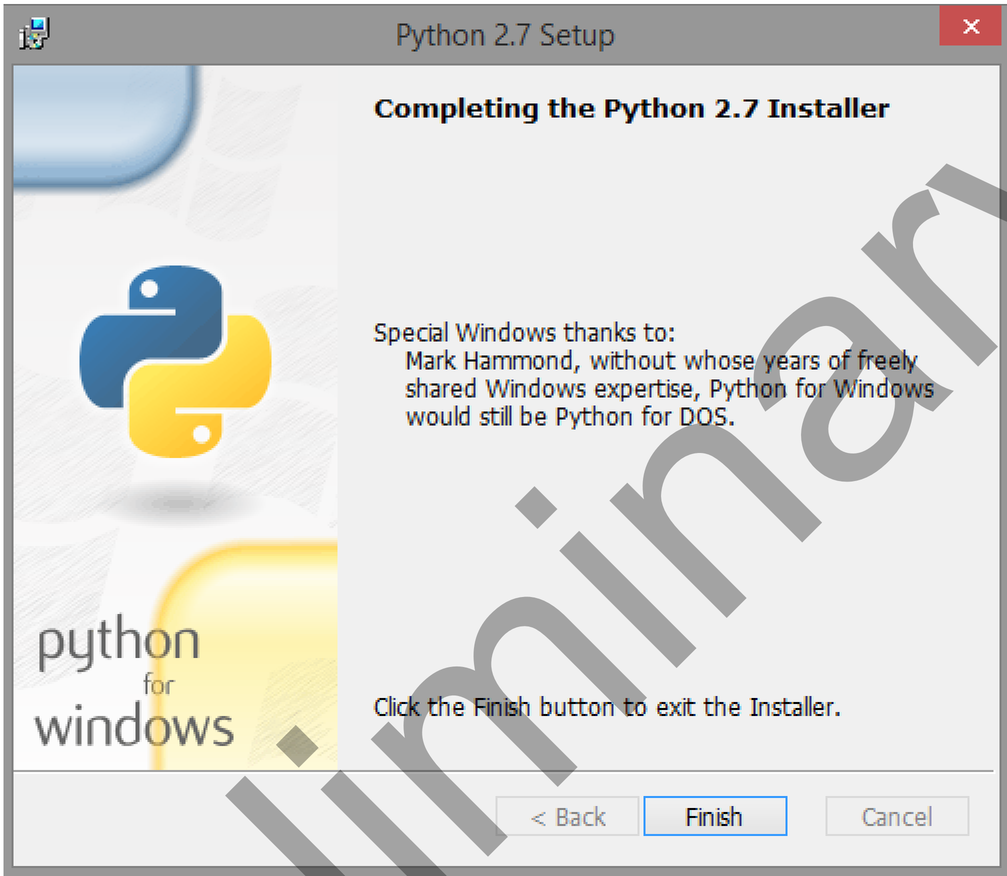


Fig 12: Python Installation Complete

Preliminary

11. The USB2ANY installation will start after Python installation is finished. From the dialog as shown in *Fig 13: USB2ANY Installation* click on **Next>>** button

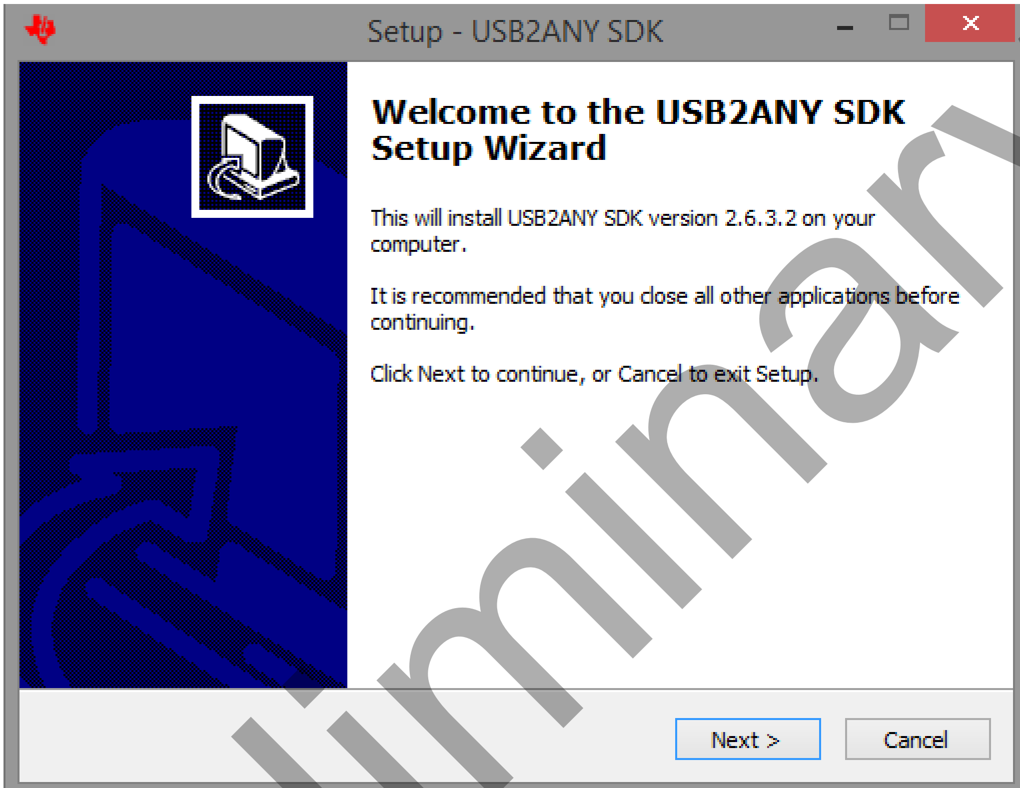


Fig 13: USB2ANY Installation

Preliminary

12. The License Agreement will appear as shown below in *Fig 14: USB2ANY license Agreement*. Please read through the agreement carefully and enable the "I Accept the License Agreement" radio button and press the **Next»** button

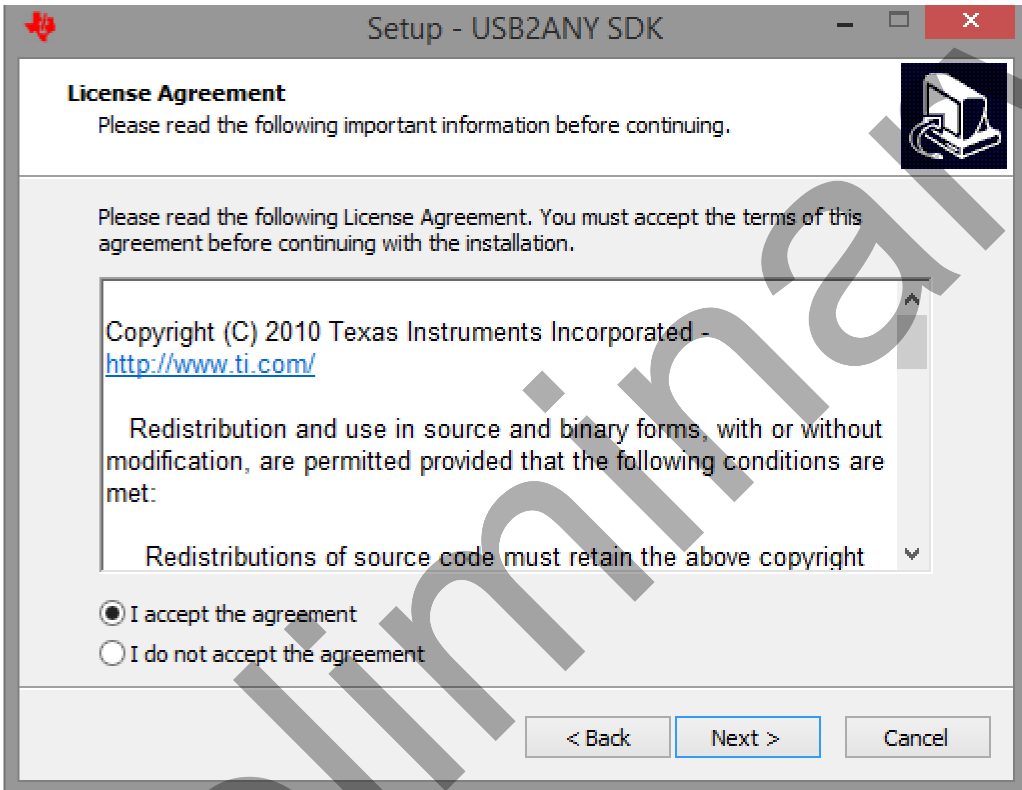


Fig 14: USB2ANY license Agreement

Preliminary

13. Set the destination directories for the PGA300 GUI installation and press the **Next»** button as shown in *Fig 15: USB2ANY Installation Folder.*

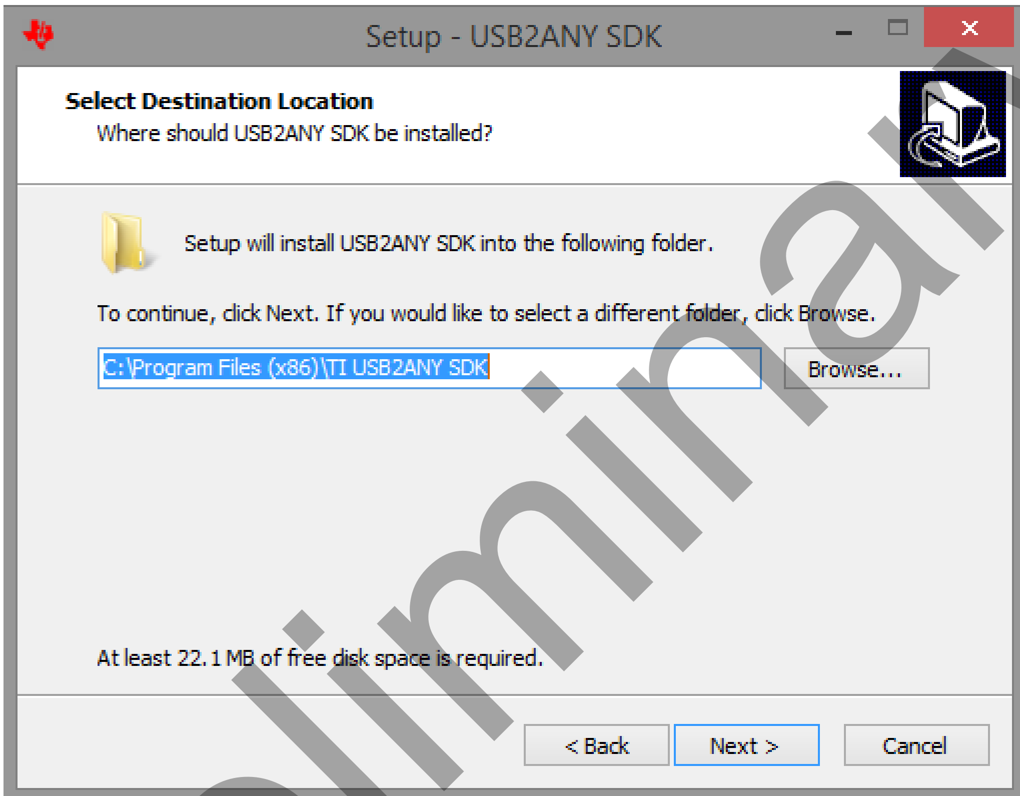


Fig 15: USB2ANY Installation Folder

Preliminary

14. After the installation is complete click on **Finish** button.

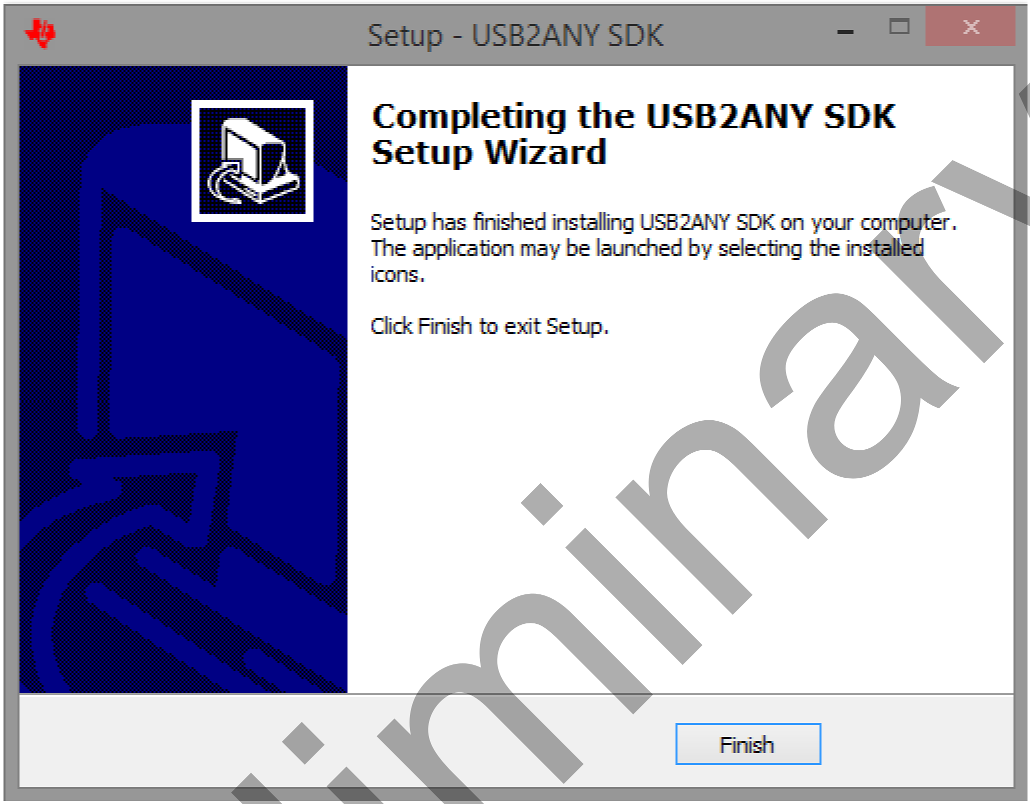


Fig 16: USB2ANY Installation Complete

Preliminary

15. Screen as shown in *Fig 17: Installation Complete* will appear that denotes the end of the PGA300 GUI installation.

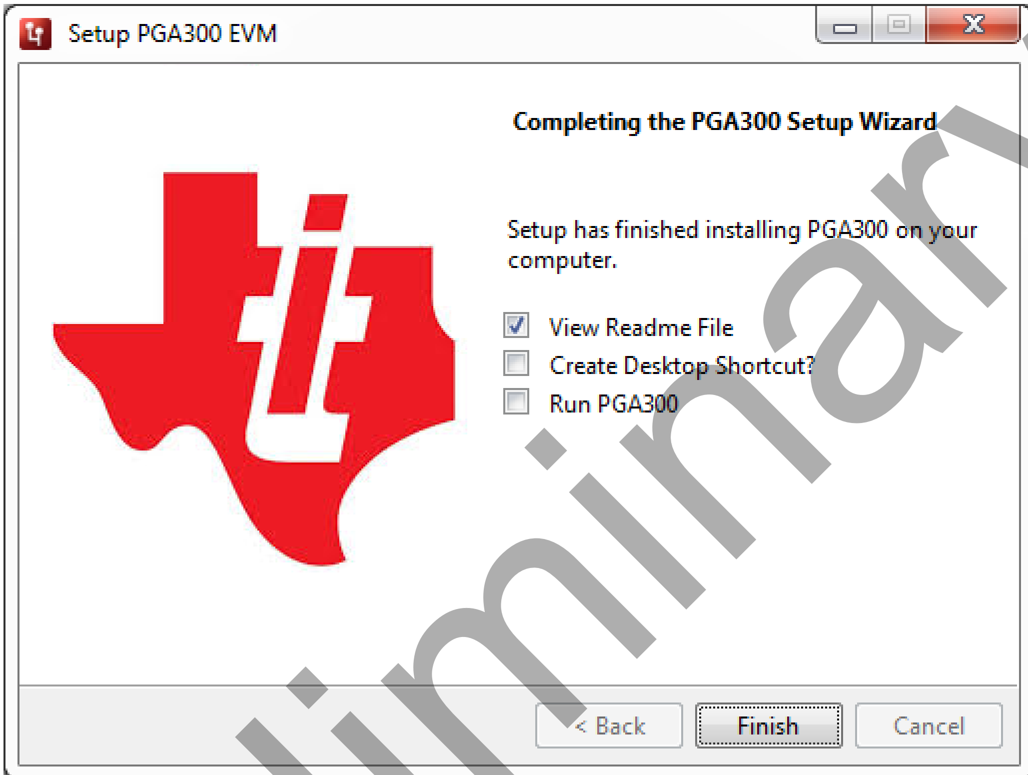


Fig 17: Installation Complete

The installer will also install Python 2.7, USB2ANY SDK along with the GUI installation.

The PGA300 GUI requires the following software to be installed before the GUI is executed.

- **National Instruments LabVIEW Run-Time Engine 2012 from the below link.**

  http://www.ni.com/download/labview-run-time-engine-2012/3433/en/

The PGA300 GUI executable has been built in LabVIEW 2012 (32-Bit) version and it expects the LabVIEW Run-Time Engine version to be LabVIEW Run-Time Engine 2012 (32-Bit) Version.

Preliminary

2. PGA300 – USER INTERFACE

This section gives a detailed description of the features of the PGA300 GUI.

The PGA300 GUI is an intuitive UI, for the PGA300 device and EVM that allows the user to read and configure the registers of the PGA300 device and control some EVM components.

It is advisable to launch the PGA300 GUI with administrator privileges

When the PGA300 GUI is launched, the following screen pops up indicating that the GUI is initializing.

Fig 18: GUI Initializing

Preliminary

When the PGA300 GUI is launched for the first time, a pop-up window (as shown in *Fig 19: Update Registry*) appears on the screen.

1. Press the **YES** button on the pop-up for the GUI to run as expected.

   This updates the python installation path in the Windows registry, so that, when the macro window is launched from the GUI, the python IDLE IDE is called.

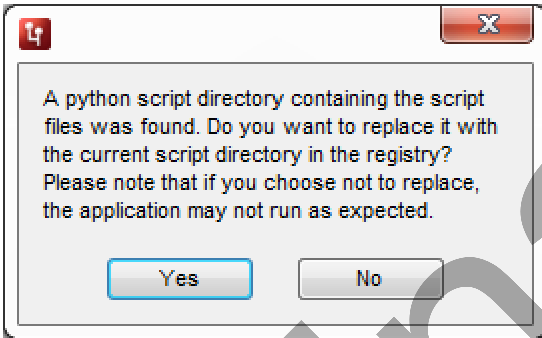


Fig 19: Update Registry

2. Once the PGA300 GUI is launched, it can be invoked in two different ways:
   - Interface Connected
   - Interface Disconnected

   'Interface Disconnected' mode is invoked when the USB2ANY is not connected. When the PGA300 GUI is loaded with no hardware interface connected, a message pops-up as shown *in Fig 20: Device Communication Error*. This allows the user to either run the PGA300 GUI in 'Interface Disconnected' mode or to terminate the usage.

Fig 20: Device Communication Error

Preliminary

When the user continues in 'Interface Disconnected' mode, the checkbox at the top of the GUI is set. This will show that the PGA300 GUI is running in 'Interface Disconnected' mode. When the GUI is in 'Interface Disconnected' mode the UI controls may appear to function as if the device is connected and will read simulated data.

2.1. PGA300 GUI Overview

The PGA300 GUI consists of the following pages,

- **Device Calibration**
  - ✓ OWI Configuration
  - ✓ EEPROM Config – Analog
  - ✓ ADC and DAC Calibration
  - ✓ EEPROM Config – Digital
  - ✓ Guided Calibration
- **Debug**
  - ✓ ADC settings
  - ✓ DAC settings
  - ✓ ADC Graph
  - ✓ Bridge Configuration
  - ✓ EEPROM Page
  - ✓ Calibration demo
  - ✓ Low Level Configuration

DEVICE CALIBRATION

OWI-Configuration

- Speed settings for OWI
- Rloop resistance configuration
- Additional voltage configuration

EEPROM Config - Analog

- This page allows the user to write and read specific EEPROM register.

Preliminary

- Analog P Gain can be calculated from this page.

ADC and DAC Calibration

- The ADC and DAC Calibration page is used to calculate the TC & NL Compensation coefficients.
- The page provides options to perform DAC Code Calculation, ADC Capture and Calculate coefficients.

EEPROM Config – Digital

- This page allows the user to write and read calculated TC and NL Coefficients, Scaling Factors and Filter Coefficients.
- EEPROM Variables can be read and written from this page.

Guided Calibration

- This page allows the user to calibrate the device in sequence of steps.
- Calibration can be verified using Accuracy verification test.

DEBUG

ADC Settings

- The ADC MUX settings page is used to control the MUX of the temperature and the pressure ADC path.
- The page has options to read the Temperature ADC and Pressure ADC and display the Temperature and Voltage values.

DAC Settings

- The DAC settings page is used to control the DAC configurations.
- User can read/write the DAC data and DAC Gain can be configured.

ADC Graph

- The ADC Graph allows the user to configure the ADC settings for Capture.
- It has options to perform continuous ADC capture with the configured settings.

Bridge Configuration

Preliminary

– The Bridge Configuration page is used to operate the Internal Bridge Circuit present in the PGA300 EVM.

Calibration Demo

– This page demonstrates the PGA300 Calibration process.
– It automatically performs the Calibration and displays the results.

EEPROM Page

– The EEPROM Page gives EEPROM access to user.
– The page has options to load EEPROM, Read EEPROM, reading EEPROM Cache, updating EEPROM data from file, saving EEPROM data to file.

Low Level Configuration

– The Low Level Configuration Page lists down all the registers that are present in the PGA300 GUI.
– This page can be used to write to and read from, the fields of the registers of the PGA300 device.

The PGA300 resources could be accessed by the **Compensation Mode** or by the **OWI Mode**. This can be controlled by selecting the microcontroller button at the top left corner of the PGA300 GUI as shown in *Fig 21: Device mode selection.* OWI Mode can be activated by clicking the **Activate OWI** button.

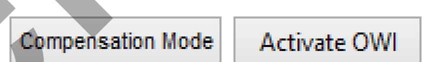


Fig 21: Device mode selection

The microcontroller can be put into reset state by writing 1 to the **MICRO_RESET** bit in the **MICRO_INTERFACE_CONTROL** register. Access to OWI Mode is enabled internally when switching the interface mode to OWI. This in turn writes **3** to MICRO_INTERFACE_CONTROL register. Disabling any of the bits that corresponds to the OWI Mode will result in disabling the OWI.

Preliminary

2.2. PGA300 Communication Interface

The Communication Interface is used to communicate (read and write) with the device's registers.

- One-Wire Interface (OWI)

The PGA300 device operates using the One-Wire Interface (OWI) and acts as a slave device.

OWI activation

When OWI is successfully activated, the status will be shown at the top right corner of the GUI. The EEPROM has to be unlocked to perform read or write into EEPROM pages. Click on **Unlock EEPROM** check box to access the EEPROM pages from the GUI and click on **Activate OWI** button.

Fig 22: Device Status

When device is in write only mode, register read operations cannot be carried out. To restore the device to Read & Write mode, the TEST_MUX_DAC_EN bit (Bit 0 of AMUX_CTRL register) should be reset. This can be done by changing the value from the 'ADC & DAC Calibration' page or the 'Low Level Configuration' page as shown in *Fig 23: TEST_MUX_DAC_EN bit in ADC & DAC Calibration* or *Fig 24: TEST_MUX_DAC_EN bit in low level configuration*.

Preliminary

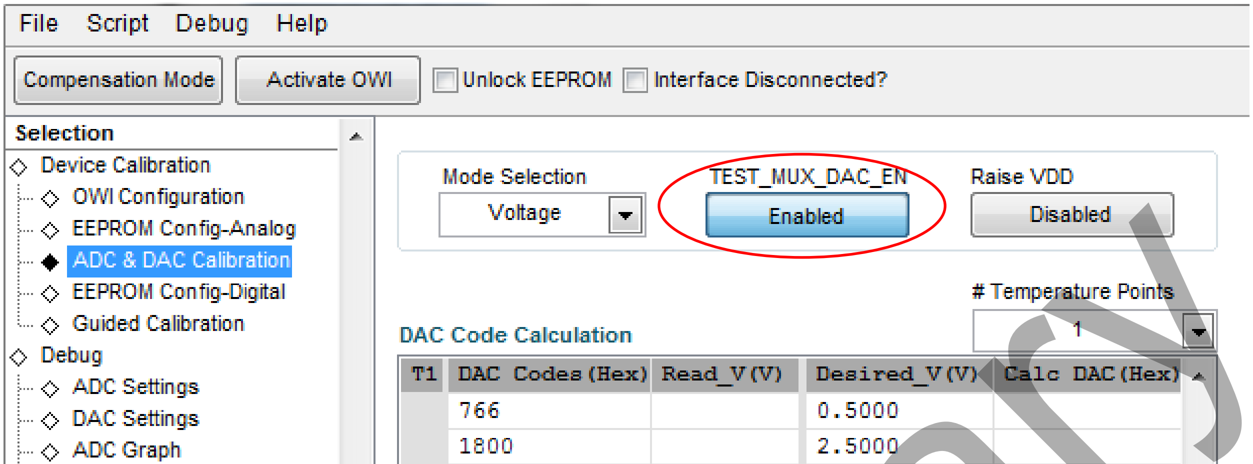


Fig 23: TEST_MUX_DAC_EN bit in ADC & DAC Calibration

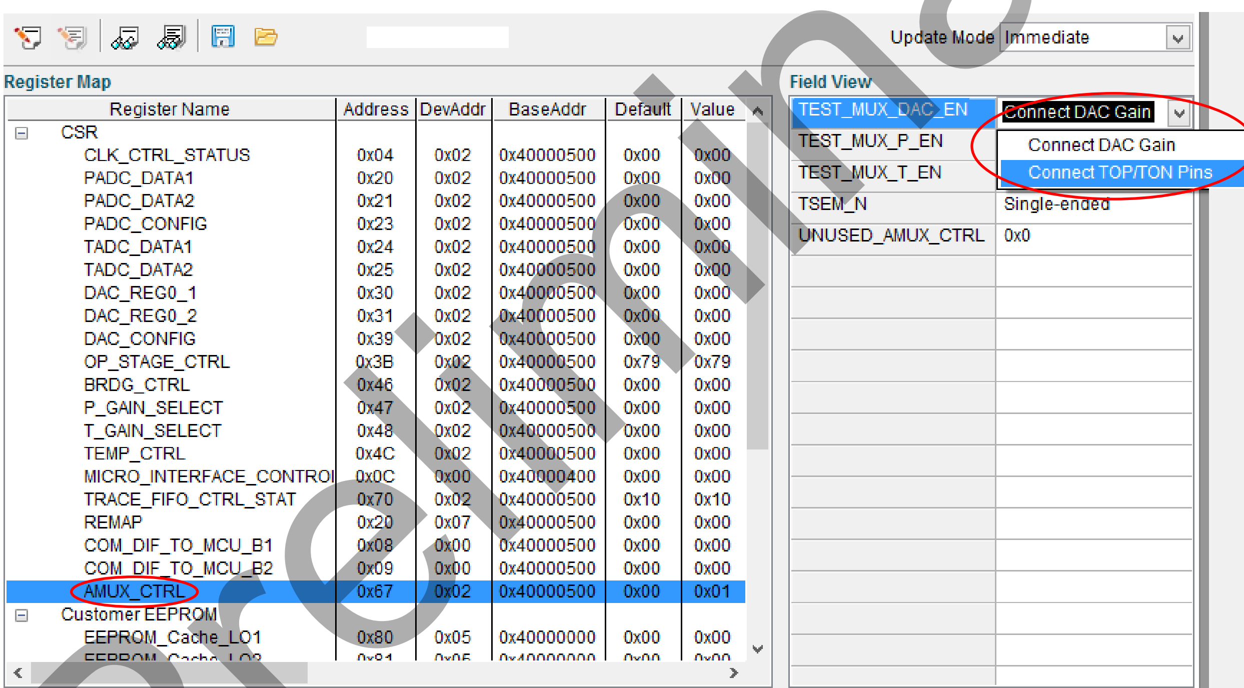


Fig 24: TEST_MUX_DAC_EN bit in low level configuration

Preliminary

2.2.1. Overview of OWI Interface

- The OWI digital communication is a master-slave communication link in which the PGA300 operates as a slave device only.
- The master device controls when the data transmission begins and ends.

- The slave device does not transmit data back to the master until it is commanded to do so by the master. The VDD pin of PGA300 is used as OWI interface, so that when PGA300 is embedded inside a system module, only two pins are needed (VDD and GND) for communication.
- The OWI master communicates with PGA300 by modulating the voltage on the VDD pin while PGA300 communicates with the master by modulating current on VDD pin.
- **OWI Write**
  - No specific configuration is needed to write using the OWI.
  - Write the data in the format defined by the data sheet
- **OWI Read**
  - The following sequence is executed for the read operation
  - Check if the USB2ANY buffer has residual data and empty the buffer
  - Send the read command with the sync byte, read initialization and read response command as defined in the datasheet
  - Check for data in the buffer until it receives it
  - Read the data from the buffer and display it. This will read all data available in the buffer. Ideally there should be only one byte of data

2.3. Page Selection

2.3.1. Device Calibration

- The Device Calibration Page provides an abstract view of the device.
- Different functions of the device are represented structurally.
- Each control that is placed in the high level configuration page is linked to a register or a field of the device.
- The Device Calibration pages consist of various high level functions. Each section below explains a different function of the PGA300 EVM.

Preliminary

2.3.1.1 OWI-Configuration

- The OWI-Configuration page contains the configuration parameters necessary for the OWI communication protocols.
- The default values are displayed when the GUI is loaded.

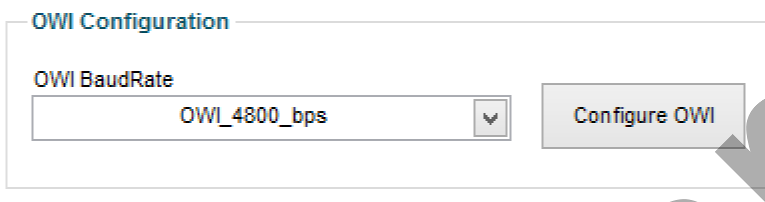


Fig 25: Interface Configuration

- The user has the provisions to change Baud **rate** for the OWI protocol.
- The default settings for OWI is 4800 bps.
- After the change is made, the **Configure OWI** button has to be clicked for the changes to take effect.
- The controls **Rloop** and **Additional Voltage** are used to configure the Potentiometers on the EVM board. These Potentiometers are configured using I2C protocol. Select desired values and click on **Configure POT** button for the configuration to take effect.

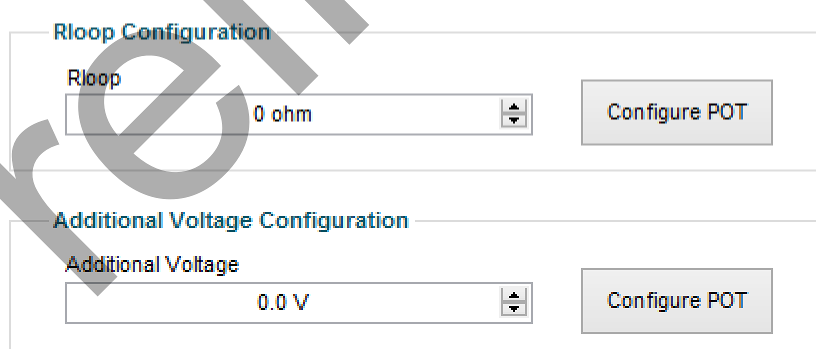


Fig 26: POT Configuration

OWI Configurations

OWI configuration involves 3 settings,

Preliminary

1. Setting the baud rate to the selected configuration.
2. Setting the receiver mode to 2.
    a. Configures the communication to be half duplex.
    b. USB2ANY transmits data with 2 stop bits but will accept data with single stop bit.
3. Setting the USB2ANY internal timeout to make sure USB2ANY waits for enough time to receive the data while at lower baud rates.

BaudRate	Timeout
<600 bps	200ms
600 to 4800 bps	100ms
>4800 bps	25ms

2.3.1.2 EEPROM Config - Analog

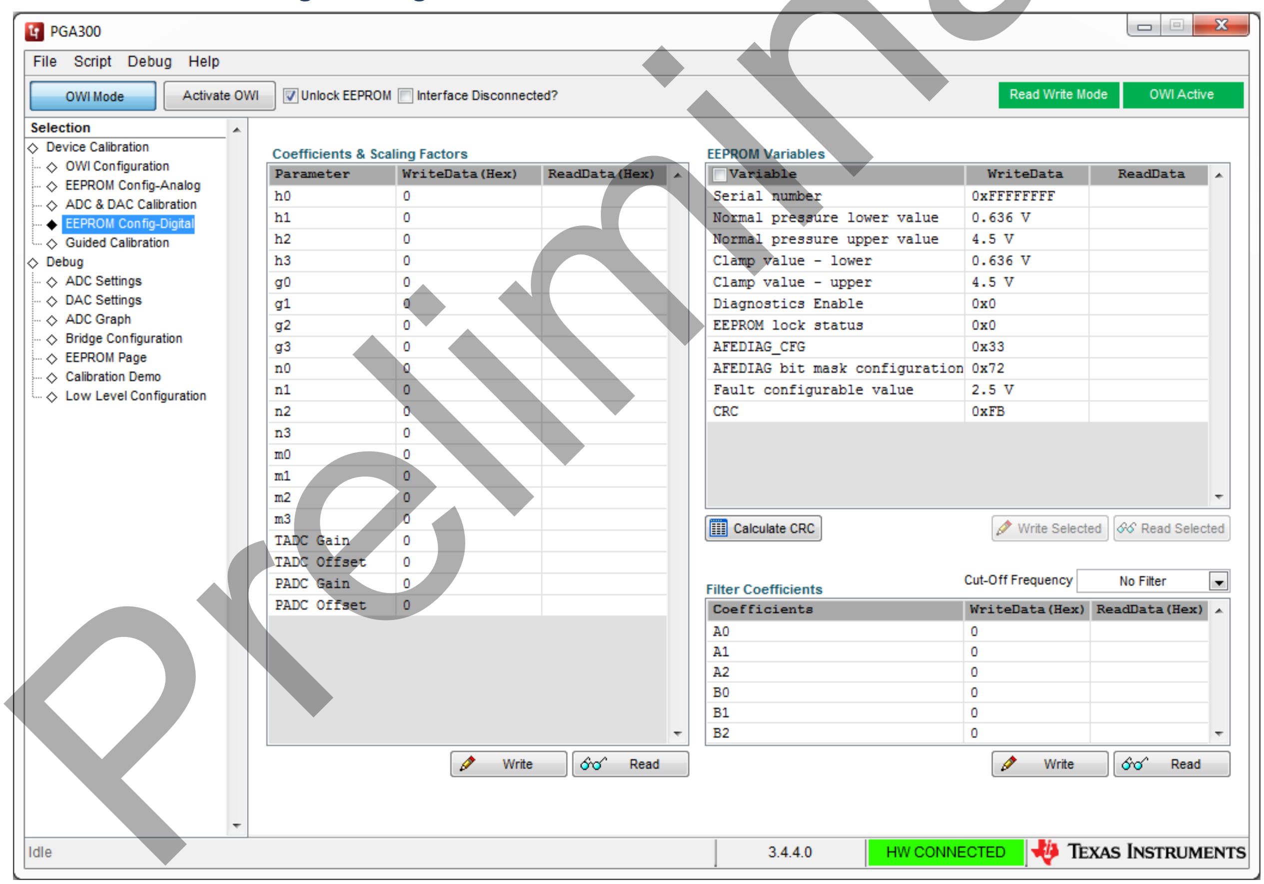


Fig 27: EEPROM Config - Analog Page

- This page allows the user to write specific fields in the EEPROM.
- The GUI provides ability to read the specific fields in the EEPROM.

Preliminary

The user can select the register by clicking on the Register Name. The Input data column when clicked gives the drop down with the appropriate list of values for the respective fields. User can select the values and write it to the EEPROM using **Write Selected** Button.

- Select the Register Name – Cell turns Green
- Click on the Input data column.
- Select the Input Data from the drop down box – Cell turns Yellow indicating the value is not updated in the EEPROM
- **Write Selected** button is enabled
- Click on the **Write Selected** button to write the values into the EEPROM

The values can be read back by selecting the register and clicking the **Read Selected** button. User can use the check box available on the Top left of the table to select/unselect all the registers.

- Select the Register Name to be read – Cell turns Green
- Click on the **Read Selected** button to read the specific register – value gets displayed in the Read Data column

Preliminary

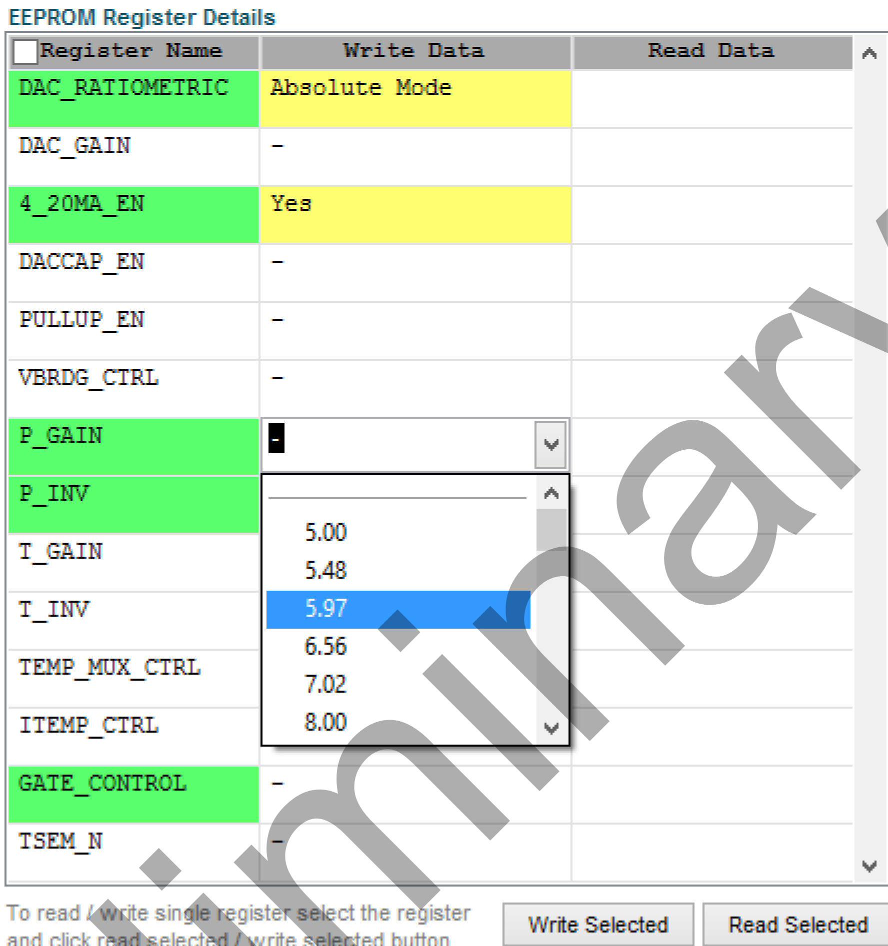


Fig 28: EEPROM Register Selection

Preliminary

Analog PADC Gain

Analog P Gain can be calculated from this page. To calculate Analog PADC Gain,

- Click on T1-P1 and click on **Read ADC** button
- Click on T1-P2 and click on **Read ADC** button
- Select a script file using the browse option. The script file is a python file and the name of the selected file will be displayed below the Calculate PADC Gain button. This file has the algorithm to calculate the P Gain. A python template and a sample script file is provided in C:\Users\Public\Documents\Texas Instruments\PGA300\Configuration Files folder.

- Click on **Calculate PADC Gain** button to calculate the P Gain value. The P Gain value will be updated in the P Gain indicator.
- The status of the calculation will be updated below the description.
- To write the P Gain value to EEPROM and the CSR register, click on the **Write P Gain** button

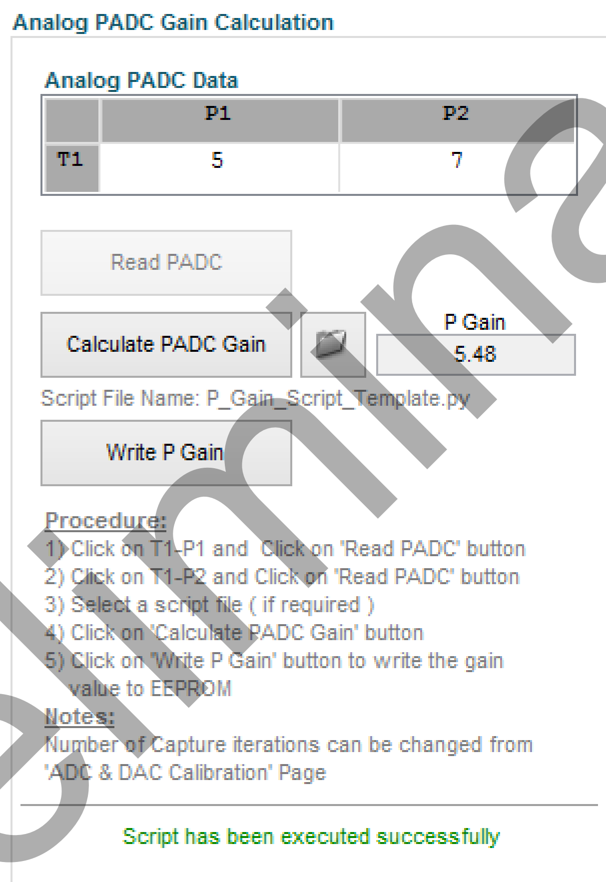


Fig 29: Analog PADC Gain Calculation

2.3.1.3 ADC and DAC Calibration

This page provides options to capture the ADC data, calculate DAC code and calculate the coefficients. Each of the functionality is explained below in detail.

Preliminary

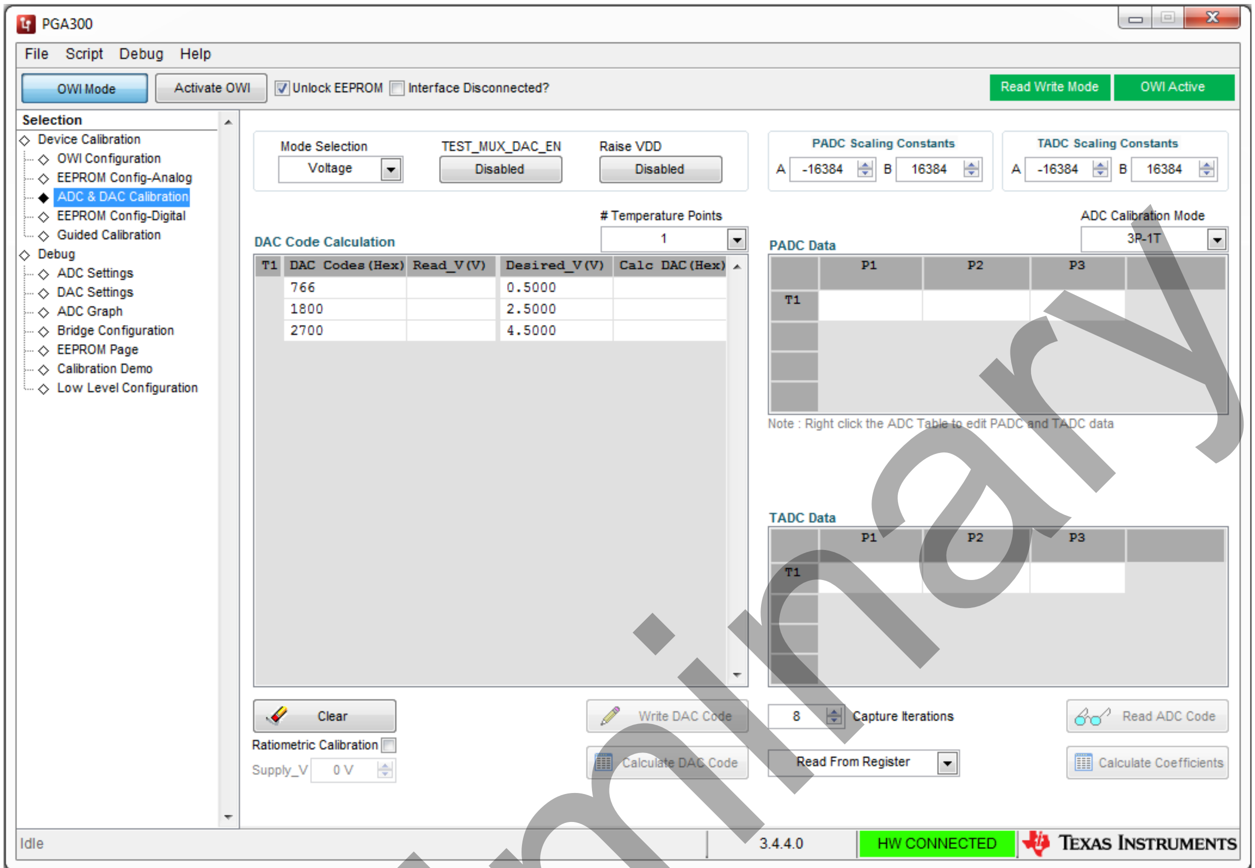


Fig 30: ADC and DAC Calibration Page

- **DAC Code Calculation Table**

The table will be populated from the configuration file based on the following Controls

- Mode selection
- # Temperature points
- ADC Calibration Mode

The **Mode Selection** control is used to switch between **Voltage** and **Current** mode.

Preliminary

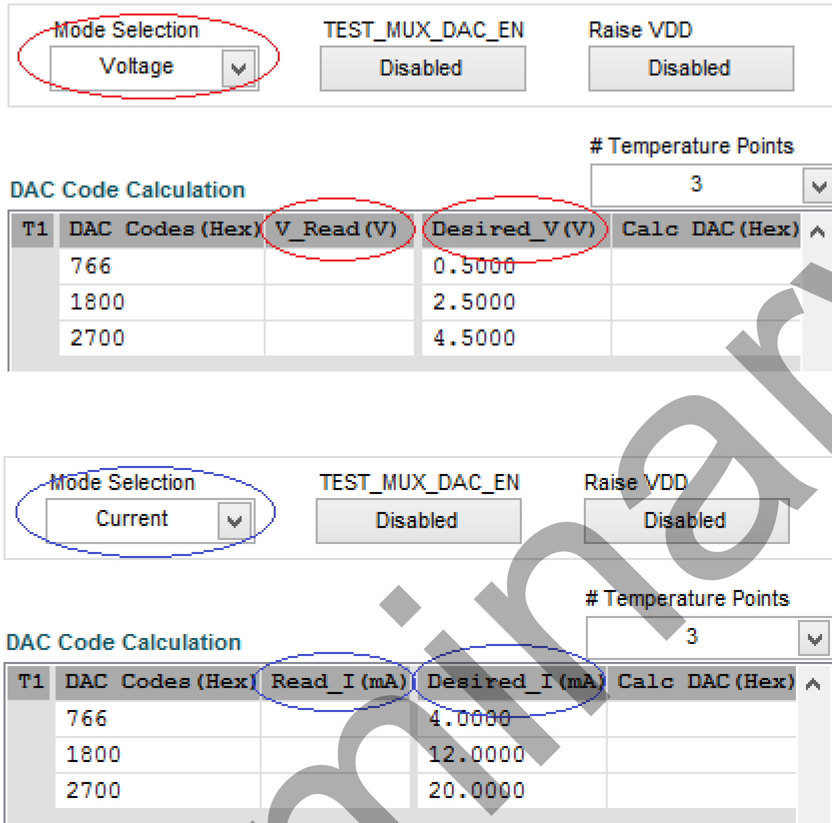


Fig 31: Mode Selection

The **# Temperature Points** can be selected based on the ADC Calibration Mode Control. It should be either 1 or 'n' in **#P-nT** in ADC calibration Mode Control. If the user selects an invalid combination, the GUI displays a POP-UP as shown below.

Preliminary

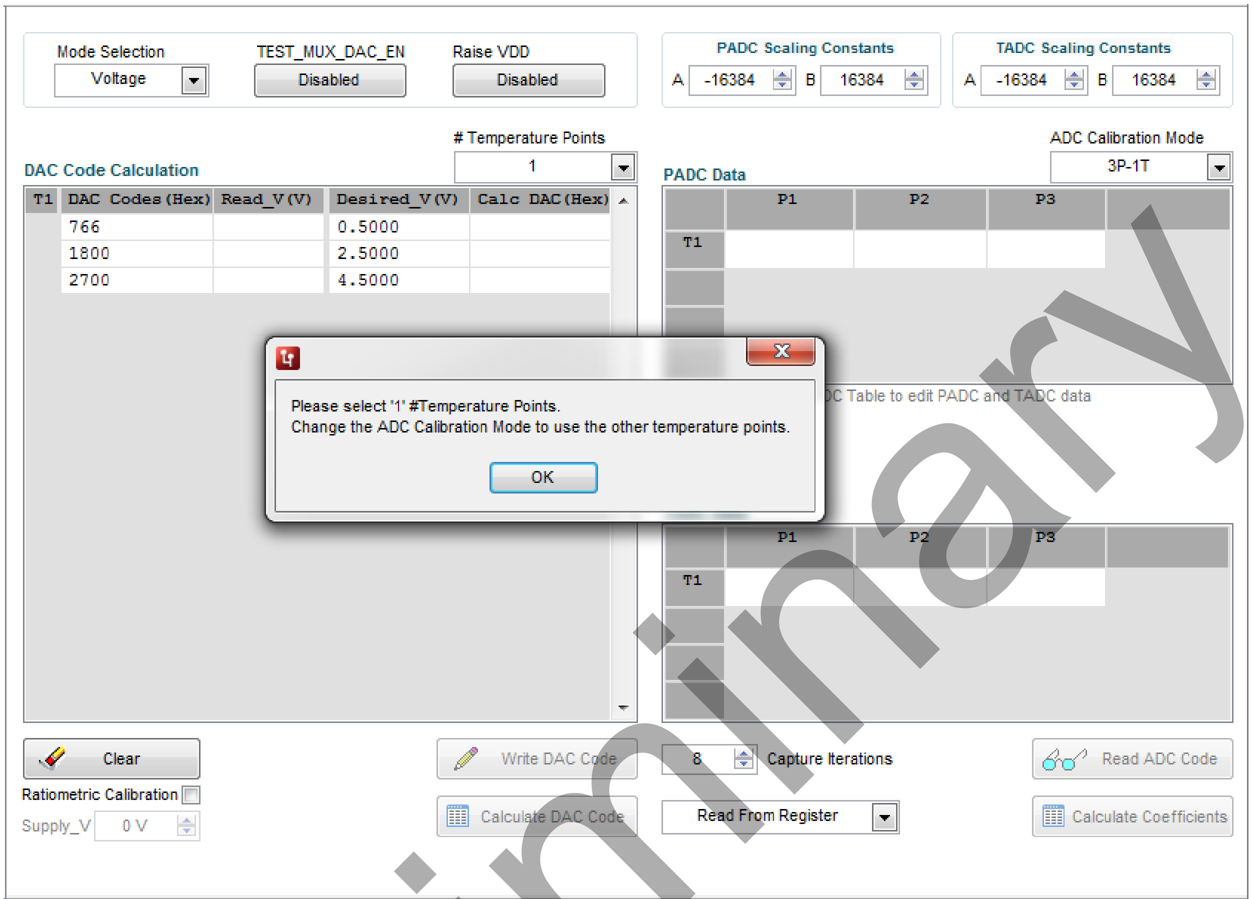


Fig 32: # Temperature Points Warning Pop-Up

Preliminary

When user changes the ADC Calibration mode to a mode that conflicts with #Temperature Points GUI throws a Pop-Up as shown in the below image.

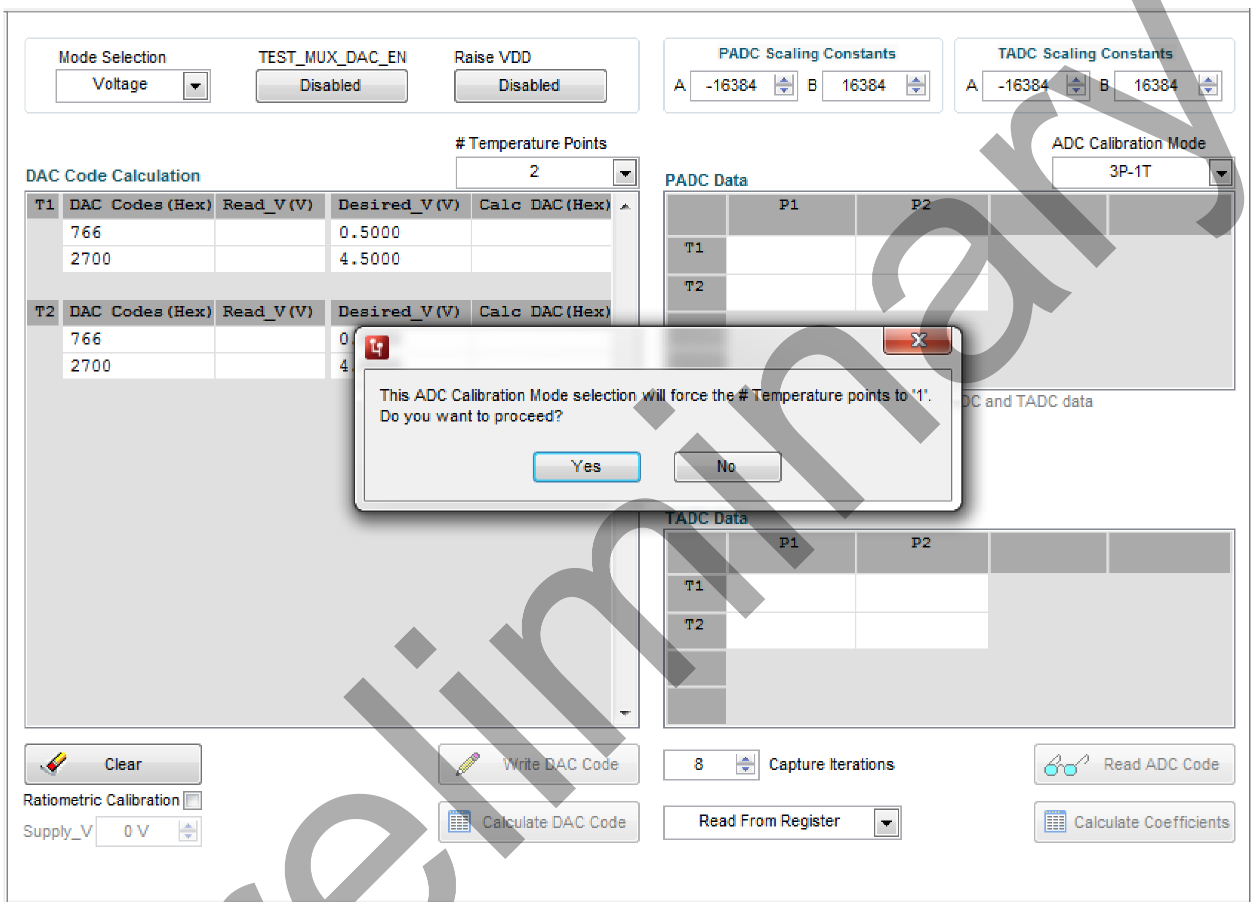


Fig 33: ADC Calibration Mode Warning Pop-Up

Preliminary

- **DAC Code Calculation**

Selecting a DAC Code will highlight the cell in green colour and enable the Write DAC code button as shown in the figure. On clicking the **Write DAC Code** button, the DAC Code is written into the DAC register and **V_Read** cell will be enabled.

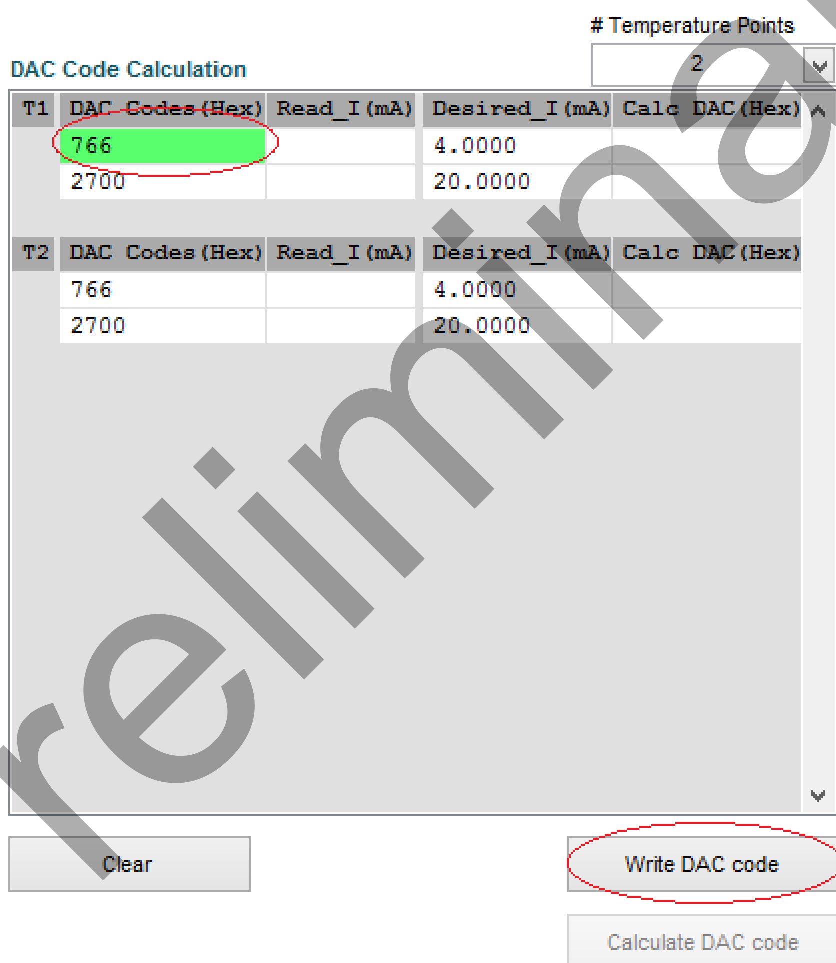


Fig 34: DAC Code Selection

Preliminary

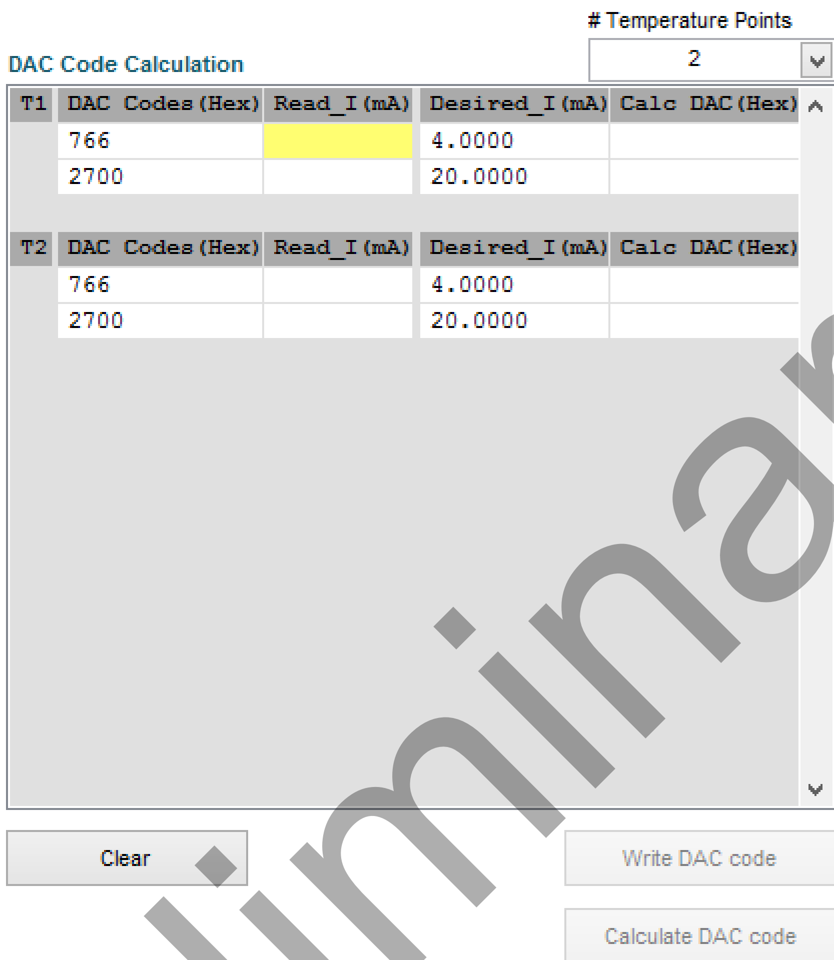


Fig 35: Updating the Read values

The user can measure the voltage and enter it in the cell which is highlighted in yellow color. Once the value is entered, the cell turns back to white color. User can edit the DAC codes by double clicking the DAC Code cell.

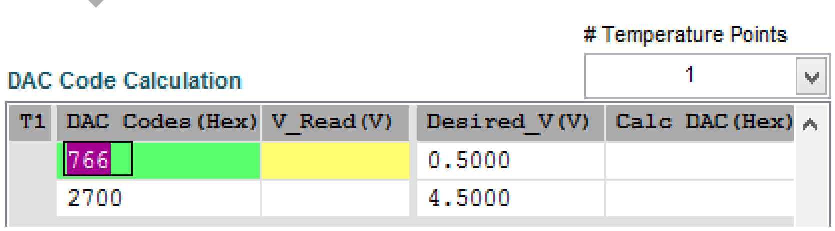


Fig 36: Editing DAC Code

Preliminary

User can also edit the **Desired_V** values by clicking on it.

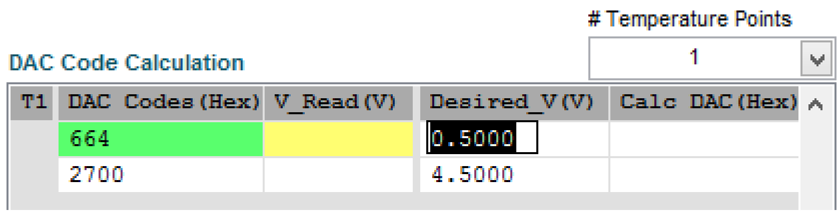


Fig 37: Editing Desired Values

Once all the DAC Codes are written and measured voltages are entered in the corresponding cells, the **Calculate DAC Code** button gets enabled as shown in the below image.

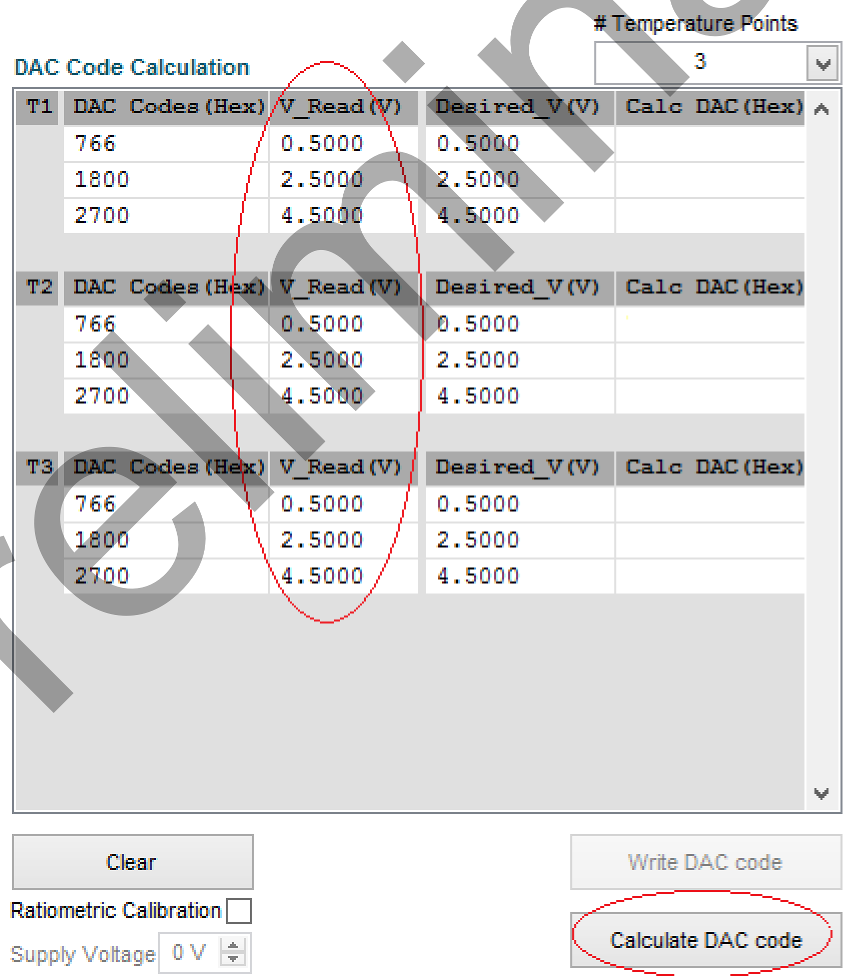


Fig 38: Calculate DAC Code

Preliminary

Click the **Calculate DAC Code** button to calculate the codes and update it in the last column in the DAC Code Calculation table. The DAC codes and desired values available are written back to the configuration file.

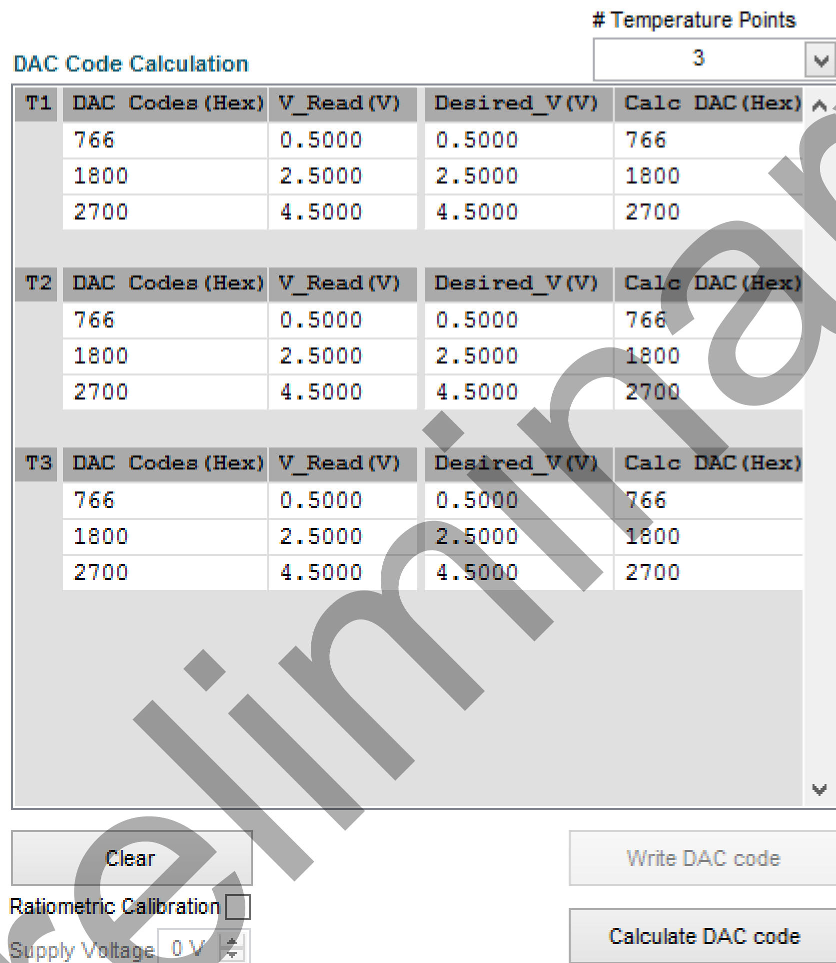


Fig 39: Completed DAC Table

The calculated DAC codes should lies between 0x0000 and 0x3FFF. When the values lies outside of this range, Popup will be shown as shown in the figure *Fig 40: DAC Code Error Popup* and the values lies outside the range will be indicated with red colors.

Preliminary

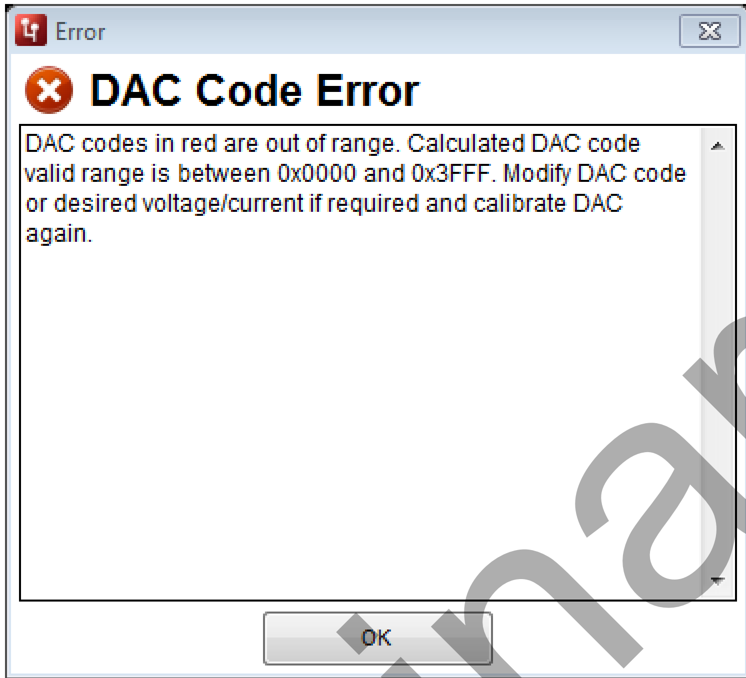


Fig 40: DAC Code Error Popup

In the below figure *Fig 41: DAC Code out of range,* the read value for the DAC code 2700 is 4.5V and so the value calculated for 9.5V exceeds 0x3FFF and it is indicated in red color. This indicates that calculated DAC codes are not proper and it needs to be recalibrated.

Preliminary

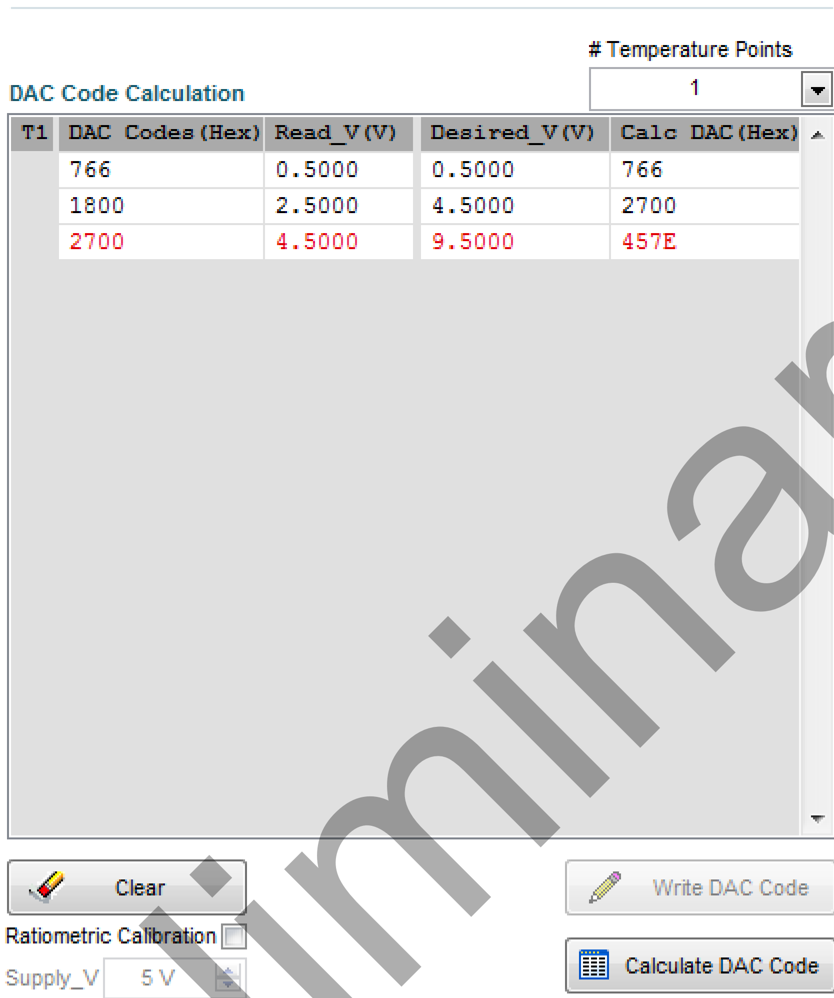
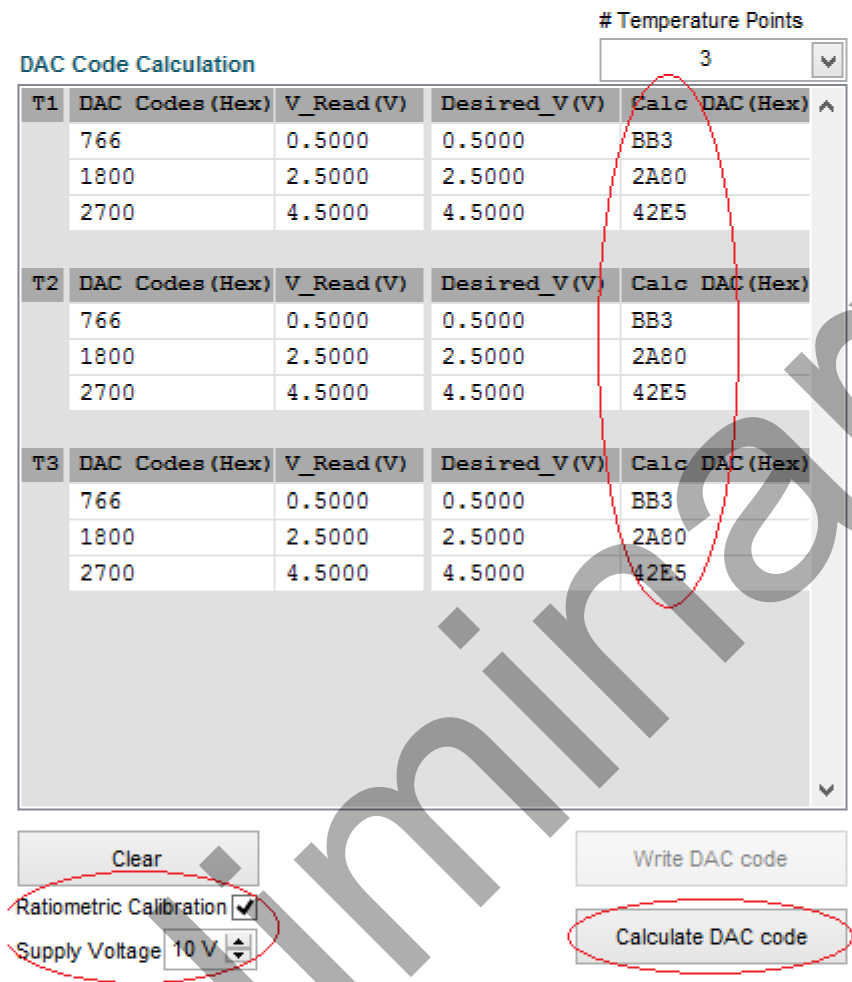


Fig 41: DAC Code out of range

Preliminary

Ratiometric Calibration

To perform ratiometric calibration, Click on **Ratiometric Calibration** check box and specify the supply voltage. Then calculate the DAC codes. The DAC codes will be scaled based on the supply voltage given. The ratiometric calibration is available only in the Voltage mode.



	DAC Codes(Hex)	V_Read(V)	Desired_V(V)	Calc DAC(Hex)
T1				
	766	0.5000	0.5000	BB3
	1800	2.5000	2.5000	2A80
	2700	4.5000	4.5000	42E5
T2				
	766	0.5000	0.5000	BB3
	1800	2.5000	2.5000	2A80
	2700	4.5000	4.5000	42E5
T3				
	766	0.5000	0.5000	BB3
	1800	2.5000	2.5000	2A80
	2700	4.5000	4.5000	42E5

Fig 42: Ratiometric Calibration

Raise VDD

The **Raise VDD** button will make the VDD line High or Low based on the value. If **Raise VDD** is enabled, then VDD line will be raised to high. If **Raise VDD** is disabled, then the VDD line will be pulled low.

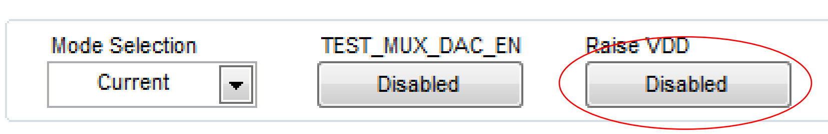


Fig 43: Raise VDD

Preliminary

PADC & TADC Table

Depending on the **ADC Calibration Mode** selected, the PADC and TADC data tables are updated with the corresponding number of elements

The dropdown box below the capture iterations show in specifies whether the ADC values are to be **Read From Register** (PADC_DATA & TADC_DATA) or **Read From FIFO**

Fig 44: ADC Read Mode

Clicking a cell of the ADC Data table highlights it in green color and enables the **Read ADC Code** button and **Capture Iterations** Control.

Preliminary

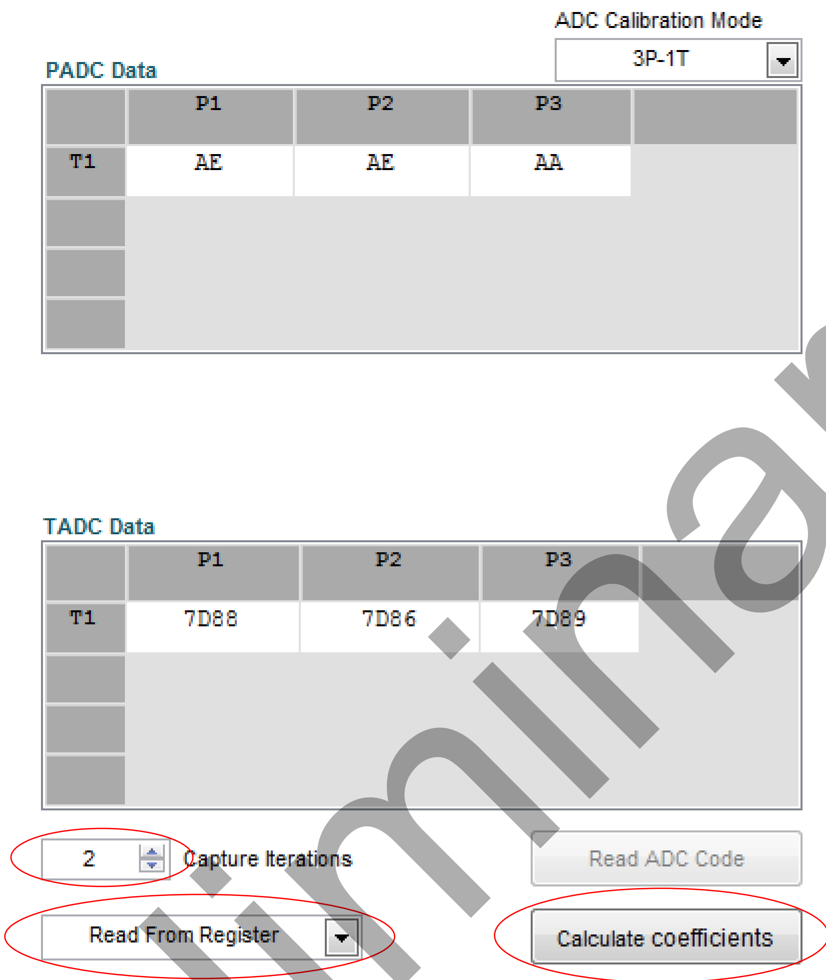


Fig 45: ADC Table Selection

Clicking the **Read ADC Code** button reads the PADC and TADC values and updates it in the corresponding cells.

Preliminary

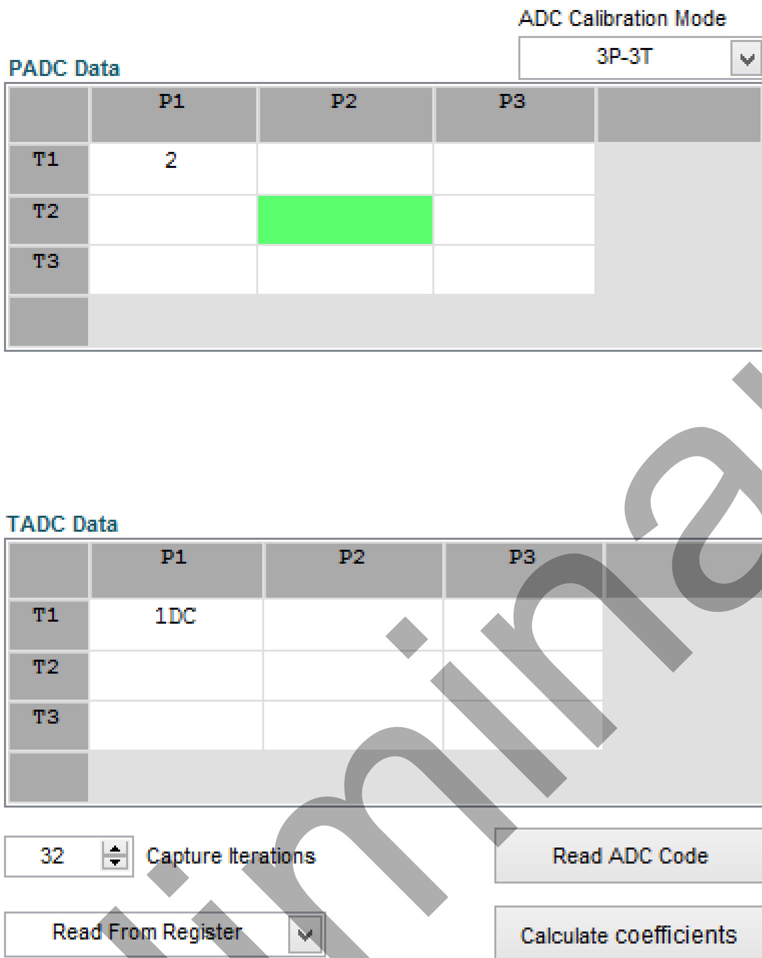


Fig 46: ADC Capture

- **Calculate Coefficients**

Once both the DAC and ADC tables are populated, **Calculate Coefficients** button gets enabled. User can click on the **Calculate Coefficients** button to calculate the coefficients and populate the TC and NL Coefficients table in the **EEPROM Config-Digital** page. The status below the **Calculate Coefficients** button will be updated based on the result of the calculation.

Preliminary

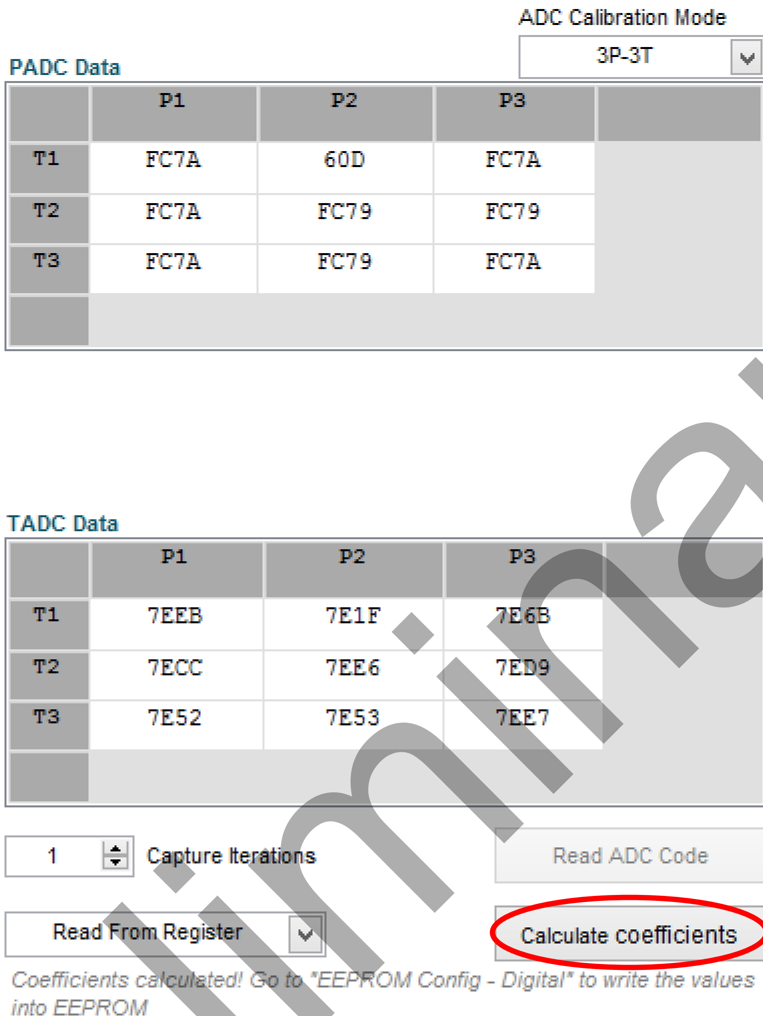


Fig 47: Calculate Coefficients

Preliminary

Coefficient calculations

- The 2nd order TC compensation equation is as follows:

$$\boldsymbol{DAC = (h_0 + h_1TADC + h_2TADC^2) + (g_0 + g_1TADC + g_2TADC^2)PADC + (n_0 + n_1TADC + n_2TADC^2)PADC^2}$$

Where, ***P*** is PADC Value and ***T*** is the TADC value

- The ADC values are normalized as follows.
  - Pn = PADC/2^{14}

- $Tn = TADC/2^{14}$
- $Dn = DAC/2^{14}$

- Based on the normalization, the compensation equation becomes,

$$D_n = \left(h_{0n} + h_{1n}T_n + h_{2n}{T_n}^2\right) + \left(g_{0n} + g_{1n}T_n + g_{2n}{T_n}^2\right)P_n + \left(n_{0n} + n_{1n}T_n + n_{2n}{T_n}^2\right){P_n}^2$$

- Store coefficients in EEPROM by multiplying the floating point version by 2^{14}

  $h_{0EE} = round(h_{0n}*2^{14})$
  $h_{1EE} = round(h_{1n}*2^{14})$
  $h_{2EE} = round\ h_{2n}*2^{14})$
  $g_{0EE} = round(g_{0n}*2^{14})$
  $g_{1EE} = round(g_{1n}*2^{14})$
  $g_{2EE} = round(g_{2n}*2^{14})$
  $n_{0EE} = round(n_{0n}*2^{14})$
  $n_{1EE} = round(n_{1n}*2^{14})$
  $n_{2EE} = round(n_{2n}*2^{14})$

- Note that the normalization should be chosen such that each coefficient is 2 bytes in width in order to fit each coefficient in 2 bytes in the EEPROM.

Preliminary

2.3.1.4 EEPROM Config – Digital

This page allows the user to write the TC & NL Coefficients, Scaling Factors, EEPROM Variables and Filter Coefficients to the EEPROM.

TC & NL coefficients and Scaling Factors

The TC & NL coefficients and scaling factors are populated after the coefficients are calculated in the ADC and DAC Calibration Page. User can edit the TC & NL coefficient and Scaling Factor values in the table.

Coefficients & Scaling Factors

Parameter	WriteData(Hex)	ReadData(Hex)
h0	0	
h1	0	
h2	0	
h3	0	
g0	0	
g1	0	
g2	0	
g3	0	
n0	0	
n1	0	
n2	0	
n3	0	
m0	0	
m1	0	
m2	0	
m3	0	
TADC Gain	0	
TADC Offset	0	
PADC Gain	0	
PADC Offset	0	

Write Read

Fig 48: TC & NL Coefficient Table and Scaling Factors Table

Preliminary

Filter Coefficients

The Filter Coefficients are populated based on the cut-off frequency. The user can edit the coefficients values and Write to EEPROM and Read it back. The edited values are updated in the configuration file.

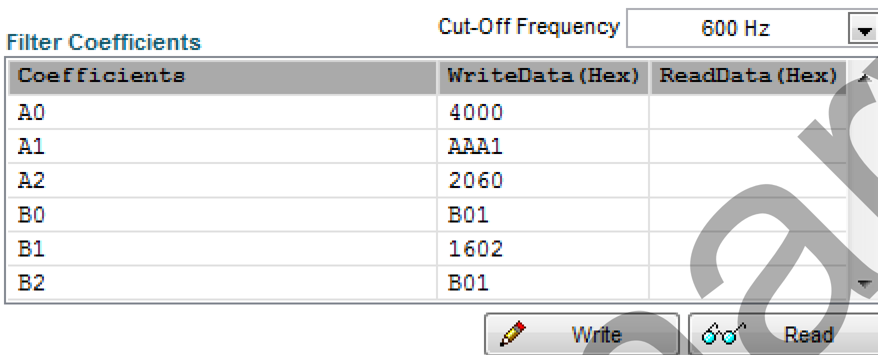


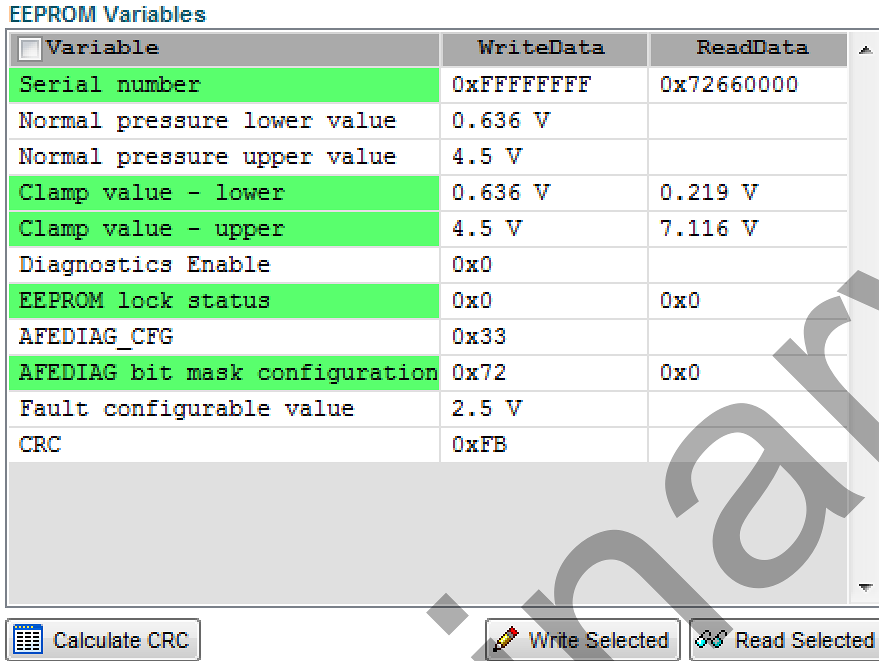
Fig 49: Filter Coefficient Table

EEPROM Variables

EEPROM Variables table lists the available variables that can be configured to EEPROM. Write or read the EEPROM variables,

- Select a Variable from the table. The cell will be highlighted in green.
- Click on **Write Selected** / **Read Selected** buttons to write the variable value into EEPROM or read the variable value from EEPROM.
- To change the value, click on the value in **WriteData(Hex)** column. In the edit box that appears, enter the required value in hexadecimal format.
- The cell will be highlighted in yellow which means that the value is not yet updated in the device.
- Click on **Write selected** to write the value into EEPROM.
- To read/write all the variables, click on the check box at the top left corner. All the registers will be highlighted in green. Now perform **Read Selecte**d or **Write Selected** operation.
- Click on *Calculate CRC* to calculate the CRC. GUI will read the EEPROM data to calculate the CRC value and update it in the *WriteData* column

Preliminary



Variable	WriteData	ReadData
Serial number	0xFFFFFFFF	0x72660000
Normal pressure lower value	0.636 V	
Normal pressure upper value	4.5 V	
Clamp value - lower	0.636 V	0.219 V
Clamp value - upper	4.5 V	7.116 V
Diagnostics Enable	0x0	
EEPROM lock status	0x0	0x0
AFEDIAG_CFG	0x33	
AFEDIAG bit mask configuration	0x72	0x0
Fault configurable value	2.5 V	
CRC	0xFB	

Fig 50: EEPROM Variables Table

2.3.1.5 Guided Calibration

Guided Calibration section helps the user to calibrate the PGA 300 device and verify the calibration through a step by step process.

Instrument Requirements

1. Temperature instrument.
2. Pressure instrument.
3. DMM instruments.

By default the GUI will support following configuration

1. Temperature instrument - User has to change the temperature manually. GUI will instruct the user with the temperature value to be set when necessary.
2. Pressure instrument - User has to change the pressure manually. GUI will instruct the user with the pressure value to be set when necessary.
3. DMM instrument – Used for VDD and DAC measurements. User has to measure manually and update the values in the GUI when necessary.

Preliminary

This section consists of two sub sections

1. Device Calibration.
2. Accuracy Verification test.

The following file is needed for calibrating the device.

1. Calibration Settings File

A. Calibration Settings File

Calibration settings file is a configuration file that contains all the necessary settings for calibrating the device and configuring the device. The GUI comes with some sample settings file which the user can use to calibrate the device without any modifications. There is one sample file for each mode and if the settings are to be changed then user can make a copy of the configuration file and make the necessary changes.

Sample calibration settings file location: *C:\Users\Public\Documents\Texas Instruments\PGA300\Configuration Files\Calibration Setting Files*

i. Overview

The calibration settings file consists of three sections

1. **GUI Settings**
   This section contains the settings to setup the calibration type and other setting that will help to guide the user to set different temperature and pressure points and collect the data at each calibration point. In this section user can also specify if only calibration is to be done or only verification is to be done or do both.
2. **Analog Settings**
   This section contains the settings to be applied to EEPROM of the device for proper calibration of the device.
3. **Digital Settings**
   This section contains the settings to be applied to the EEPROM that will ensure proper functioning of the device after calibration.
4. **Accuracy Verification Test Settings**
   This section contains the settings to verify if the device has been calibrated properly.

Preliminary

ii. Calibration Settings File Editor

GUI has internal editor to edit the calibration settings file as shown in *Fig 51: Calibration Settings File Editor.* It is recommended for the user to use this editor since GUI might not respond properly for any syntax errors in the calibration settings

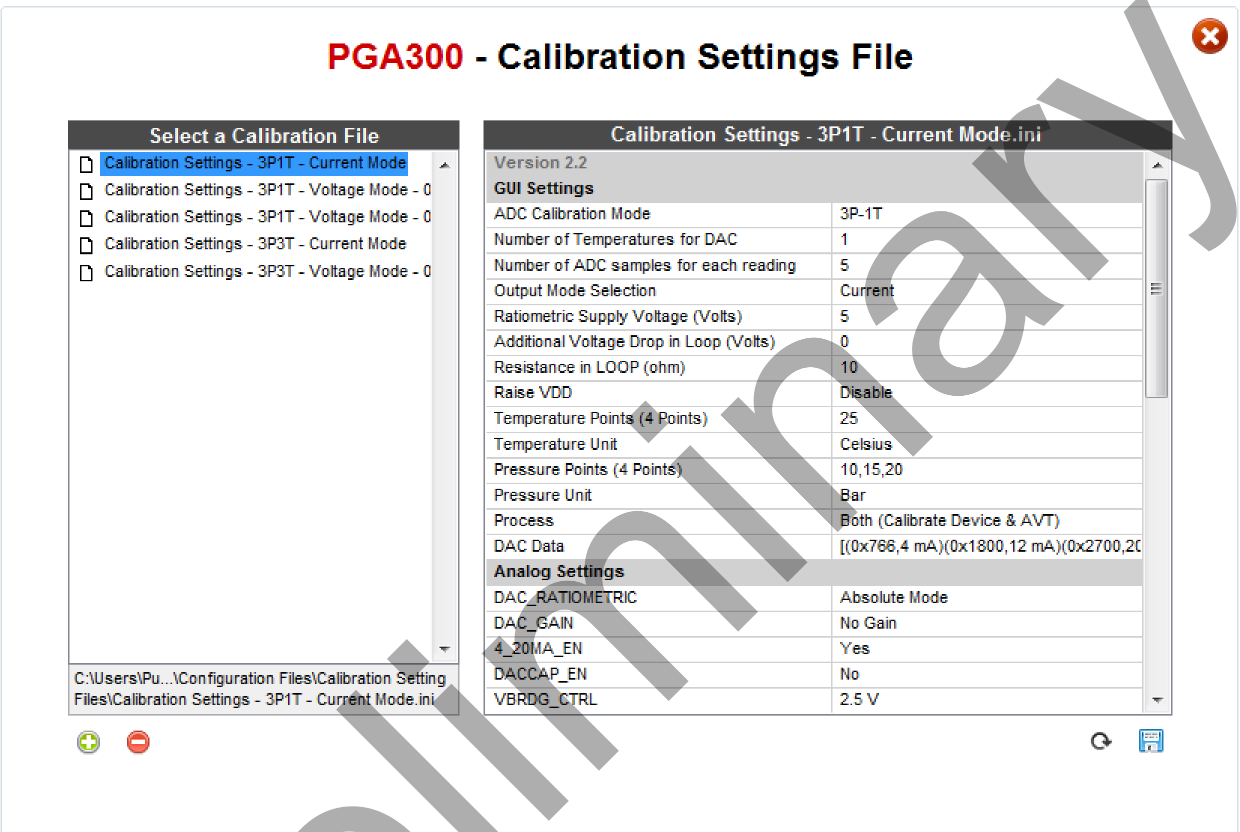


Fig 51: Calibration Settings File Editor

iii. Editing Calibration Settings File

1. Select the file to be loaded on the editor form the files list. For example in the above figure *Fig 51: Calibration Settings File Editor*, Calibration Settings - 3P1T – Current Mode.ini is selected form the list. The selected file name will be shown on the editor section and the full path will be shown below the selection list.
2. To add a new settings file or remove an existing file, *Add* button or *Remove* button can be used as highlighted in *Fig 52: Add or Remove Calibration Settings File*

Preliminary

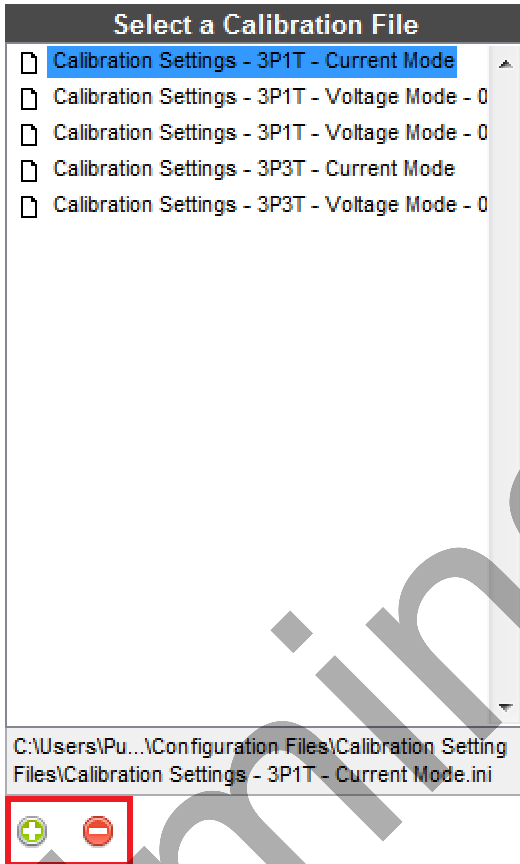


Fig 52: Add or Remove Calibration Settings File

3. Select the settings to be edited and click on the setting value to be changed. A drop down list will appear showing the list of available values that can be configured. Select required value from the drop down as shown in *Fig 53: Calibration Settings File Editor Drop Down*

Preliminary

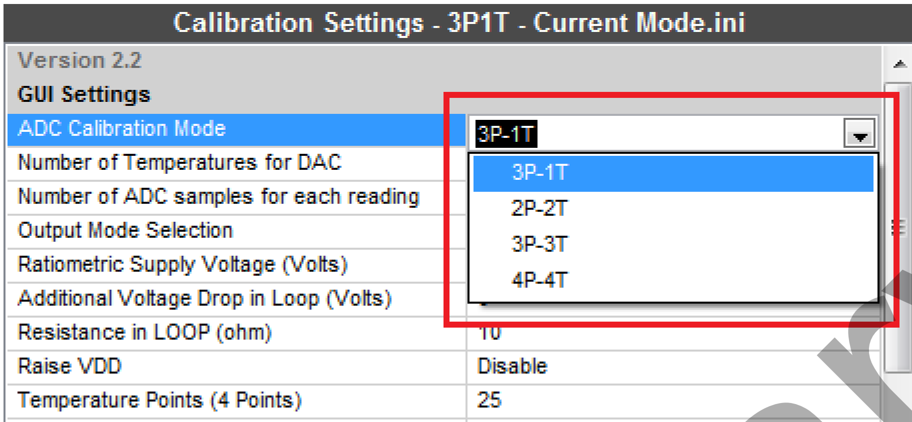


Fig 53: Calibration Settings File Editor Drop Down

For some of the settings, drop down will not appear instead text entry box will appear where the user can type the values. One such example is Pressure Points (4 Points) Settings where the user has to type the pressure values separated by comma as shown in *Fig 54: Calibration Settings File Editor Text Box*

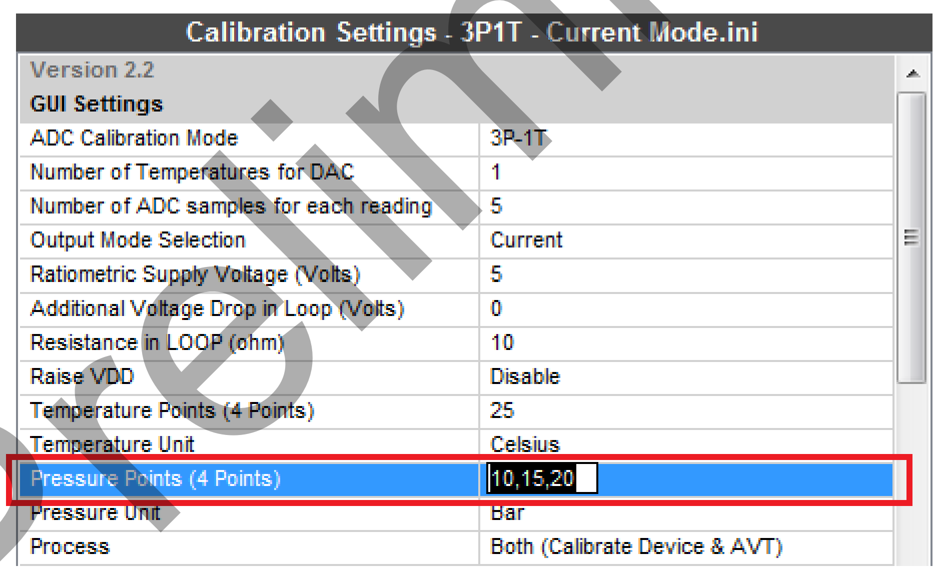


Fig 54: Calibration Settings File Editor Text Box

For few settings, a popup will appear which will help the user to change the settings and give the values. One such example is example DAC Data values. DAC data values will contain the DAC codes and desired DAC output corresponding to the DAC code. To

Preliminary

change the values, click on the DAC Data value, a popup will appear as shown in *Fig 55: Calibration Settings File Editor DAC Data.* On the popup, click on any of the value. A text entry box will appear where the user can edit the Data.

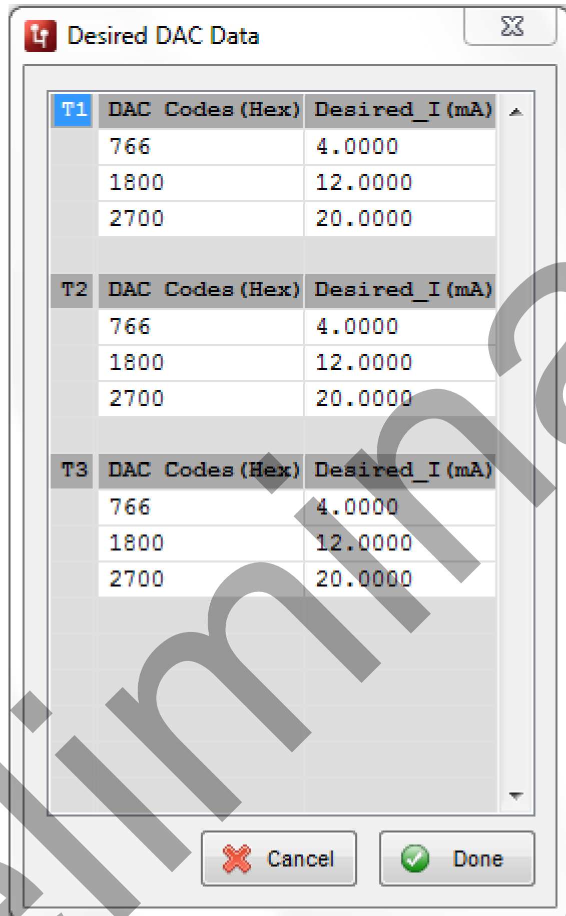
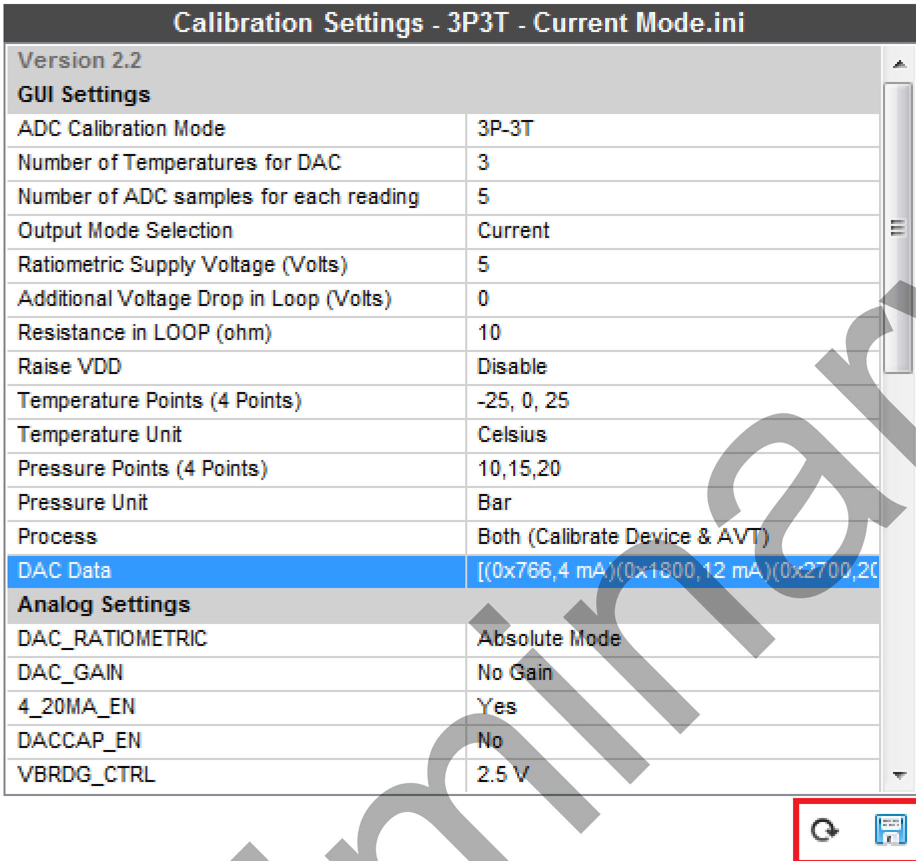


Fig 55: Calibration Settings File Editor DAC Data

4. Once the required settings have been edited, click on *save* option below the editor. To discard the changes and reload the original file, click on reload option below the editor. A confirmation will be asked when reloading the file. The reload and save option are highlighted in the figure *Fig 56: Calibration Settings File Save and Load*

Preliminary



Calibration Settings - 3P3T - Current Mode.ini	
Version 2.2	
GUI Settings	
ADC Calibration Mode	3P-3T
Number of Temperatures for DAC	3
Number of ADC samples for each reading	5
Output Mode Selection	Current
Ratiometric Supply Voltage (Volts)	5
Additional Voltage Drop in Loop (Volts)	0
Resistance in LOOP (ohm)	10
Raise VDD	Disable
Temperature Points (4 Points)	-25, 0, 25
Temperature Unit	Celsius
Pressure Points (4 Points)	10,15,20
Pressure Unit	Bar
Process	Both (Calibrate Device & AVT)
DAC Data	[(0x766,4 mA)(0x1800,12 mA)(0x2700,20
Analog Settings	
DAC_RATIOMETRIC	Absolute Mode
DAC_GAIN	No Gain
4_20MA_EN	Yes
DACCAP_EN	No
VBRDG_CTRL	2.5 V

Fig 56: Calibration Settings File Save and Load

B. *Device Calibration*

These sequence of steps will guide the user to calibrate a PGA300 device either in current mode or voltage mode.

i. *Calibration Procedure*

1. Go to Guided Calibration page. A page as shown in below figure will appear. Click on *Start»* button to start the calibration

Preliminary

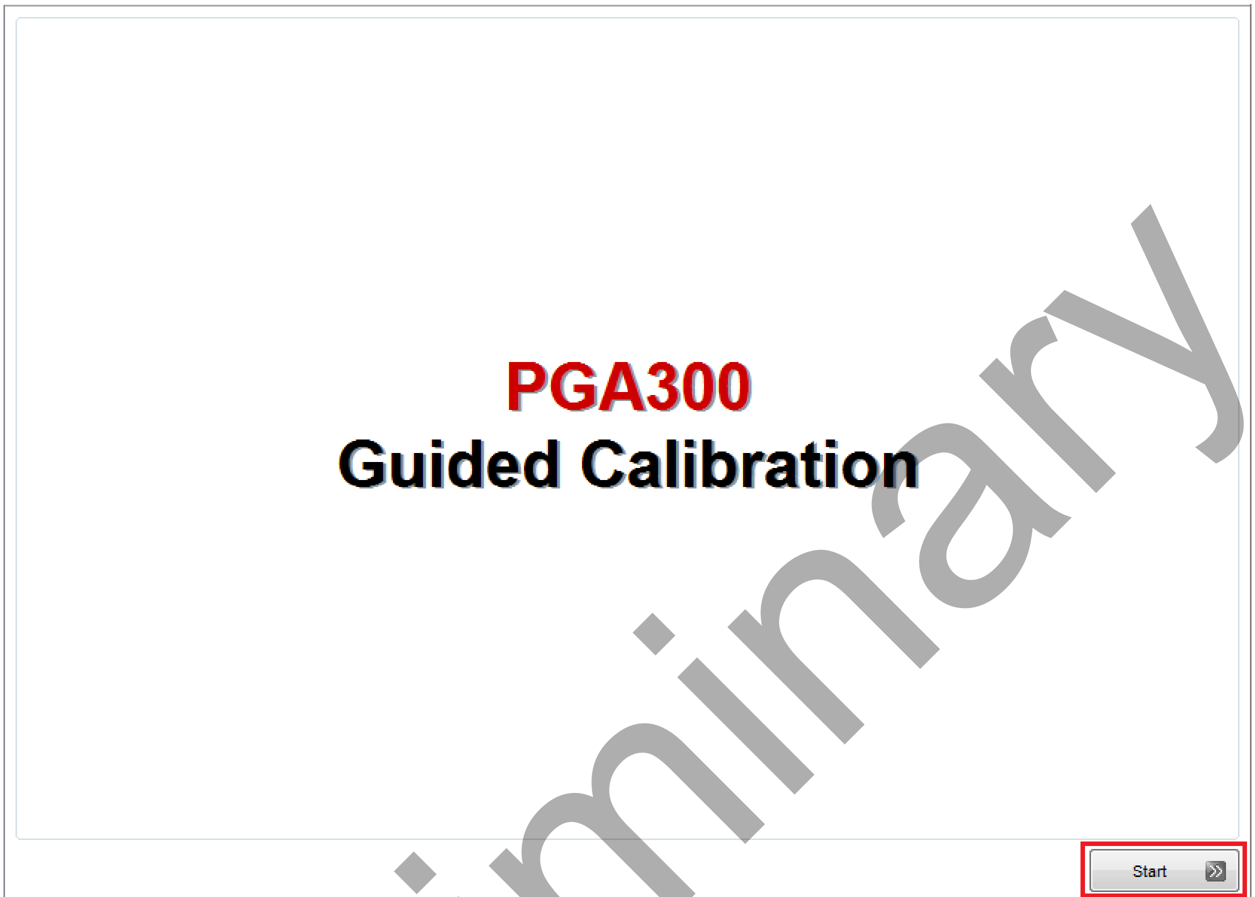


Fig 57: Guided Calibration - Start

2. The next page is the calibration file editor. Detailed information on this page has been described in *Calibration Settings File* section on *Page 59.* After making the required changes click on *Next»* button
3. The next page is POT configuration page. The default values will be loaded from the calibration settings file loaded at the start of the calibration. If there are any changes to the loaded values, the user can change the values in the corresponding edit box and click on *configure* button to apply the values to the device.

Preliminary

Fig 58: POT Configuration

4. The next page sends OWI activation pulse to the connected PGA300 device when *Activate OWI* button is pressed. After sending activation pulse the GUI will read the microcontroller status and update it in the GUI as shown in the figure below

Preliminary

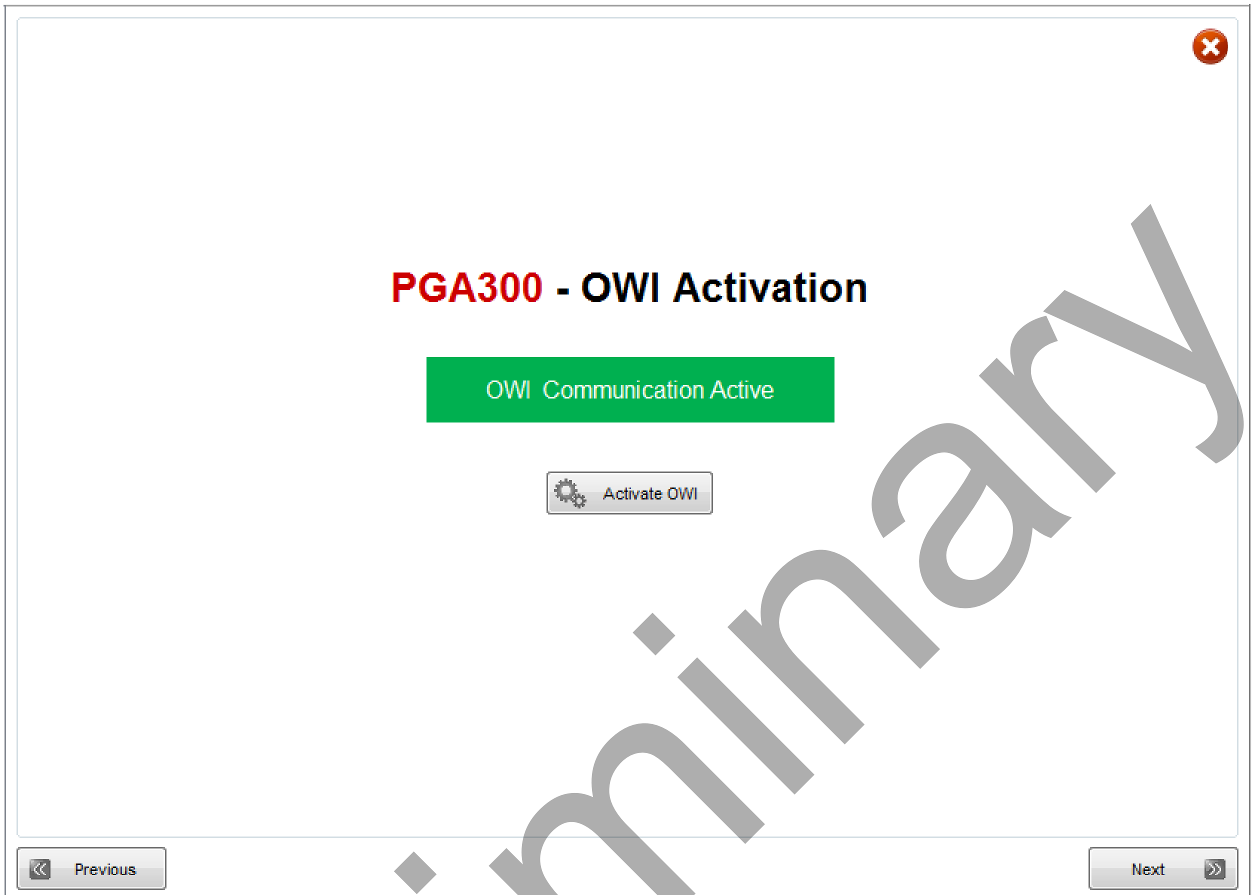


Fig 59: OWI Activation

Notes:

The OWI activation is done using USB2ANY scripting.

5. The next page will configure the PGA300 device for calibration. As soon as the user enters this page, GUI will start to prepare PGA300 device. The progress of the operation will be shown at the bottom of the page. The below figure *Fig 60: Guided Calibration Initial Configuration* shows the operations that are taking place in this step

Preliminary

Fig 60: Guided Calibration Initial Configuration

Notes:

In this step, settings from the *Analog Settings* section of *Calibration Settings* file will be applied to the PGA300 device by writing the data to EEPROM and Device registers. EEPROM Data is sent as a script containing the sequence of register writes with register address and data to write. The data to be written to the EEPPROM will depend on the previous EEPROM data in the device. After script has completed the *Analog Settings* will be written to the device CSR registers.

6. Depending on if the PGAIN value is to be *calibrated* or a *constant value* is to be written, the next page will either be *Analog Calibration* page or start with data collection. If analog PGAIN is to be calibrated, then Analog calibration page will be appearing.

Preliminary

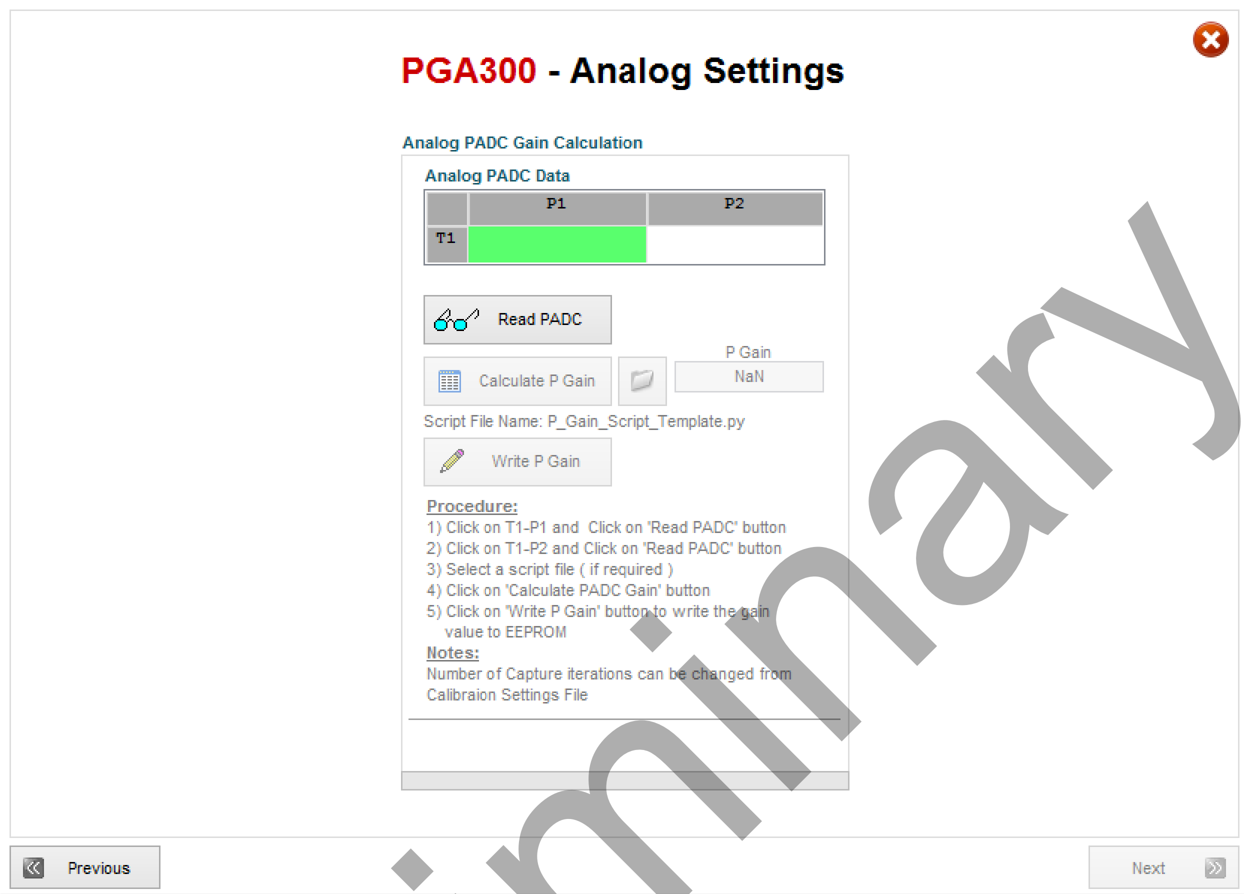


Fig 61: Guided Calibration Analog Settings

a. Click on *T1-P1* cell. The cell will turn into green
b. Click on *Read PADC* to read PADC Data from the device.
c. Click on *T1-P2* cell. The cell will turn into green
d. Click on *Read PADC* to read the PADC data from the device
e. If the algorithm has to be changed then select the browse button and select a Python file.
f. Then click on *Calculate P Gain* button
g. After calculating P Gain, click on *Write P Gain* button. This will write P Gain value to EEPROM and CSR register in the PGA300 Device.

7. From the next page Output data collection will start. This section consists of
   a. VDD Data Collection
   b. DAC Data Collection

Preliminary

c. ADC Data collection

Note:

VDD data collection will be done only if calibration is in voltage mode and ratiometric.

Data collection consists of following steps

a. Temperature Setting
This page will guide the user to set proper temperature for calibration. Set the temperature manually and click on *OK* button.

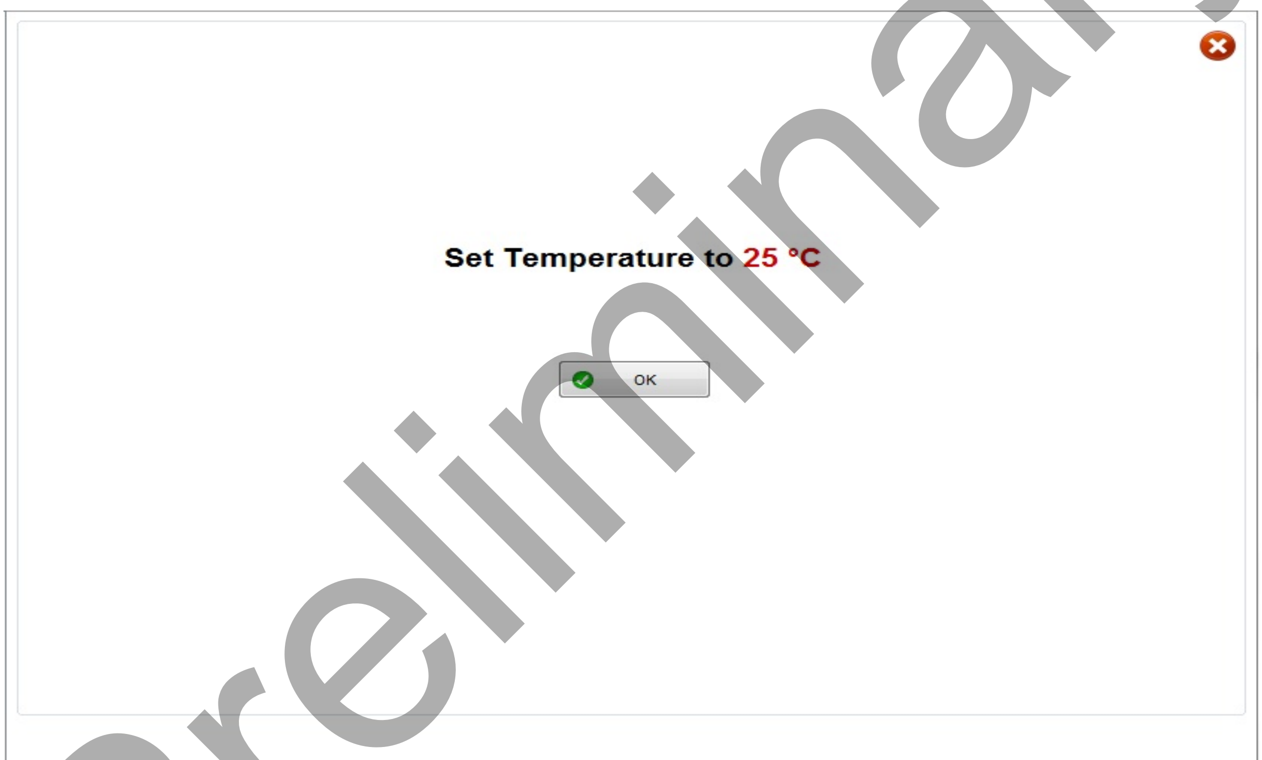


Fig 62: Guided Calibration Temperature Setting

a. Pressure Setting
This page will guide the user to set proper pressure for calibration. Set the pressure manually and click on *OK* button.

Preliminary

Fig 63: Guided Calibration Pressure Setting

b. VDD Voltage Measurement
VDD Voltage measurement will be done only if the calibration is in voltage mode and ratio metric. User needs to measure the VDD voltage manually from the device and enter it in the numeric control. Next button will be disabled until the user enters VDD data in the numeric control.

Preliminary

Fig 64: Guided Calibration VDD Data Collection

Preliminary

c. DAC Data Measurement
   This page sets a particular DAC value to the device and waits for the user to enter the measured voltage/current. The DAC data to be set will be taken form the *Calibration Settings File.*
   a. User can change the DAC code in the numeric control if required.
   b. Click on *Write* button. This will write the DAC data to the device.
   c. Measure and enter the voltage or current form the device depends on the mode configured in the *Calibration Settings File*.

Fig 65: Guided Calibration DAC Data Collection

d. ADC Data Measurement

This page reads PADC and TADC data form the Device and displays the data in the table.

PADC and TADC data can be read from register or from FIFO based on the selection on *ADC & DAC Calibration* page. Click on the *Read ADC* button to read the PADC and TADC data.

Note:

ADC data read from FIFO are done using scripting.

Preliminary

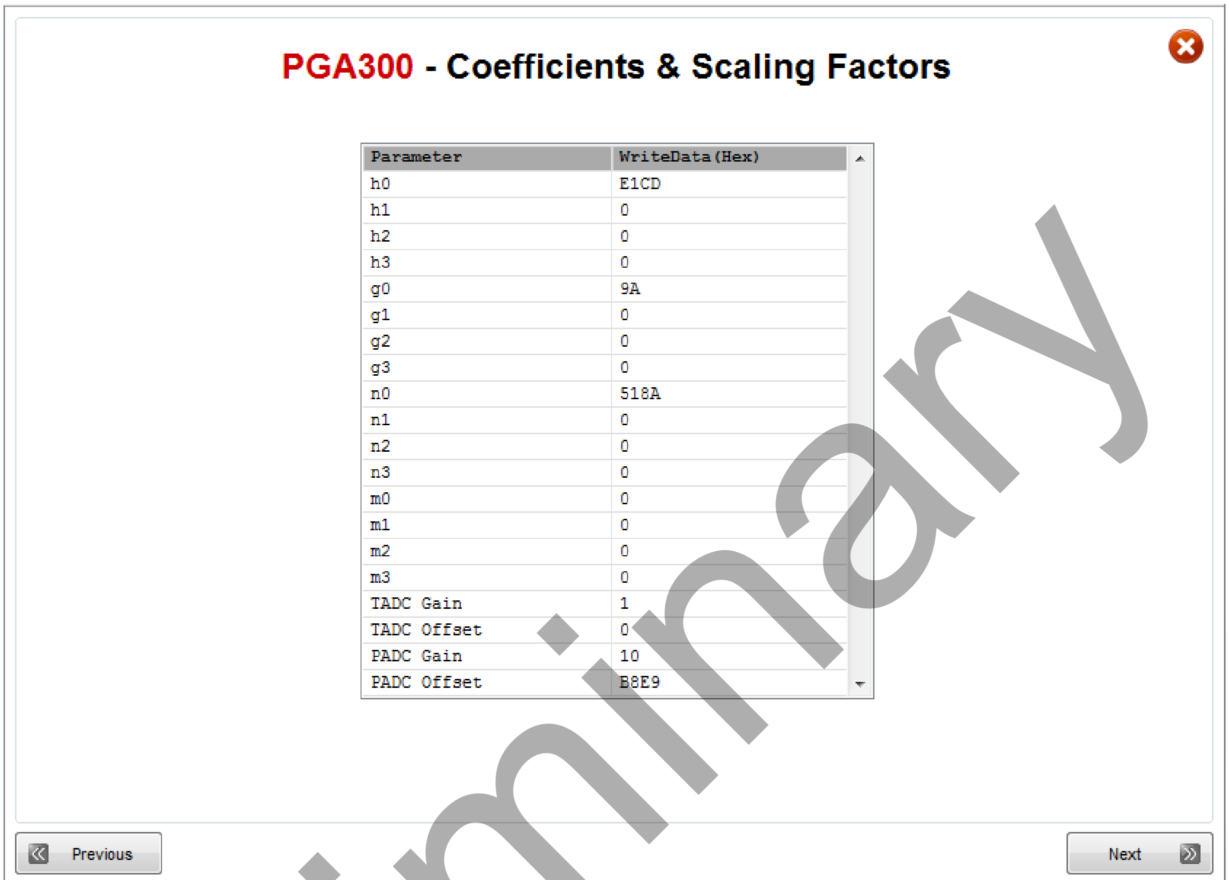
Fig 66: Guided Calibration ADC Data Collection

Preliminary

The above four steps will be repeated till all the data required for calibration is collected.

8. After all the data required for calibration is collected following parameters will be calculated
    a. TC and NL Coefficients
    b. TADC Gain
    c. PADC Gain
    d. PADC Offset
    e. TADC Offset

All these parameters will be calculated for the device and displayed on the table.



PGA300 - Coefficients & Scaling Factors

Parameter	WriteData(Hex)
h0	E1CD
h1	0
h2	0
h3	0
g0	9A
g1	0
g2	0
g3	0
n0	518A
n1	0
n2	0
n3	0
m0	0
m1	0
m2	0
m3	0
TADC Gain	1
TADC Offset	0
PADC Gain	10
PADC Offset	B8E9

Fig 67: Guided Calibration Summary

The calculated DAC codes should lies between 0x0000 and 0x3FFF. When the values lies outside of this range, Popup will be shown as shown in the figure *Fig 40: DAC Code Error Popup* and window will be opened for correcting the DAC codes if required and then click on *Recalibrate* button to start the data collection process with the corrected data or close the window to discard the changes and start data collection with old data.

Preliminary

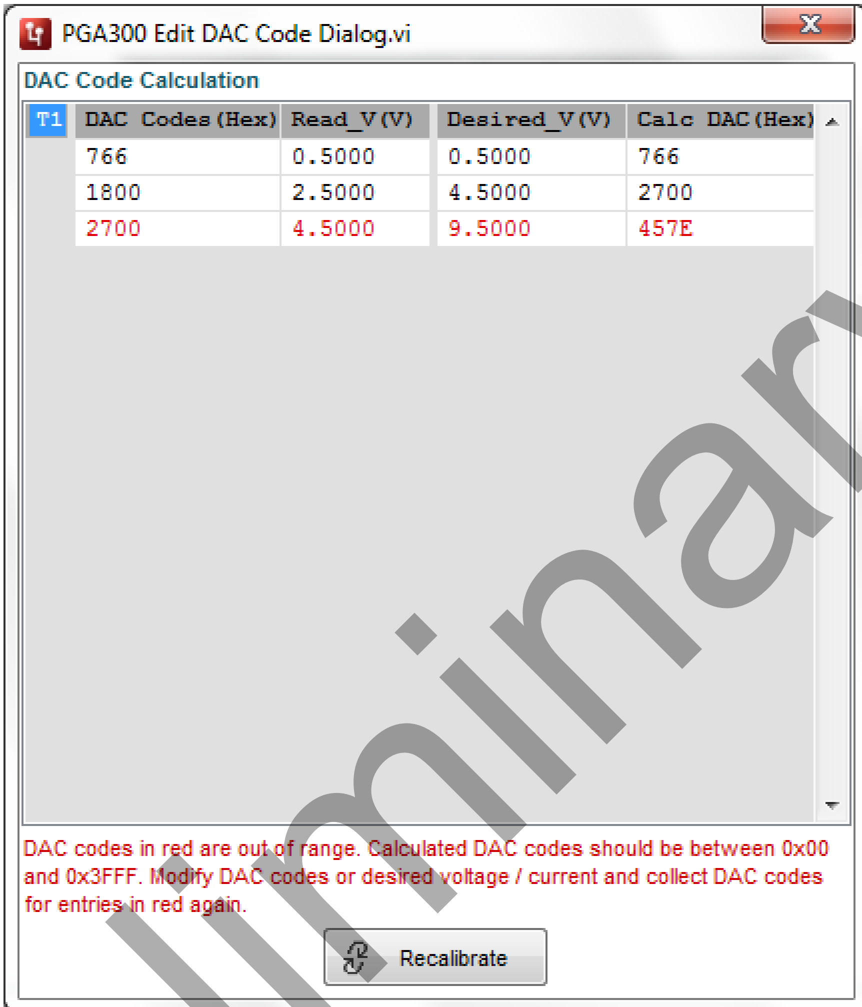


Fig 68 Edit DAC Code Dialog

9. After Calibration summary page, Digital Calibration starts. Following are the parameters that can be configured
   a. Serial Number
   b. Normal Pressure Values
   c. Clamp Values
   d. Fault Values
   e. EEPROM Lock
   f. Diagnostics configuration

Preliminary

All the parameters will be loaded from *Calibration Settings* file. User can change these values if required. After changing the values, when going to next or previous page the GUI will ask if the values are to be updated to the *Calibration Settings.* Based on the user selection the settings will be saved or not saved.

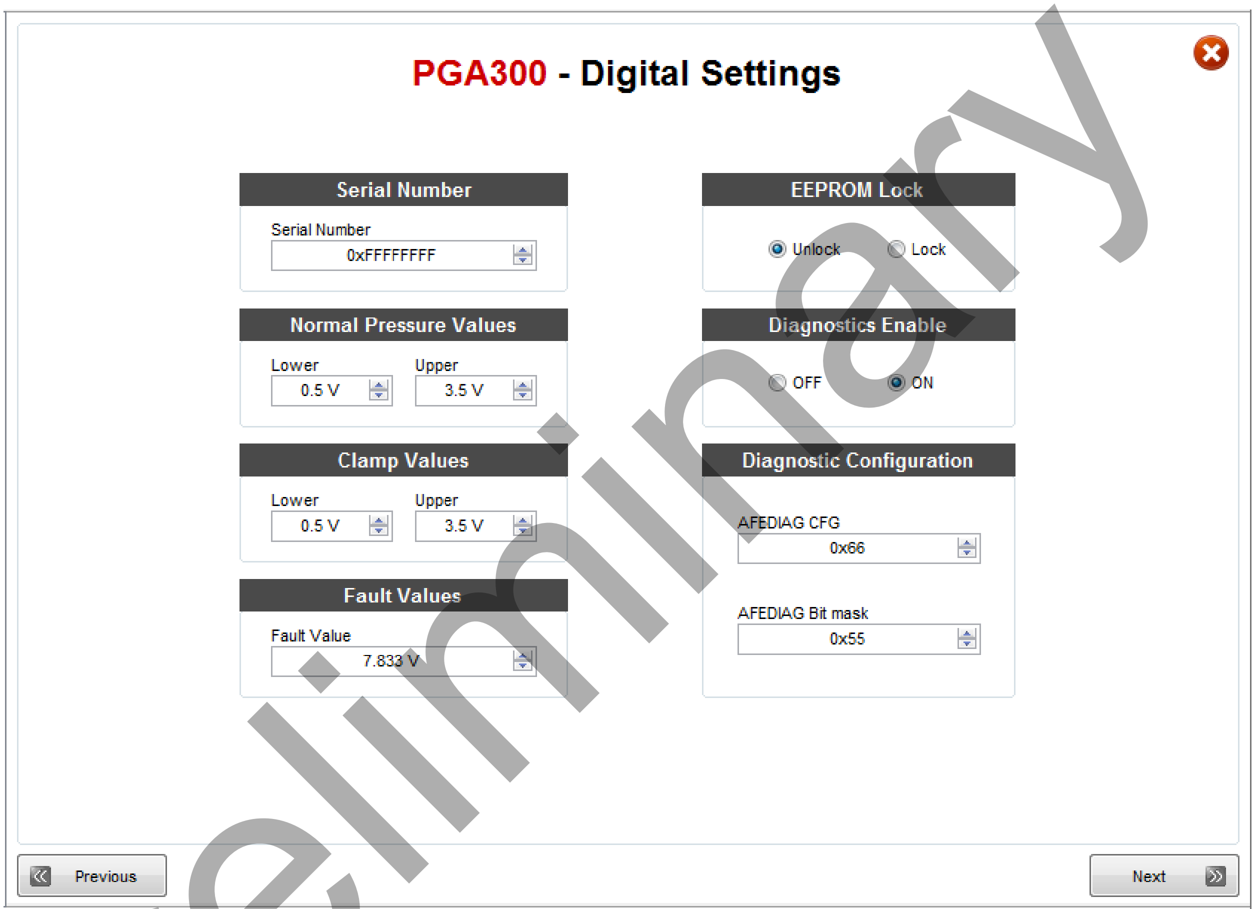


Fig 69: Guided Calibration Digital Settings

10. When user enters the next page the, the GUI will write the calibration coefficients and digital settings to the device. The calibration coefficients and digital settings will be written to specific EEPROM locations. The progress of the step will be shown in the progress bar at the bottom of the page. Once the calibration is completed, the GUI will show PGA300 calibrated successfully… as shown in figure *Fig 70: PGA300 Calibration Successful.*

Preliminary

Fig 70: PGA300 Calibration Successful

After this step the user can start calibration for the device.

C. Accuracy Verification Test

After Calibration has been finished to test whether the device has been calibrated properly, accuracy verification test (AVT) can be done. Accuracy verification test will be done when microcontroller is running on the PGA300 Device. This section involves setting a temperature and pressure point combination and verifying if the device outputs proper DAC value. This test can be started following the calibration process or separately. If this section is invoked without doing a calibration, then *POT Configuration* step will be performed and after which Accuracy Verification Test will start

1. This section starts with a welcome page as shown in figure *Fig 71: Accuracy Verification Test - Welcome Page* below

Preliminary

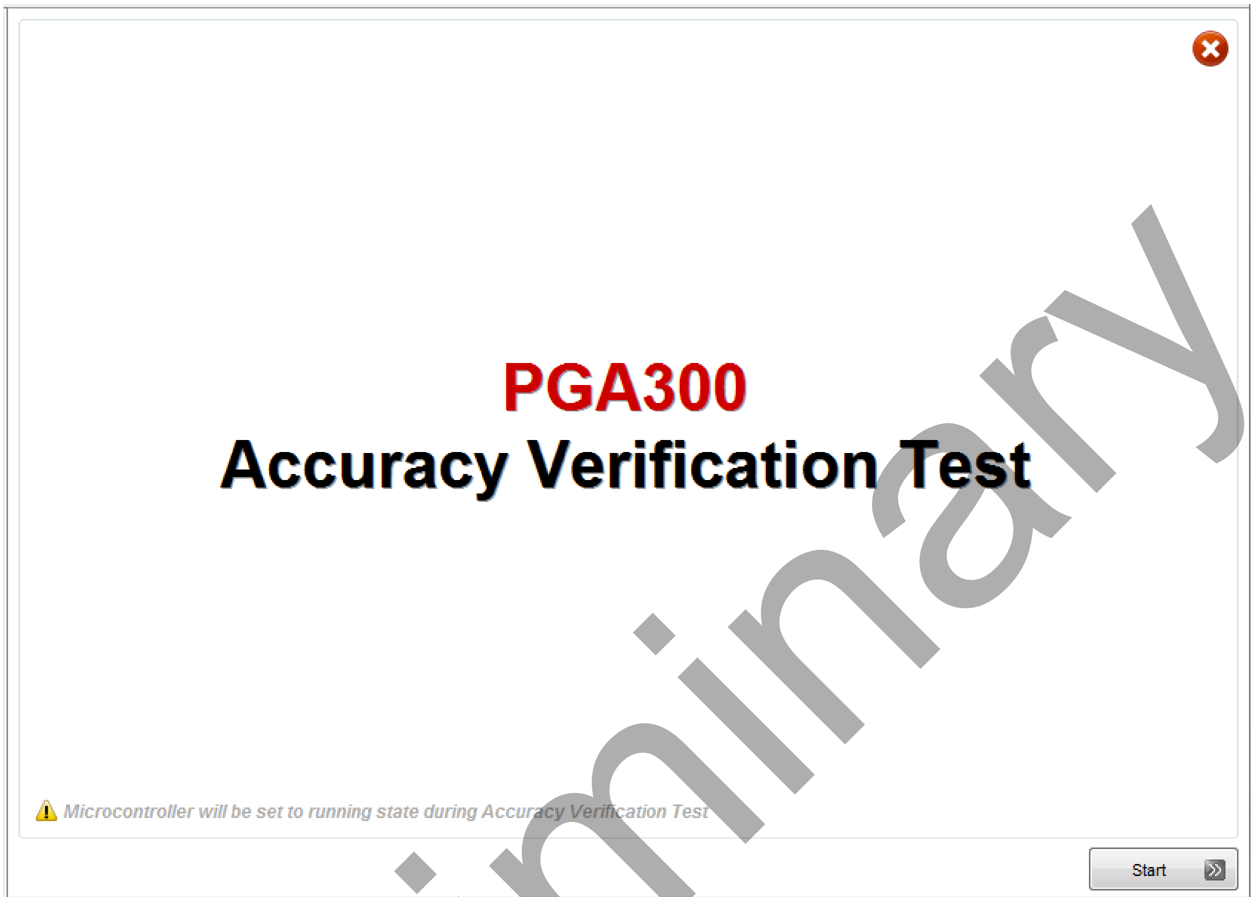


Fig 71: Accuracy Verification Test - Welcome Page

2. The next page shows the preview of the settings loaded form the selected Calibration settings file. The user can change the settings if required and proceed with the test. Here user have the options to add or remove test points using buttons or using right click menu options. Reload button can be used to load the settings from calibration setting file.

Preliminary

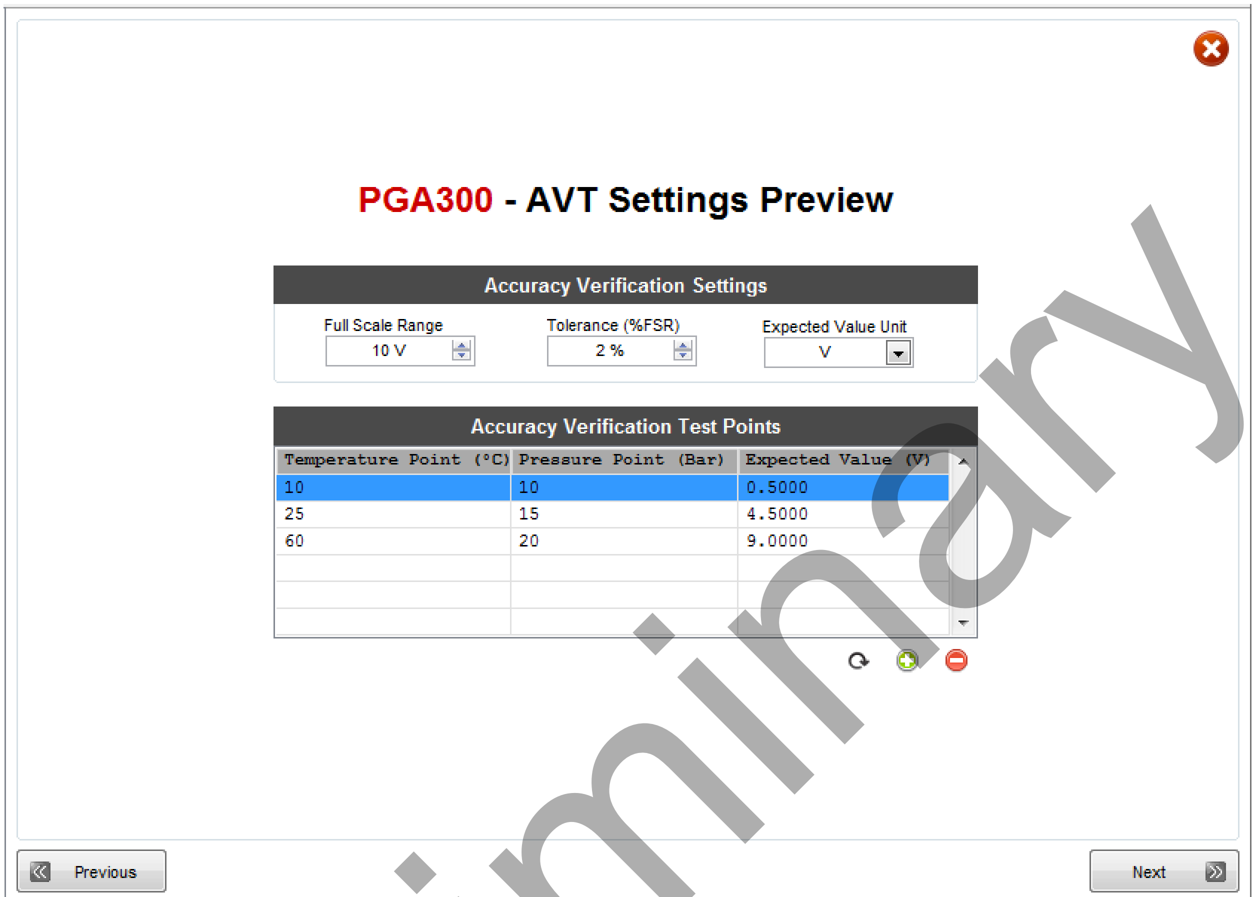


Fig 72: Accuracy Verification Test- Settings Preview

3. The next step is data collection for Accuracy Verification Test. This section consists of following steps
   a. Temperature Setting
      This page guide the user to set the proper temperature for accuracy verification test. Set the temperature manually and click *OK* button.

Preliminary

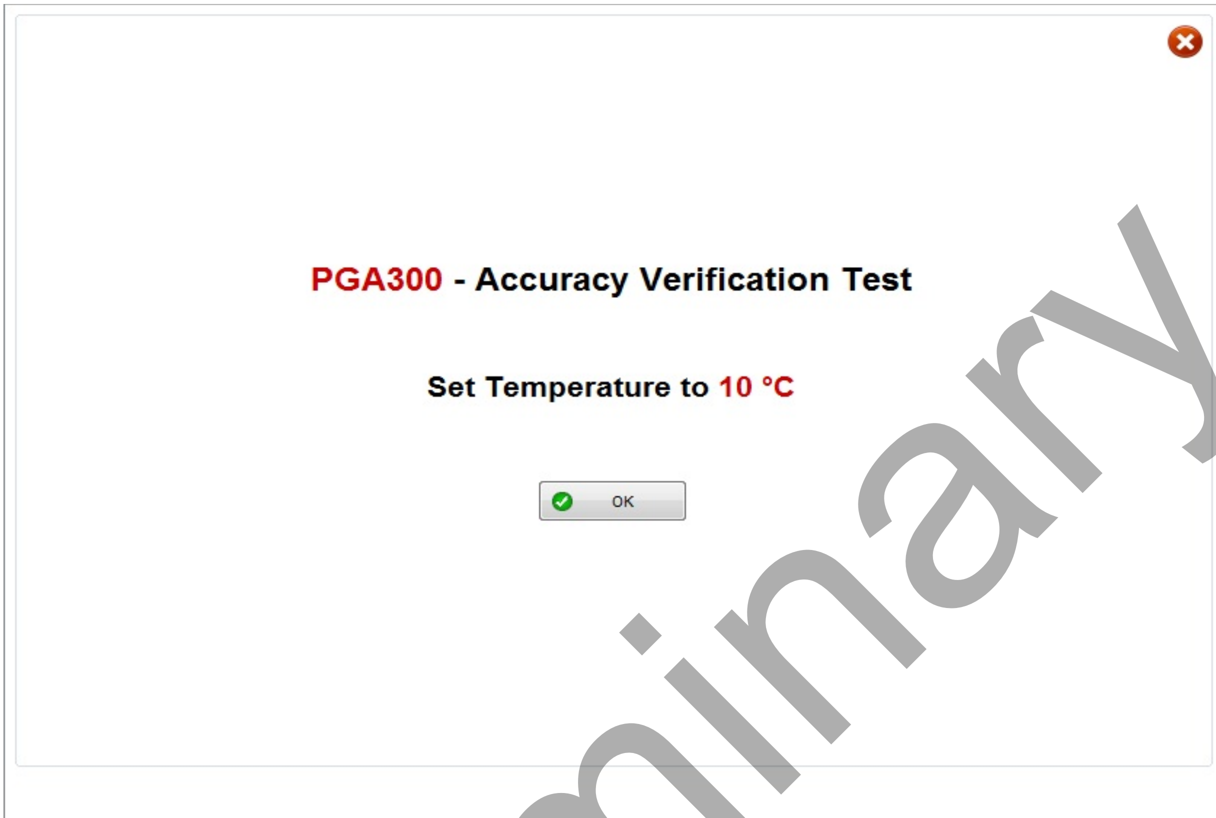


Fig 73: Accuracy Verification Test - Temperature Setting

b. Pressure Setting
   This page guide the user to set the proper pressure for accuracy verification test. Set the pressure manually and click *OK* button.

Preliminary

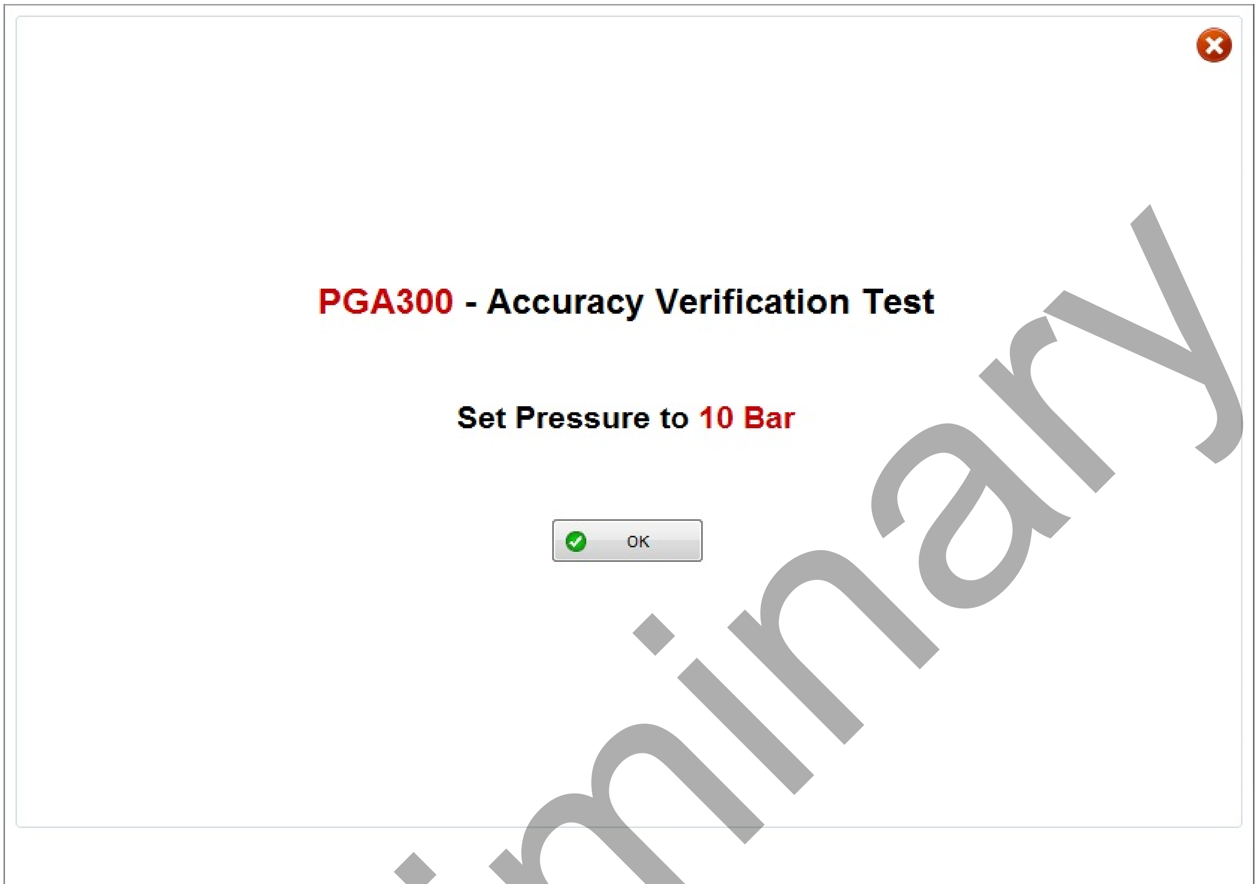


Fig 74: Accuracy Verification Test - Pressure Setting

c. AVT Data Collection
Measure the read voltage or current from the device depends on the mode and enter the values in the numeric control. This page also displays the desired current/voltage given in the *Calibration settings* File. When *OK button* is pressed it compares the measured value and expected value are within the %FSR specified by the user and update the status LED.

Preliminary

Fig 75: Accuracy Verification Test- Data Collection

The above steps will be repeated for all the temperature and pressure points given in the *Accuracy Verification Test - Settings Preview* Page. After collecting the required data proceed further to see the results.

d. The next page will show the result of Accuracy verification test. A device is considered as pass only if it is passed in all the data collection points. Even if the device has failed a single data collection point then it is considered as fail. For example in the figure shown below, Overall test result is considered to be failed because it failed in second test point even though it passed in first and third test point.

Preliminary

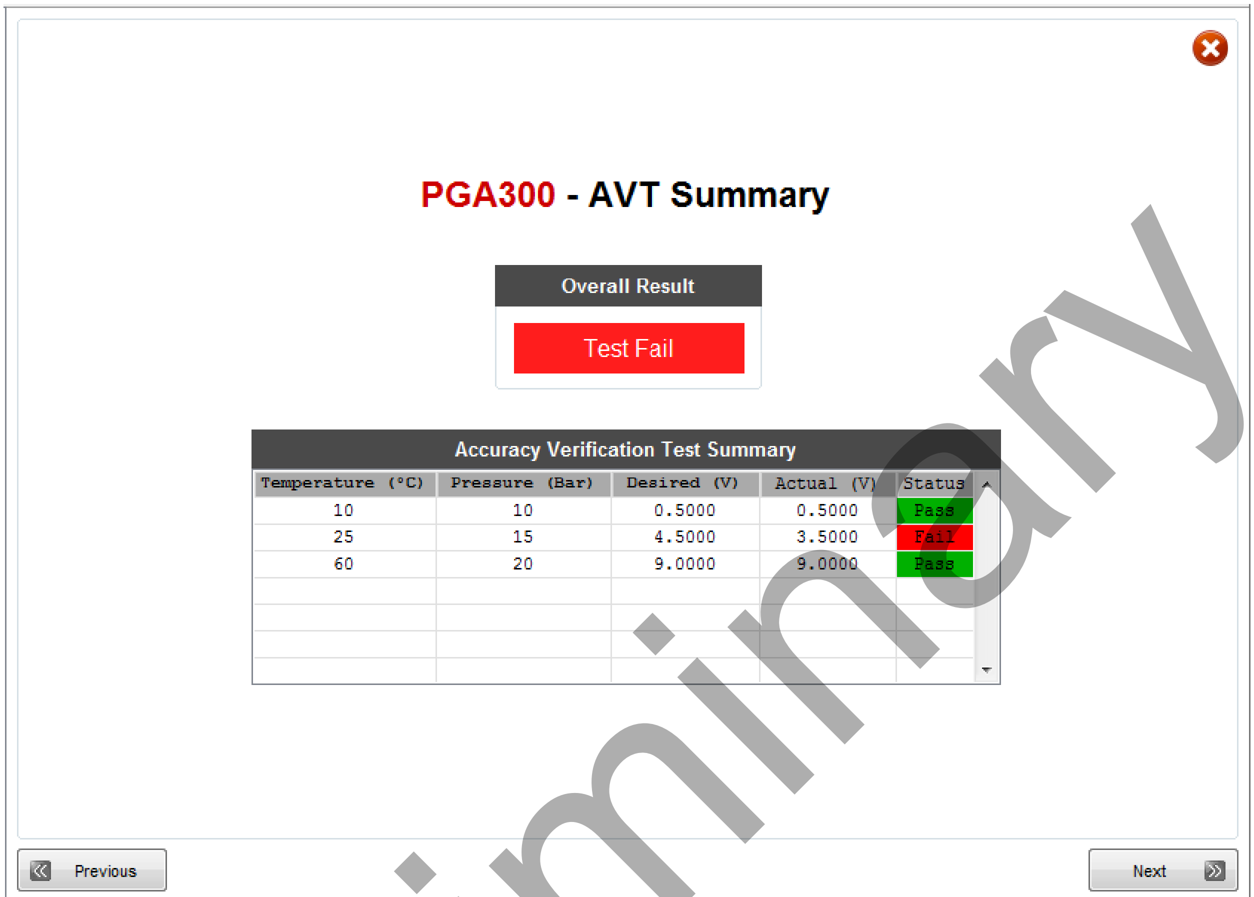


Fig 76: Accuracy Verification Test Summary

Preliminary

2.3.2. Debug

2.3.2.1. ADC Settings

- The ADC settings page displays the various ADC configuration settings of the PGA300 EVM board.
- The page contains the temperature and pressure sensor settings for the ADC. Temperature and Pressure operations could be switched by selecting the appropriate tab control.
- The table at the bottom of the tab control is used to read the ADC register values and display the corresponding Voltage/Temperature.

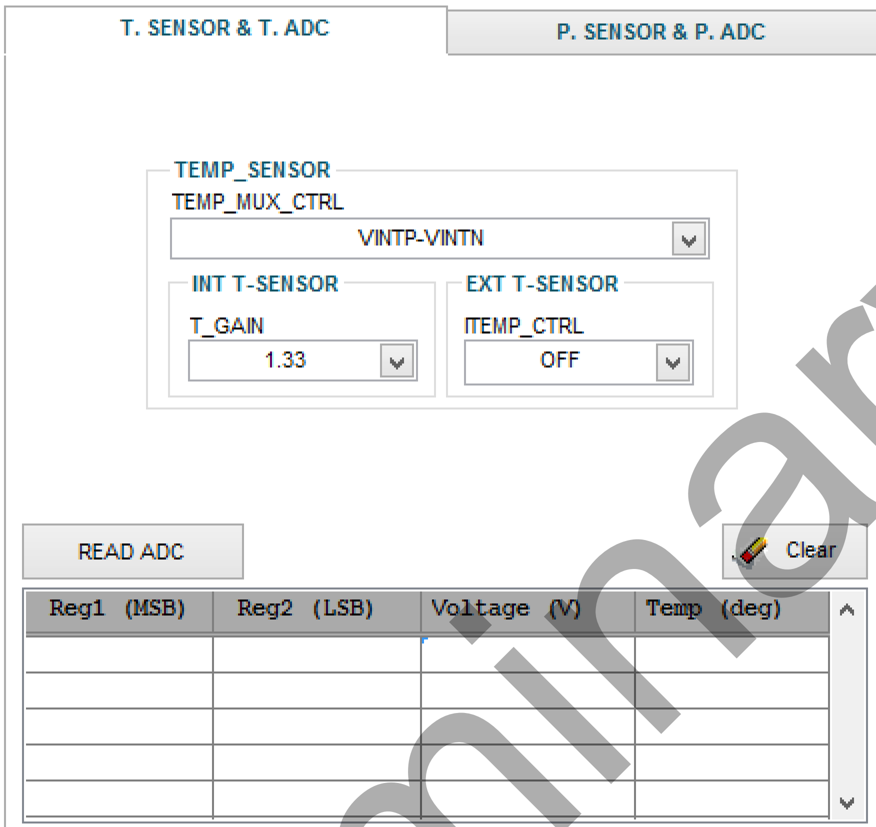


Fig 77: Temperature & Pressure Sensor Tab Selection

Preliminary

2.3.2.2. DAC Settings

The DAC Settings Page contain the DAC features of the device explained in the below subsections.

- The device has PWM logic that can drive the DAC Gain Buffer directly.
- The PWM functionality uses a 16-bit 4MHz Free Running Timer.

DAC Configuration

- The device includes a 14-bit digital to analog converter that produces an absolute output voltage with respect to the Accurate Reference voltage or ratio metric output voltage with respect to the VDD supply.

- When the microprocessor undergoes a reset, the DAC registers are driven to **Lower Clamp value** which can be configured to EEPROM by changing the value in **EEPROM Variable** section of **EEPROM Config - Digital Page**.
- The page also has options to Write and Read the DAC registers data

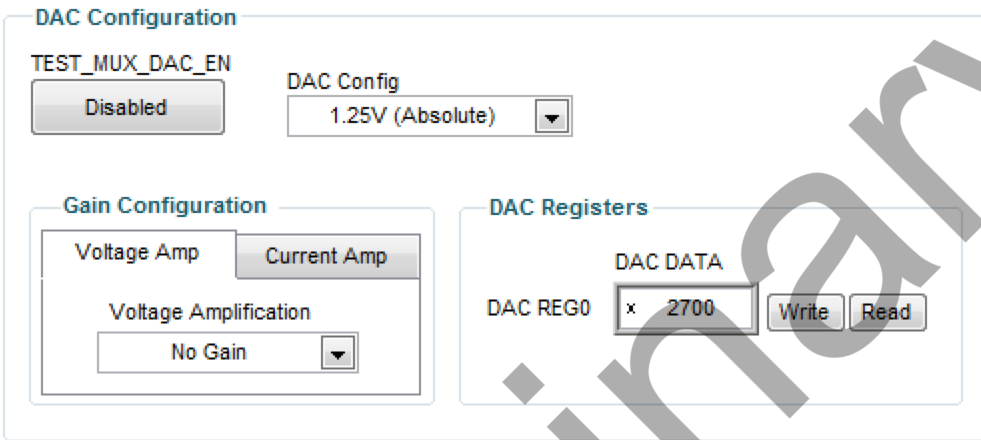


Fig 78: DAC Configuration

Ratiometric vs. Absolute

- The DAC output can be configured to be either in ratiometric-to-VDD mode or independent-of-VDD (or absolute) mode using the DAC_RATIOMETRIC bit in DAC_CONFIG.
- In Ratiometric mode, changes in the VDD voltage result in a proportional change in the output voltage because the current reference for the DAC is derived from VDD.

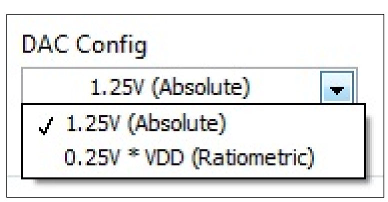


Fig 79: Configure DAC

DAC Gain

- The DAC Gain buffer is a configurable buffer stage for the DAC Output.

Preliminary

- The DAC Gain amplifier can be configured to operate in voltage amplification mode for voltage output or current amplification mode for 4-20mA applications.
- In voltage output mode, DAC Gain can be configured for a specific gain value by setting the **DAC_GAIN** bits in **DAC_CONFIG** register to a specific value.
- The DAC Gain can be configured to one of five possible gain configurations using the 2 bit DAC_GAIN field.
- The final stage of DAC Gain is connected to Vddp and Ground. This gives the ability to drive VOUT voltage close to VDD voltage.
- The DAC Gain buffer also implements a COMP pin in order to allow implementation of compensation when driving large capacitive loads.

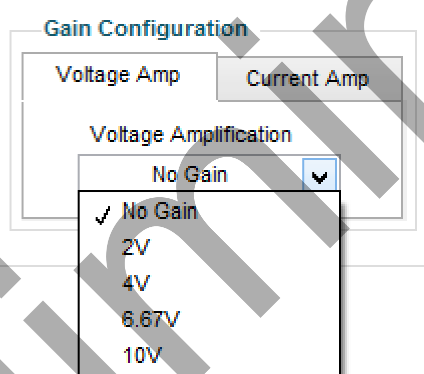


Fig 80: DAC Gain Configuration

2.3.2.3. ADC Graph

- The ADC Graph Page allows the user to configure the required ADC configuration and perform a continuous ADC capture.
- This page allows the user to switch between ADC Codes and Voltages.
- Once the required configurations are done on the UI, the capture can be started by hitting the **Start** button
- The continuous capture data will be displayed on the waveform graph on the UI
- Hit the **Stop** button to stop the capture at any point

Preliminary

Fig 81: ADC Graph

2.3.2.4. Bridge Configuration

- This page allows the user to operate the External Bridge Circuit present in the EVM.
- The GUI provides the ability to set variable resistance and balance the bridge.

The PGA300 device has an external bridge circuit. The external bridge has four legs and the arrow near the leg corresponds to the variable resistor. The balance button at the middle will be used to balance the leg resistances. The Top Left leg is a configurable resistance. The variable resistance value ranges to 3 decimal places but this is a theoretical calculation based on the device specification. Hence there could be minor variations in the equivalent resistance in the top left leg.

Preliminary

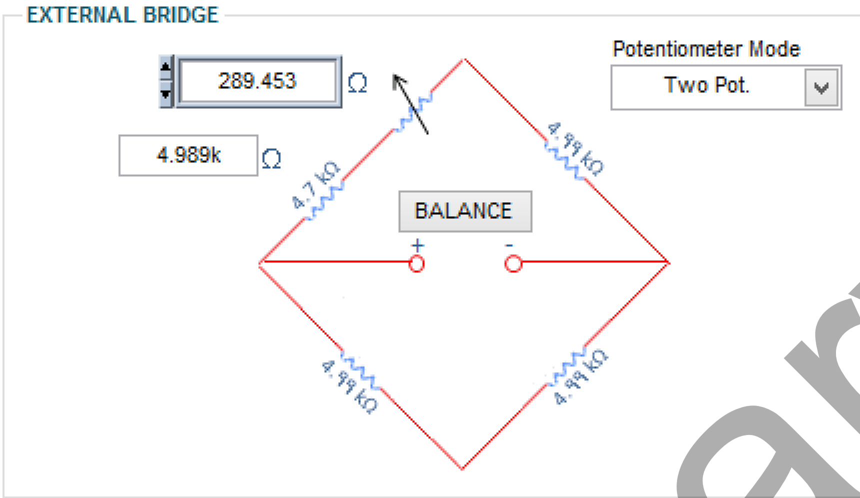


Fig 82: External Bridge Circuit

- There are also options available to set the Potentiometer mode using the drop down. The two available modes are,
  - Two Pot. Mode
  - One Pot. Mode

Preliminary

2.3.2.5. EEPROM Page

This page describes the EEPROM programming functionalities of the PGA300 EVM as shown in Fig.57. Each of the device functionalities are discussed in the following sub-sections.

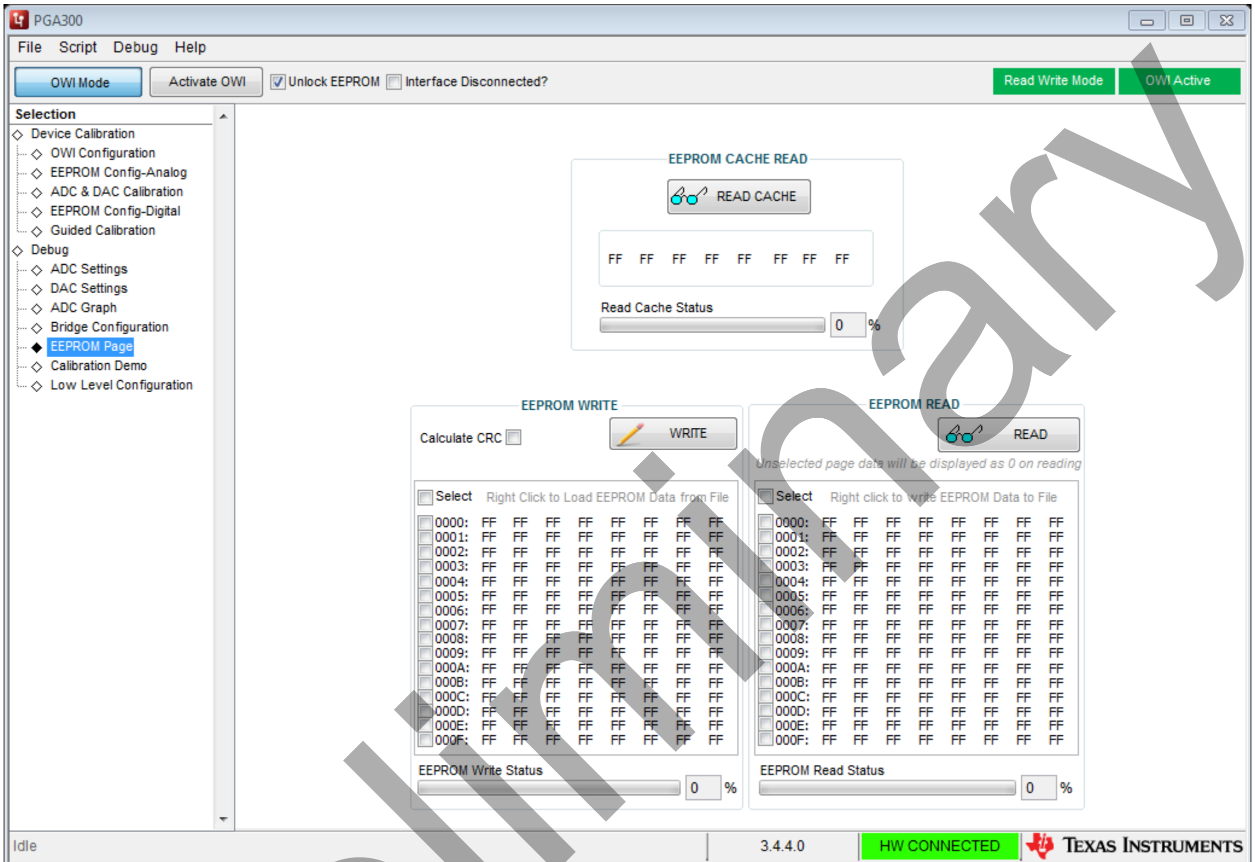


Fig 83: EEPROM Page

EEPROM WRITE & READ

- This section allows you to read-from or write-to the text files
- In the EEPROM Write Area on the GUI, Right-click to select an EEPROM File.

Preliminary

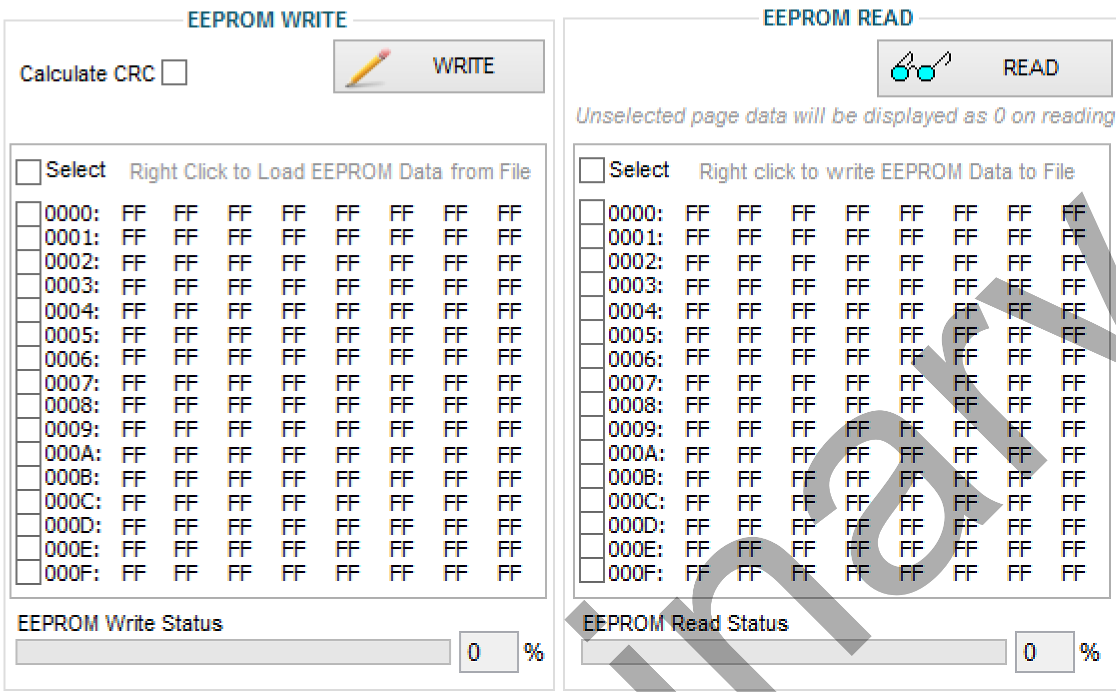


Fig 84: Selecting EEPROM Data File

- When a particular file is selected, the data is loaded as shown above. Individual byte can also be edited using the GUI before writing to the Device. The Calculate CRC (cyclic redundancy check) will calculate the CRC as per the logic defined in the datasheet for the write operation.
- CRC conditions
  - When checked with not all the pages selected, the GUI will read all the 128bytes of existing EEPROM data and replace the data of the selected page with the newly configured data from the EEPROM write box and calculate the resulting CRC. After the calculation, the GUI will write the data of the selected pages and the last page with updated CRC value to the EEPROM.
  - When all the pages are selected, the GUI will not read all the data but will directly calculate CRC from the EEPROM WRITE BOX and write to all the pages with updated CRC value to the last byte

Preliminary

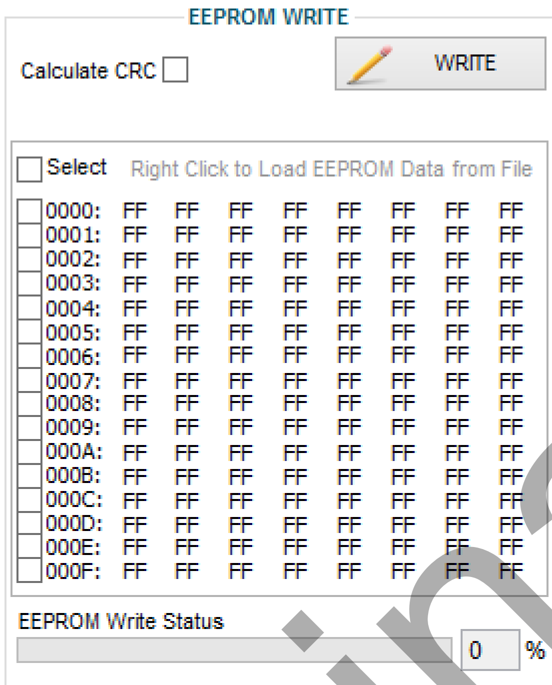


Fig 85: EEPROM Data Loaded

- Read EEPROM function provides the user the feature to read back the EEPROM DATA

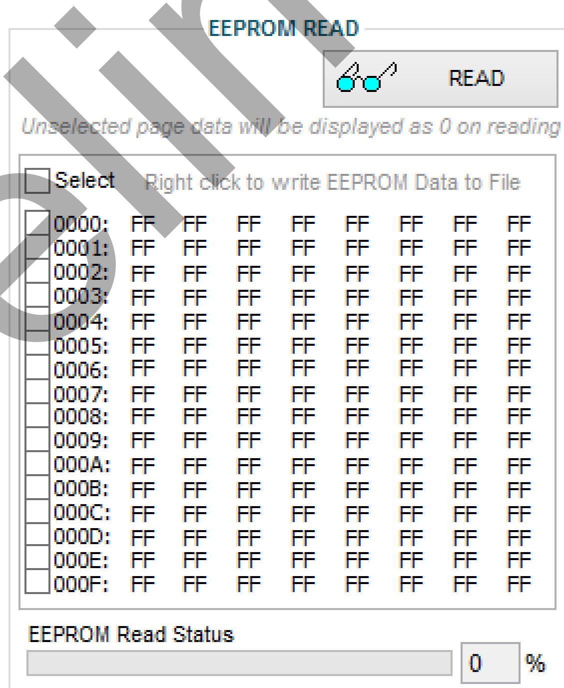


Fig 86: EEPROM Read Data

Preliminary

EEPROM pages can be individually selected and written or read using the check box on the left side of the data map.

EEPROM Cache read

This function will help the user to read back the 8 bytes of data from the EEPROM Cache

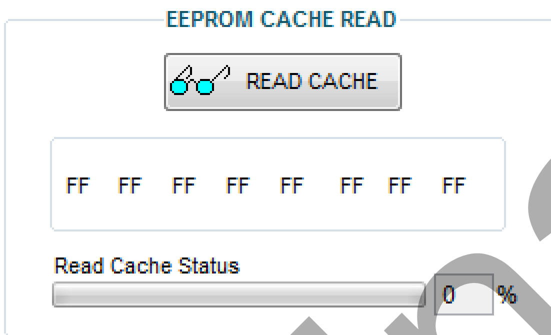


Fig 87: EEPROM CACHE Read

Preliminary

2.3.2.6. Calibration Demo

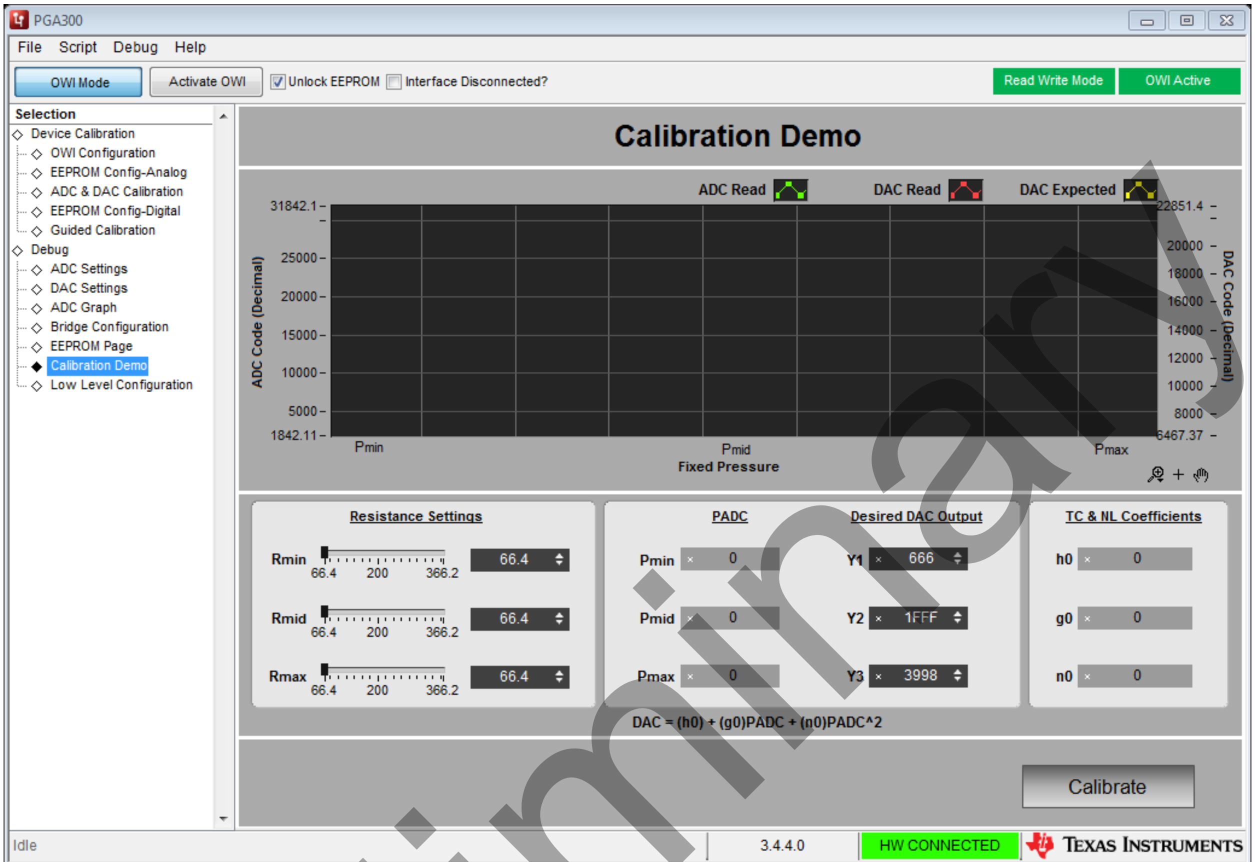


Fig 88: Calibration Demo

This page demonstrates the PGA300 Calibration process. Follow the below mentioned steps to carry out a calibration process.

- Hit the Activate OWI button
- Configure the three resistance values (**Rmin**, **Rmid** & **Rmax**)
- Enter the Desired DAC output values (**Y1**, **Y2** & **Y3**)
- Hit the **Calibrate** Button
- The various stages of the Calibration process will be displayed on the status bar which appears on the bottom of the page
- Hit the **Stop** button next to the Calibrate button to abort an on-going calibration process

Preliminary

User has to hit the Activate OWI button every time before starting the calibration process.

When the user presses Calibrate button, there are set of operations which are performed internally by the automated calibration process. Mentioned below are the various operations carried out during the process.

- Bridge set to One Pot. mode
- Resistance values are configured to the bridge
- For every configured resistance value, corresponding PADC value will be read.
  - PADC capture is done in multiple iterations by capturing multiple samples and averaging the acquired samples.
  - The number of samples can be specified in the **Capture Iterations** control present in the ADC and DAC Calibration Page. *(Default value is 8)*
- Once all the ADC values are captured, the coefficients are calculated and written into the EEPROM
- After writing the coefficients into the EEPROM, the Micro controller is enabled
- Next step is to read the DAC values from the Comm buff
  - DAC capture is also done in multiple iterations by capturing multiple samples and averaging the acquired samples.
  - The number of samples can be specified by right clicking on the **Desired DAC Output** option and entering the required value in the appearing popup window. *(Default value is 8)*
- Once all the DAC values are read, the final process is to plot all resulting values on the waveform graph. The DAC values read will be annotated on the graph at the respective points.

Preliminary

2.3.2.7. Low Level Configuration Page

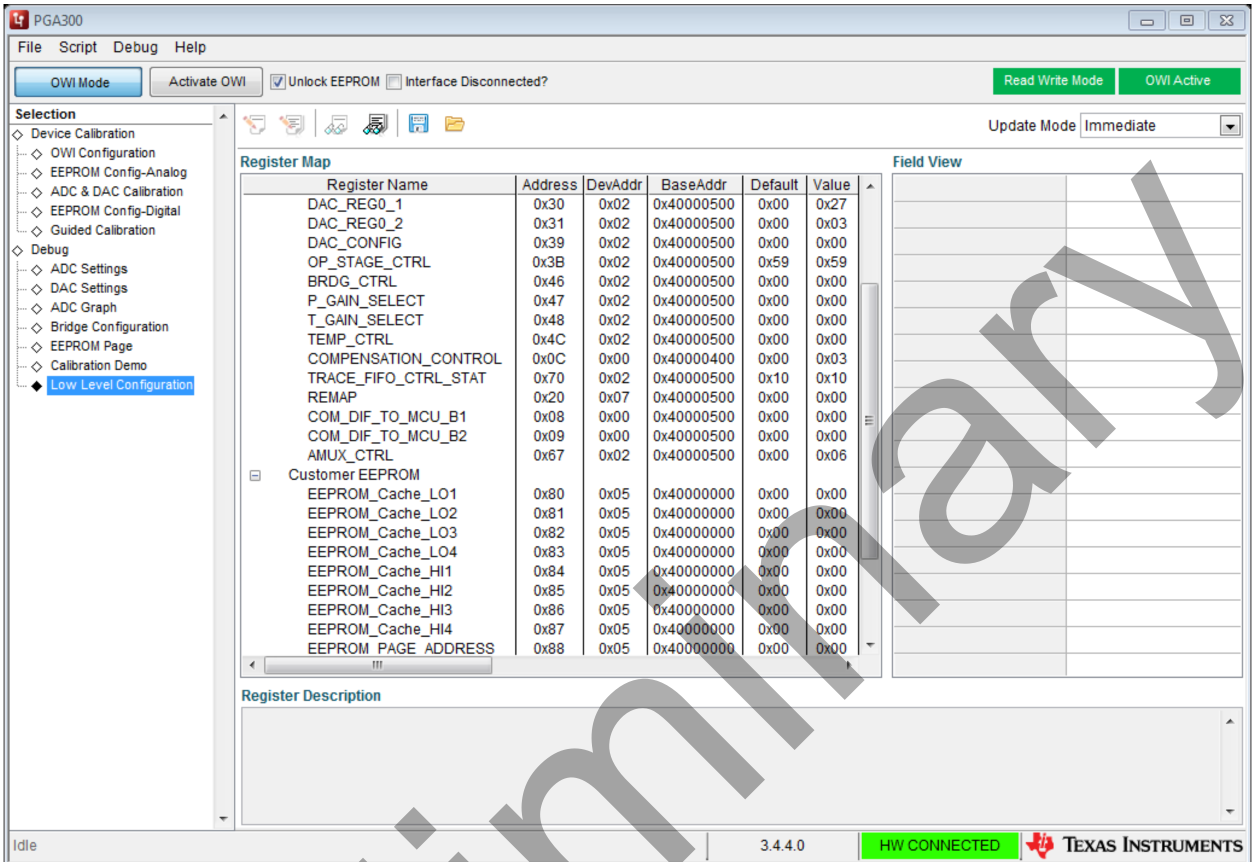


Fig 89: Low level Configuration

- Low level configuration page provides a detailed view of all the registers that the device possesses.
- This page allows the users to read from and write to the registers.
- When a particular register is selected, the corresponding register description is displayed at the bottom left of the page.
- Register write modes
  - **Immediate mode** - The register values will be written to device immediately.
  - **Manual mode** - The register values will be written to device when 'Write Register' or 'Write Modified' button is pressed. The changed register values will

Preliminary

be highlighted in blue. When there are some pending changes and update mode is changed from manual to immediate, a dialog box will appear. Choose required operation to be carried on from the dialog box.

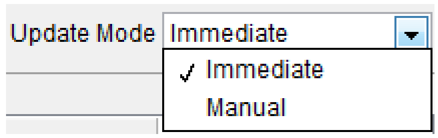


Fig 90: Update Mode

Fig 91: Pending Changes Dialog Box

- Changing the Value of register can be done by
    - **Register level operation**
        - Select the register that has to be edited
        - Double click on the value column corresponding to the register
        - Enter the register value (Hex) in the edit box.

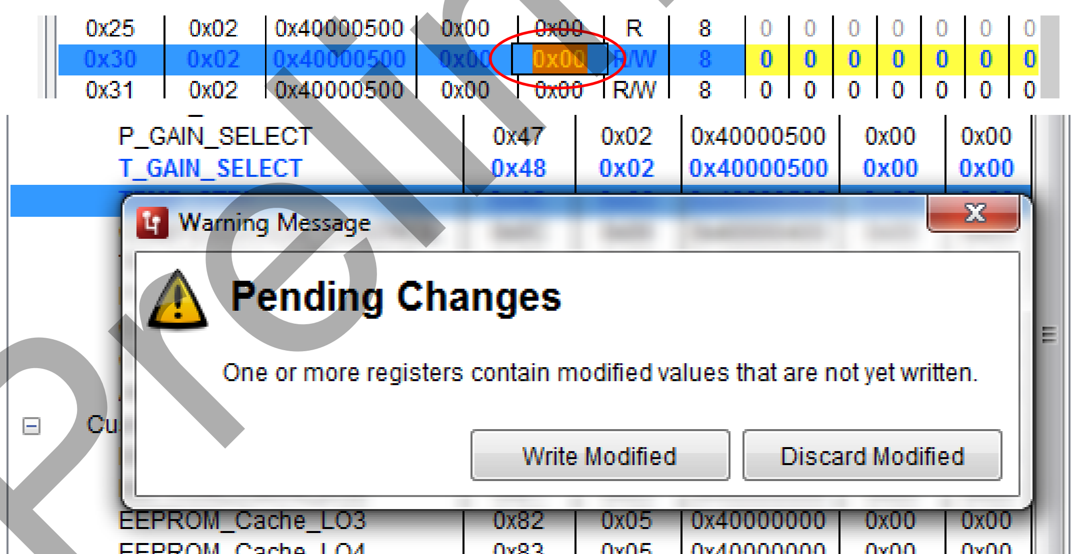


Fig 92: Register level value change

-
    - **Field level Operation**
        - Select the register that has to be edited

Preliminary

- The fields corresponding to the register will be listed in the 'Field View' section
- When you hover mouse over the value of the field, edit box will appear and the corresponding bits will be highlighted. You can change the appropriate value in the edit box.

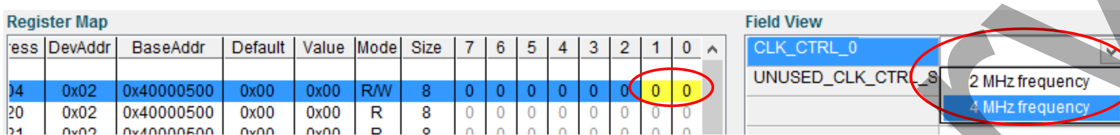
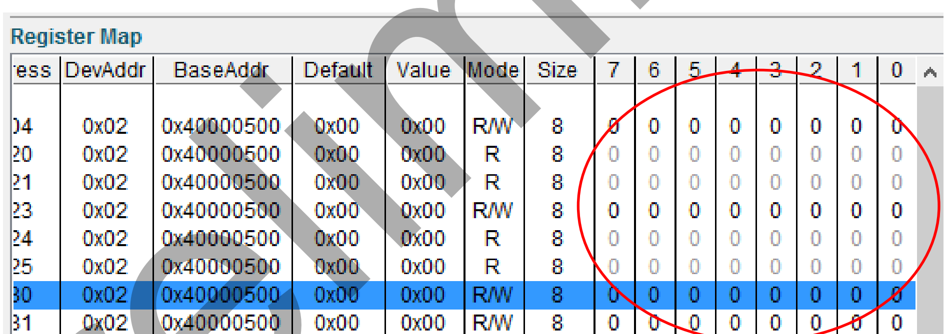


Fig 93: Field level value change

- **Bit level operation**
  - Select the register that has to be edited.
  - Value of each bit in the register can be changed by clicking on the '0' or '1' in the corresponding bit column. The bits that are greyed out are read only bits. These bits cannot be written.



Register Map

ess	DevAddr	BaseAddr	Default	Value	Mode	Size	7	6	5	4	3	2	1	0
04	0x02	0x40000500	0x00	0x00	R/W	8	0	0	0	0	0	0	0	0
20	0x02	0x40000500	0x00	0x00	R	8	0	0	0	0	0	0	0	0
21	0x02	0x40000500	0x00	0x00	R	8	0	0	0	0	0	0	0	0
23	0x02	0x40000500	0x00	0x00	R/W	8	0	0	0	0	0	0	0	0
24	0x02	0x40000500	0x00	0x00	R	8	0	0	0	0	0	0	0	0
25	0x02	0x40000500	0x00	0x00	R	8	0	0	0	0	0	0	0	0
30	0x02	0x40000500	0x00	0x00	R/W	8	0	0	0	0	0	0	0	0
31	0x02	0x40000500	0x00	0x00	R/W	8	0	0	0	0	0	0	0	0

Fig 94: Bit level value change

Load and Save Register Configuration Feature

- **Save Config** - When you click on this button, the current register configuration will be saved into a file which can be later loaded into the GUI using the Load option.

Fig 95: Save Config

Preliminary

- **Load Config** – Click on this button to load the configuration file which was saved earlier to bring the device to a known state. Please make sure Microcontroller is reset before you load the config.

Fig 96: Load Config

- When you select any of the above options, a message pops up on the screen. Select the file path (to load /save the configuration file) and press **Ok**

Load Config will overwrite the existing data in registers with the value specified in the .cfg file loaded.

Register Read and Write

- The Register Data displays the fields of the selected register and value of each bit.
- **Read Register** - When the Read Register button is pressed, the value will be read from the device and displayed in the Register map tree. The filed view will also be updated with the new values
- **Write Register**- The field view will display the values to be written to register (field wise view). The hex equivalent data that will be written to the register is displayed in the value column corresponding to the selected register in Register Map tree.
- **Read All** – When you press this button, the data is read from all the registers based on the mode (Read and Read/Write mode).
- **Write Modified** – When you press this button, any value entered in the display box is written to all the registers that are in Write and Read/ Write mode. This option will be enabled only when update mode is manual.

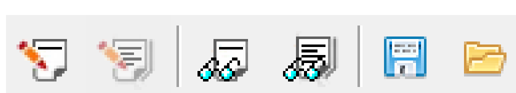


Fig 97: Low level page operations

Preliminary

2.3.3. About Page

The About page contains information about the GUI such as the GUI Name, Version Information and Supported OS. The about page appears as shown below.

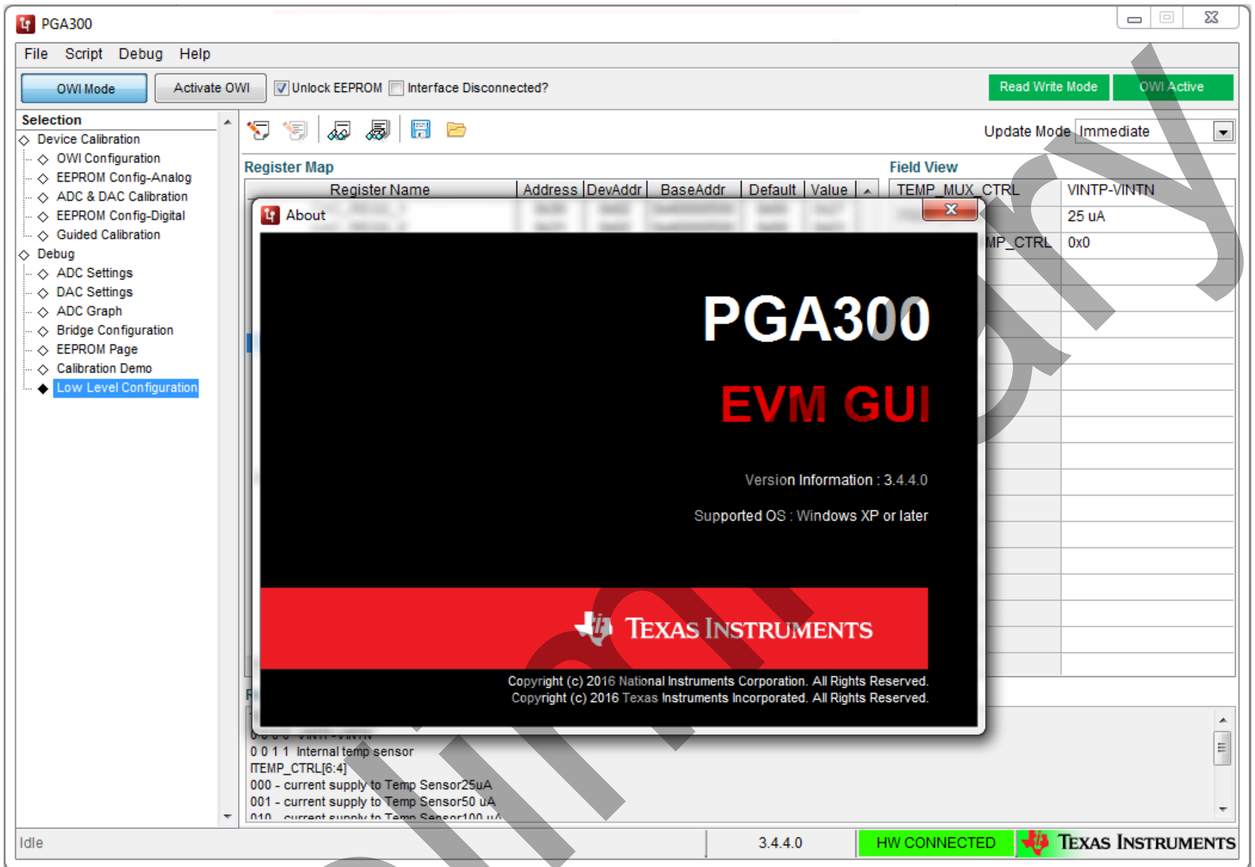


Fig 98: About Page

2.4. Menu Options

2.4.1. File

The File menu contains the Exit option as shown in *Fig 99: File Menu*. The **Exit** option is used to stop the execution of the PGA300 GUI.

Preliminary

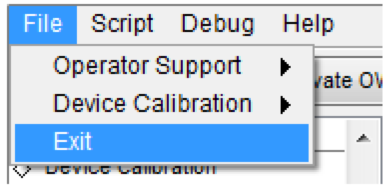


Fig 99: File Menu

2.4.2. Script

- Scripting is used to automate the device operations and reduce the time consumption in repeating similar operations.
- This is helpful in situations where performing a particular device function may require setting 10 to 15 registers on the device to a particular value. In these circumstances, scripts could be recorded and run whenever needed.
- In PGA300 GUI, the scripting is done using Python because,
  - ✓ It's easier to implement
  - ✓ More widely used
  - ✓ More user friendly

2.4.2.1. Performing Macro Recording

- To create a custom macro, click **Script->Launch Window,**

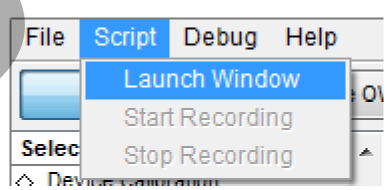


Fig 100: Launch Window

- Python IDLE window appears as shown below,

Preliminary

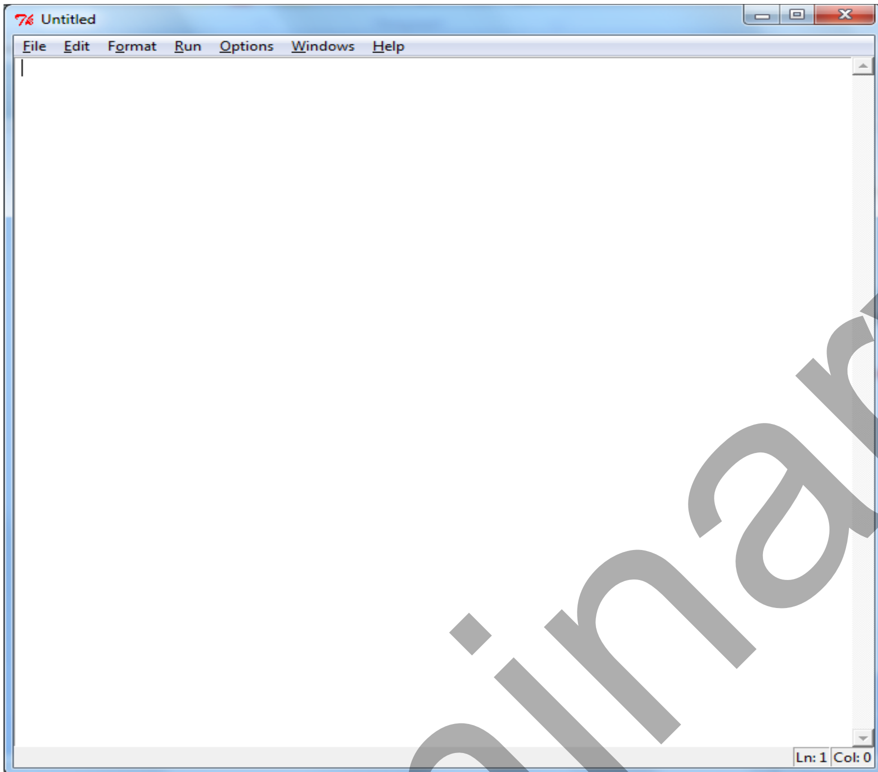


Fig 101: Python Idle Window

Selecting the Launch Window again will open another untitled window and the one opened last will be active.

- Click **Script->Start Recording**, IDLE window becomes green showing that recording is started

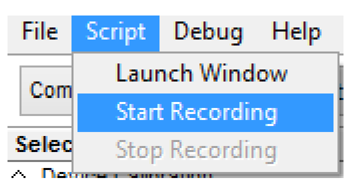


Fig 102: Start Recording

- Go to "Low Level Page" of PGA300, select "P_GAIN_SELECT" register and write "0x01" to "Data" control and press "Write Register". Action is recorded as follows,

Preliminary

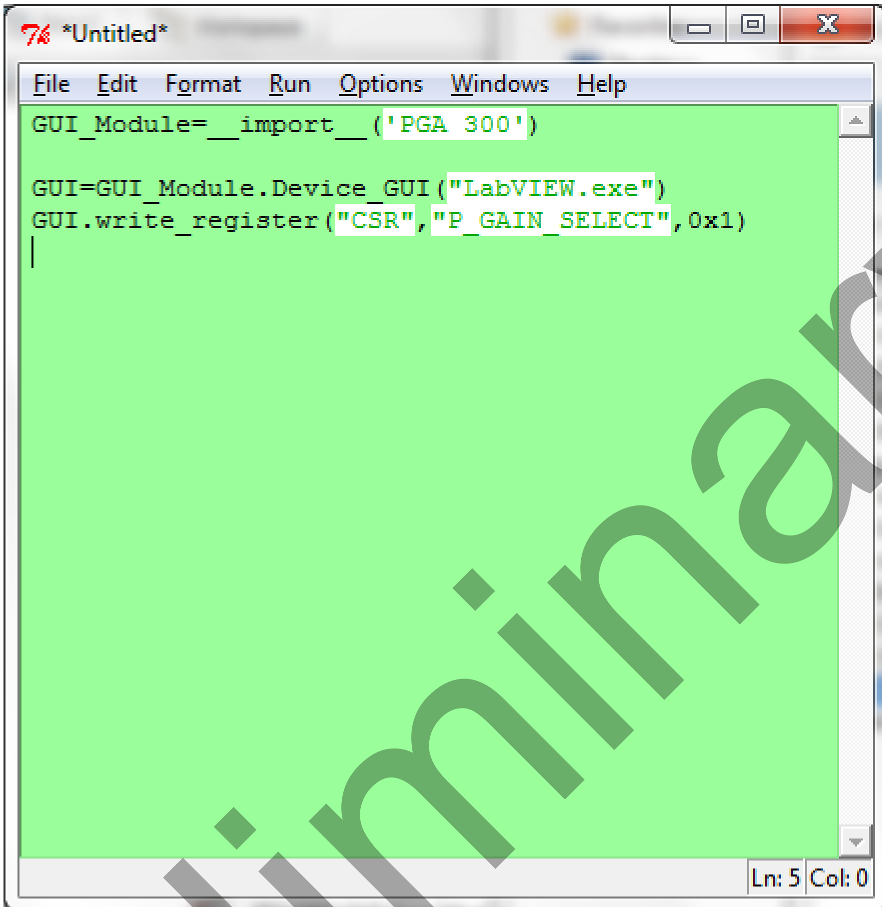


Fig 103: Macro Recording

- Stop the recording by clicking on **Script -> Stop Recording**.

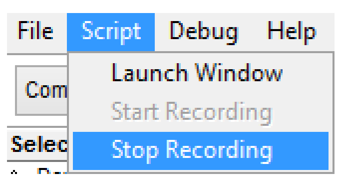


Fig 104: Stop Recording

- Python IDLE window will no longer be green indicating that the recording has been stopped.

Preliminary

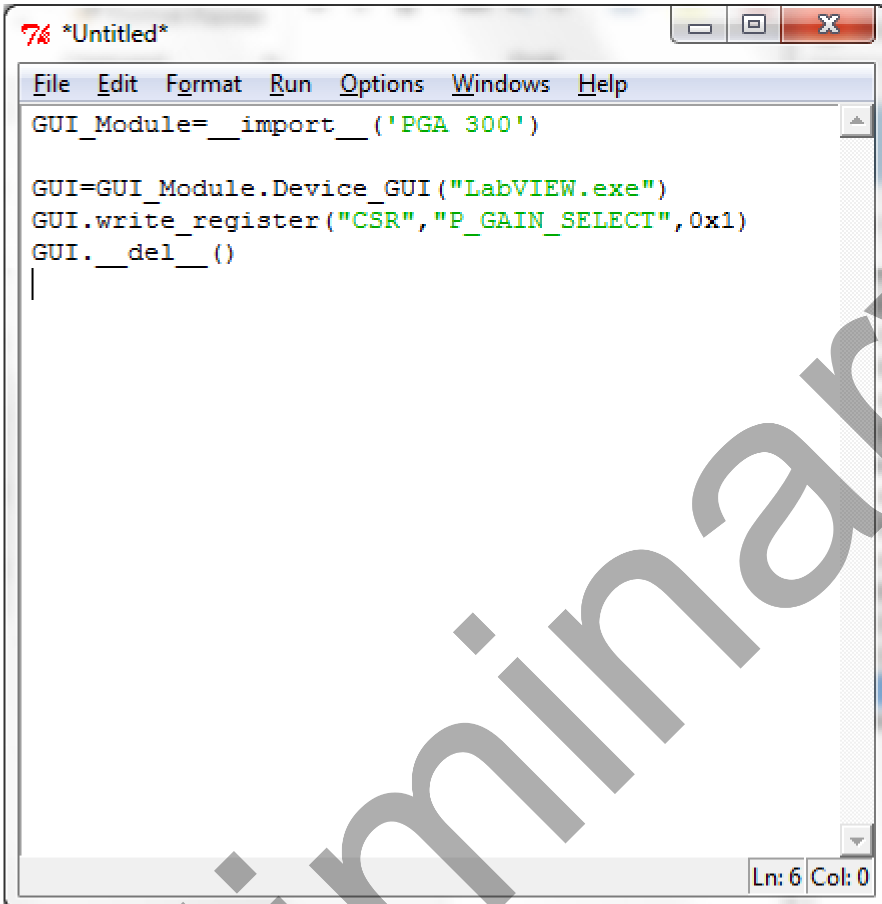


Fig 105: Finished Macro recording

- Run the script by either clicking on **Run -> Run Module** in the IDLE window or press F5.

Preliminary

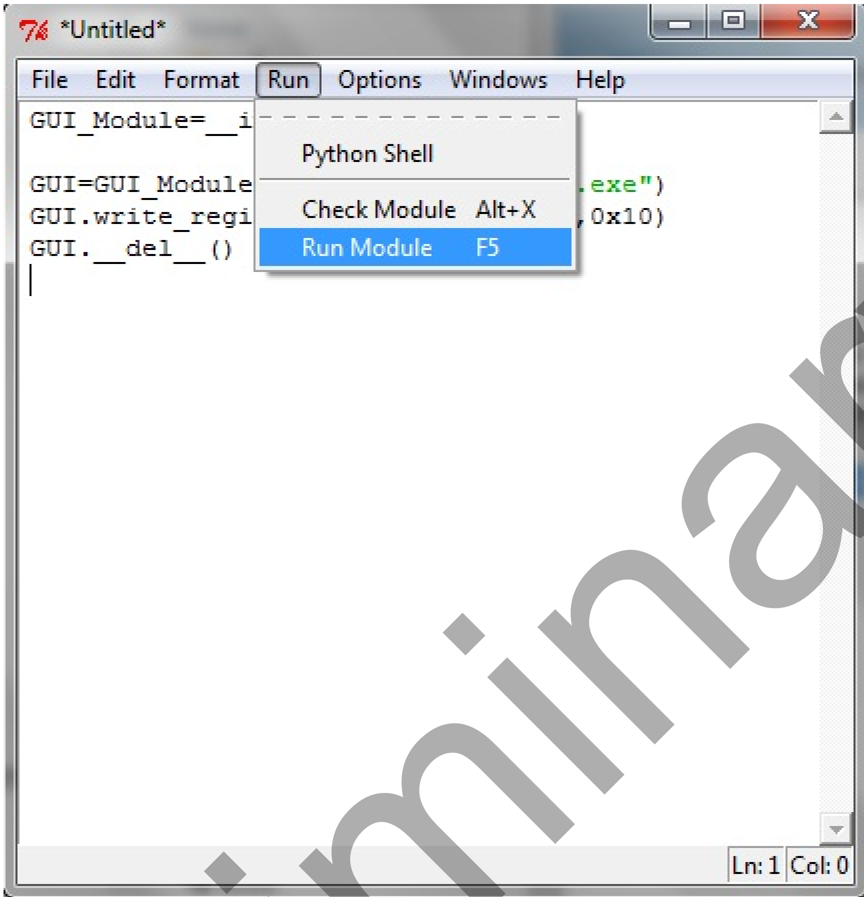


Fig 106: Run Module

- "Save As" dialog will appear asking for the file name for the script.
- Give any name say "Test.py" and click "Save".
  NOTE: Name of the script should have the extension "py"

Preliminary

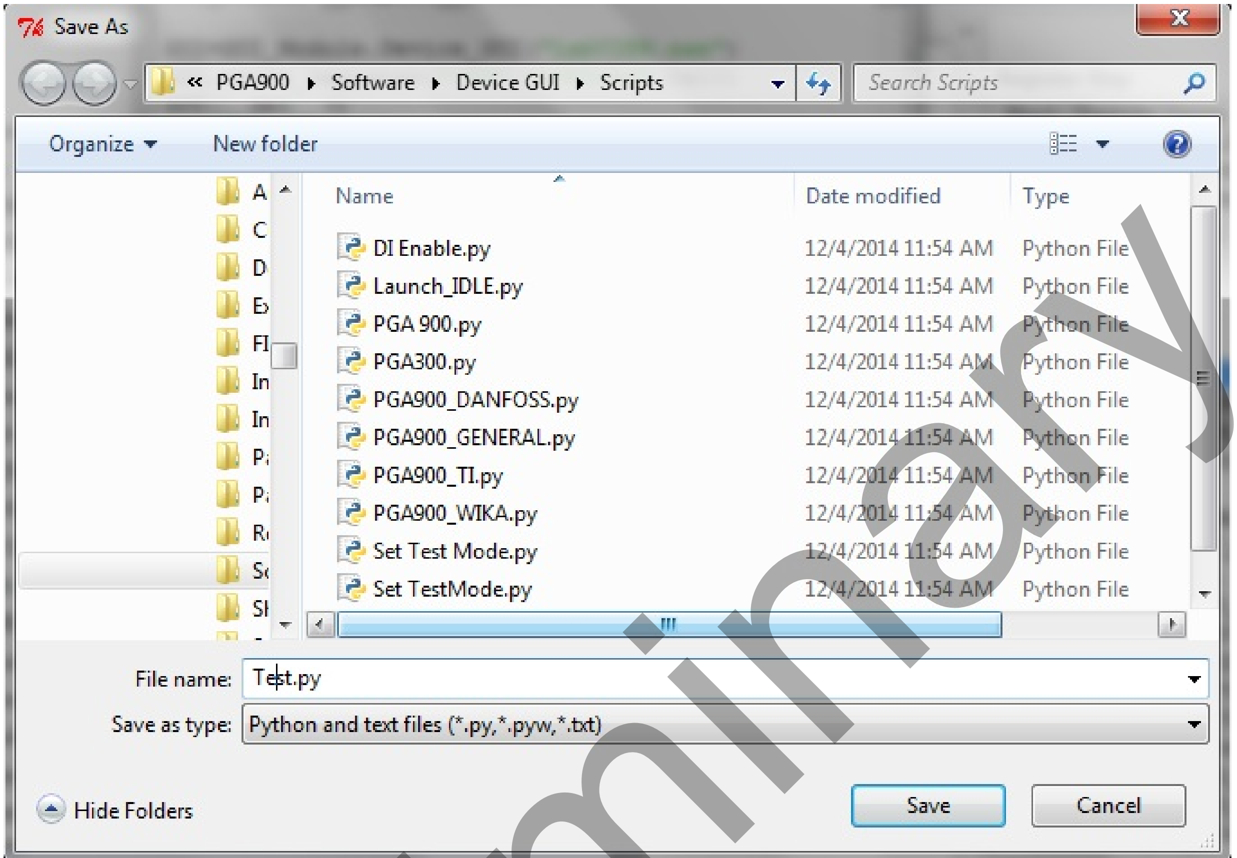


Fig 107: Save browser window

- Now the script will run and the status will be displayed in the python shell window as shown below,

Preliminary

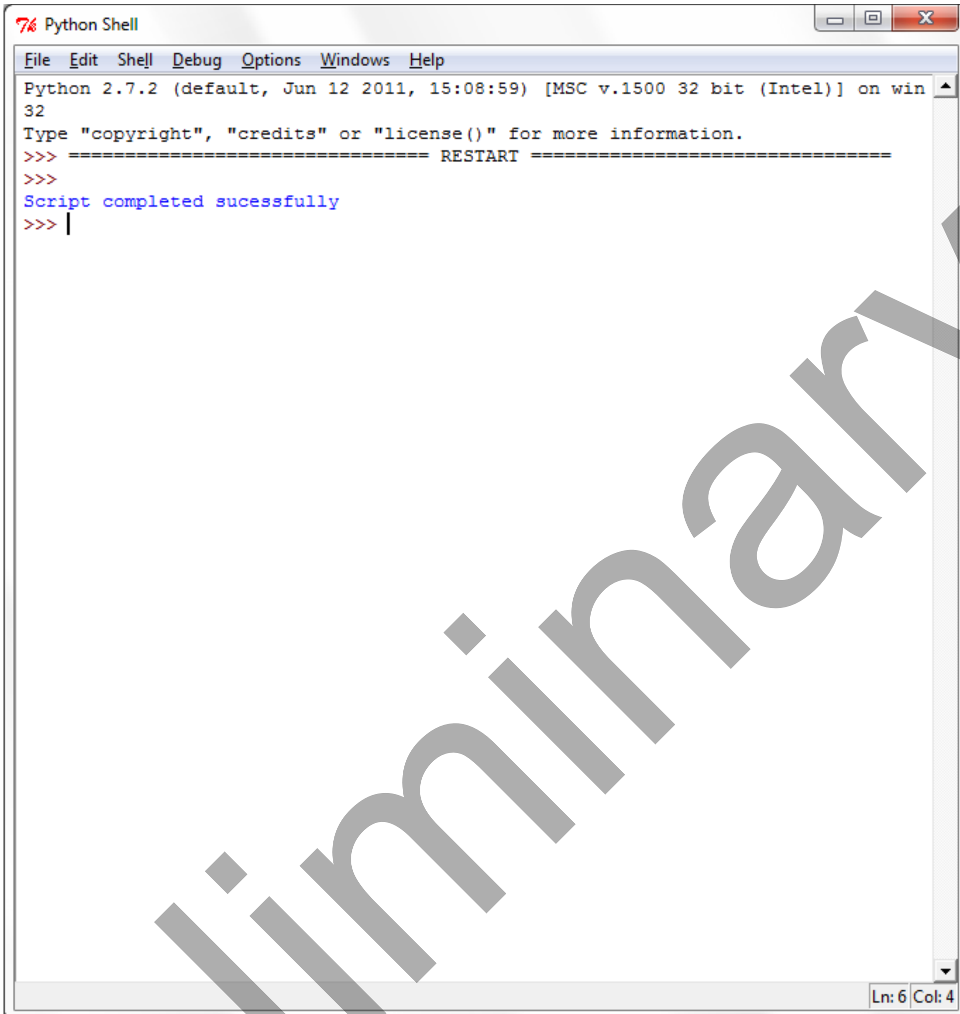


Fig 108: Run saved Macro

- To see the result, refer to the register values in the application and they will be the same as configured by the script. "Read Register" operation can also be recorded similarly

Preliminary

2.4.3. Debug

The Debug option can be used for the following operations

- **USB2ANY Disconnected** - By selecting this option from the submenu, the PGA300 GUI is run with no device connected and by unselecting it, the PGA300 GUI is run in USB2ANY connected mode.
- **File Logging** - The log to file submenu is used to log the GUI activities to a log file that is specified.
- **Debugging** - The Debug log option will enable to log all the activities of the user. If that is not selected, only the high level operations will be logged.

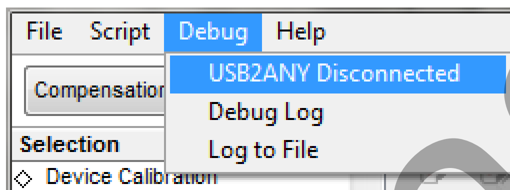


Fig 109: Debug Menu

3. Handling Configuration File

The GUI loads the required configuration settings for the various Calibration pages from a configuration file (Operator Support.ini). This can be found parallel to the application. Operator support configuration files can be selected from **File Menu >> Operator Support >> Load Configuration**. From the dialog box select an Operator Support file to be loaded. The various sections of the configuration file have been explained in detail below.

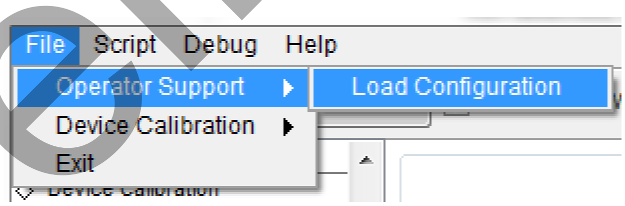


Fig 110: Operator Support File Selection

Preliminary

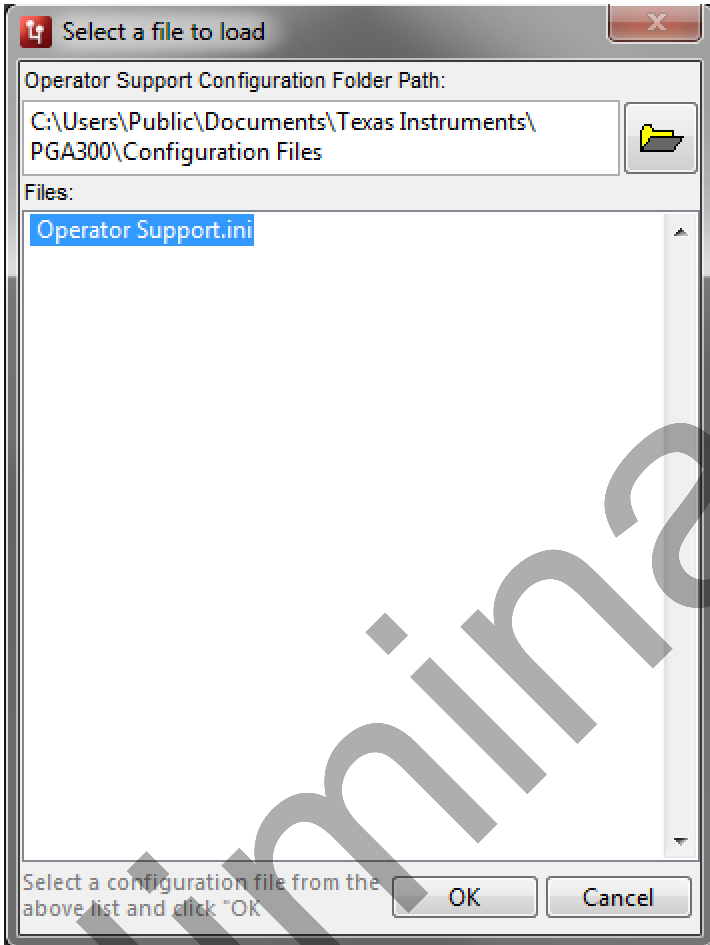
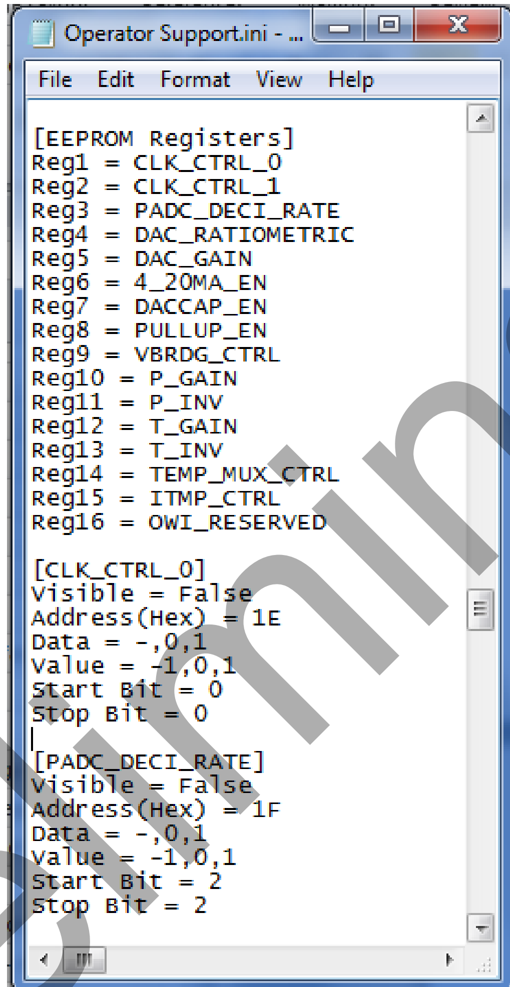


Fig 111: Load Configuration Menu & Dialog Box

Preliminary

3.1. EEPROM Registers

This section of the Configuration file is used to handle EEPROM Config – Analog Page. It enables the user to Add/Remove/Hide registers into/from the EEPROM Register Details table.



```
[EEPROM Registers]
Reg1 = CLK_CTRL_0
Reg2 = CLK_CTRL_1
Reg3 = PADC_DECI_RATE
Reg4 = DAC_RATIOMETRIC
Reg5 = DAC_GAIN
Reg6 = 4_20MA_EN
Reg7 = DACCAP_EN
Reg8 = PULLUP_EN
Reg9 = VBRDG_CTRL
Reg10 = P_GAIN
Reg11 = P_INV
Reg12 = T_GAIN
Reg13 = T_INV
Reg14 = TEMP_MUX_CTRL
Reg15 = ITMP_CTRL
Reg16 = OWI_RESERVED

[CLK_CTRL_0]
Visible = False
Address(Hex) = 1E
Data = -,0,1
Value = -1,0,1
Start Bit = 0
Stop Bit = 0

[PADC_DECI_RATE]
Visible = False
Address(Hex) = 1F
Data = -,0,1
Value = -1,0,1
Start Bit = 2
Stop Bit = 2
```

Fig 112: Configuration File - EEPROM Registers

- To add a register into the EERPOM Config – Analog page, enter the Register Name in the [EEPROM Registers] section as displayed in the above image.
- Add details of the register in a new section with the register name as the section name. (eg: [CLK_CTRL_0]) The various elements of the register section has been described below
  - ➢ VISIBLE – If set to 'False' the register will not be displayed on the GUI.
  - ➢ ADDRESS – Denotes the address location in the EEPROM to which the particular register corresponds to. This field takes Hexadecimal values

Preliminary

- DATA – Denotes the string which will be displayed in the drop down menu which appears in the Input Data column for particular register.
- VALUE – Denotes the actual value to be written into the EEPROM on selecting a particular DATA from the Input Data Drop down. This field takes decimal values
- START/STOP BIT – Denotes the bit position of the particular register in the EEPROM address

- To permanently remove a register from the GUI, delete the register name from the [EEPROM Registers] section.

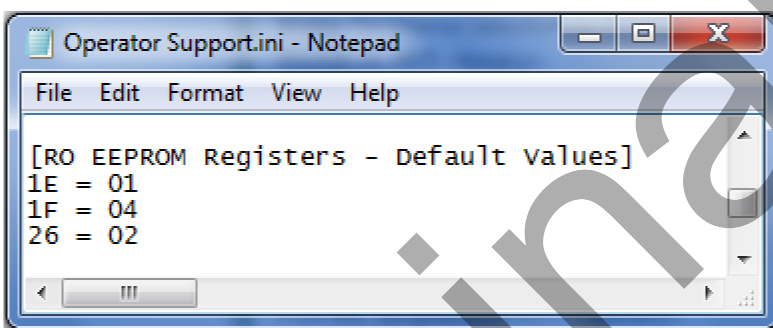


Fig 113: Restricted EEPROM Default Values

- The restricted EEPROM registers details are mentioned in the [RO EEPROM Registers - Default Values] section. The Key names are the Register Addresses and the respective values are their default values. Both field takes hexadecimal values.

3.2. ADC & DAC Calibration

This section of the configuration file contains the default values for the various ADC and DAC calibration modes. To change the default values to be displayed on the GUI, make required changes to this section in the configuration file. The various sections of the ADC & DAC Calibration configuration file have been explained below.

Preliminary

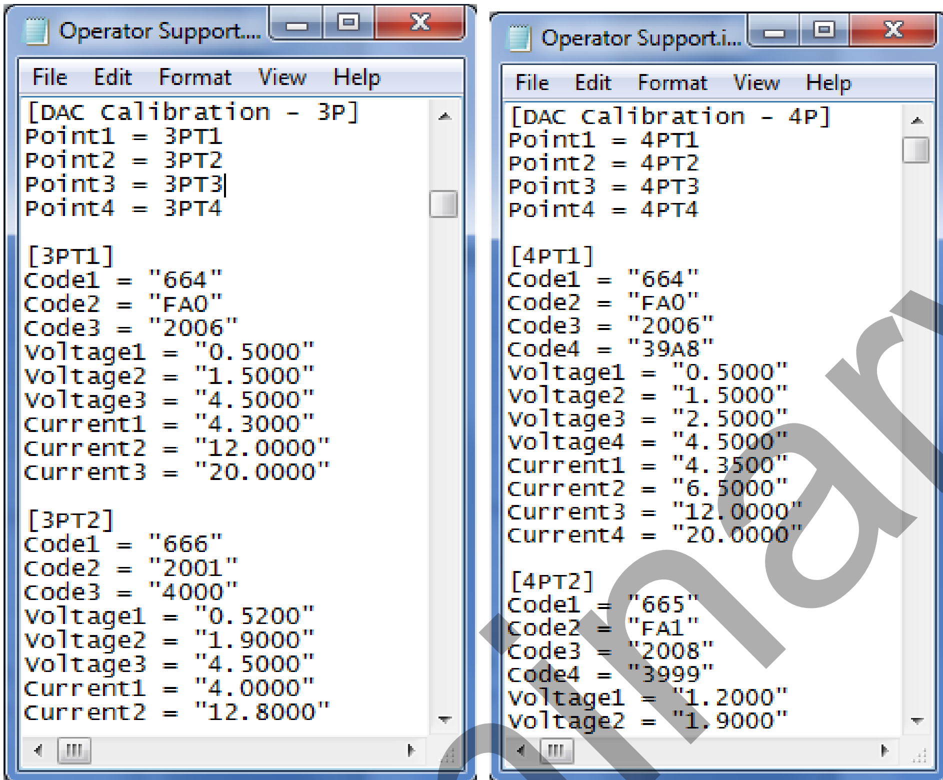


Fig 114: Configuration File – Default DAC Codes

- The default values for each ADC Calibration mode, to be filled into the DAC tables have to be updated into the configuration file as displayed in the above image. The Voltage values are in V. The current values are in mA.
- For 2P – 2T mode, the DAC codes are taken from the [DAC Calibration – 3P] section. The Code1 and Code3 are considered for the DAC Code calculation. Similarly for Voltage and Current, the first and third values are considered.
- For 3P – nT mode, the DAC Codes are taken from the [DAC Calibration – 3P] section and all the 3 codes are considered for the calculation.
- For 4P – 4T mode, the DAC Codes are taken from the [DAC Calibration – 4P] section and all the 4 codes are considered for the calculation.

Preliminary

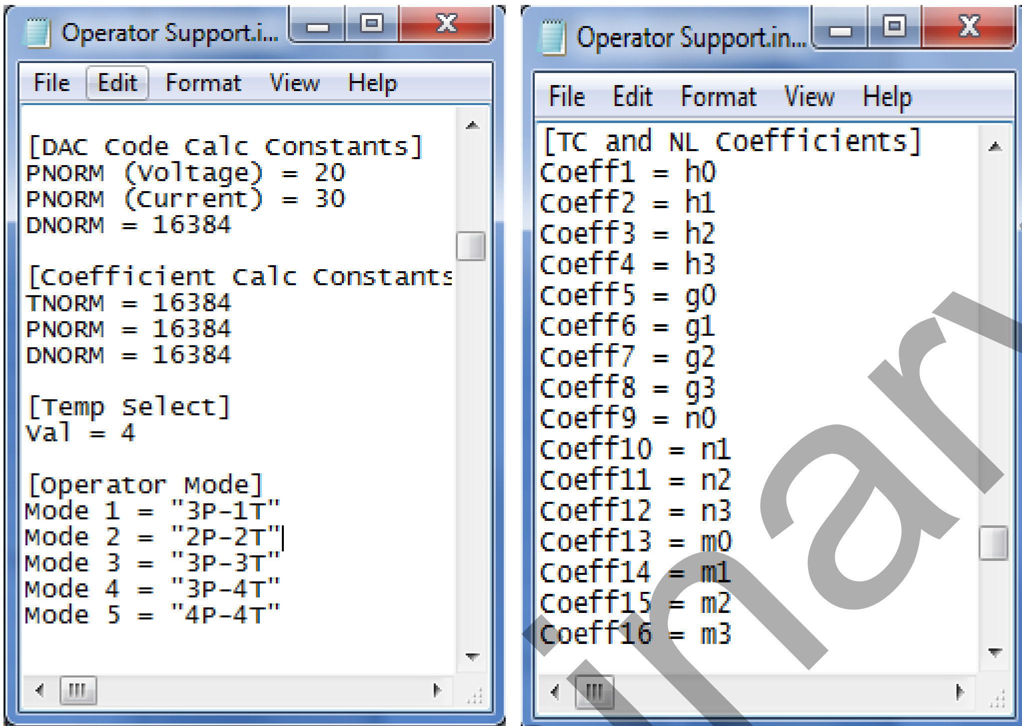


Fig 115: Configuration File – Coefficient Calculation

- The various ADC Calibration Modes available in the GUI are getting loaded from the [Operator Mode] section in the configuration file.
- To Add/Remove a particular calibration mode, make the corresponding changes to this section by adding/removing the required modes.
- While Adding/Removing a particular calibration mode, the related TC and NL Coefficients also need to be added or removed.
- The [Temp Select] section defines the maximum number of temperature points available in the GUI. (The maximum number of points allowed is 4)
- The [Coefficient Calc Constants] section defines the various constants used for the coefficient calculation.

Preliminary

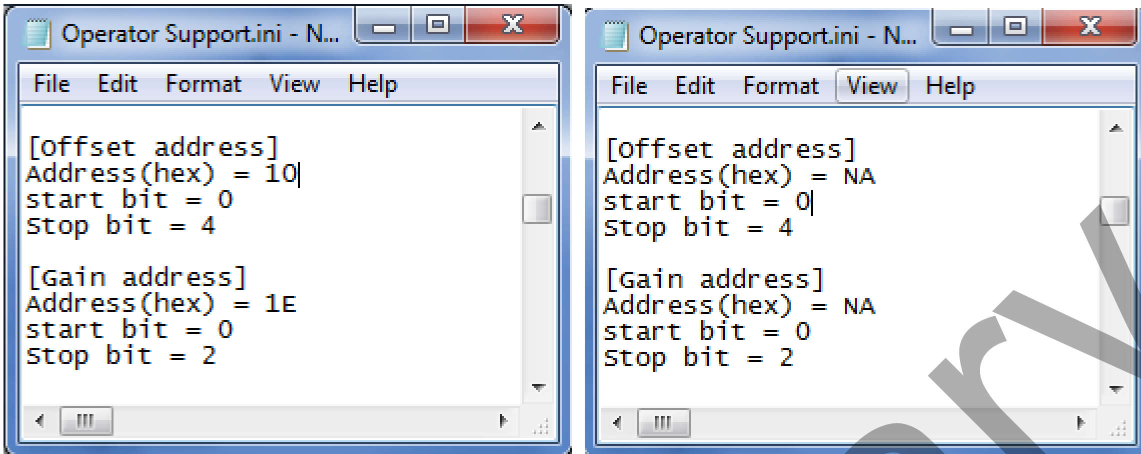


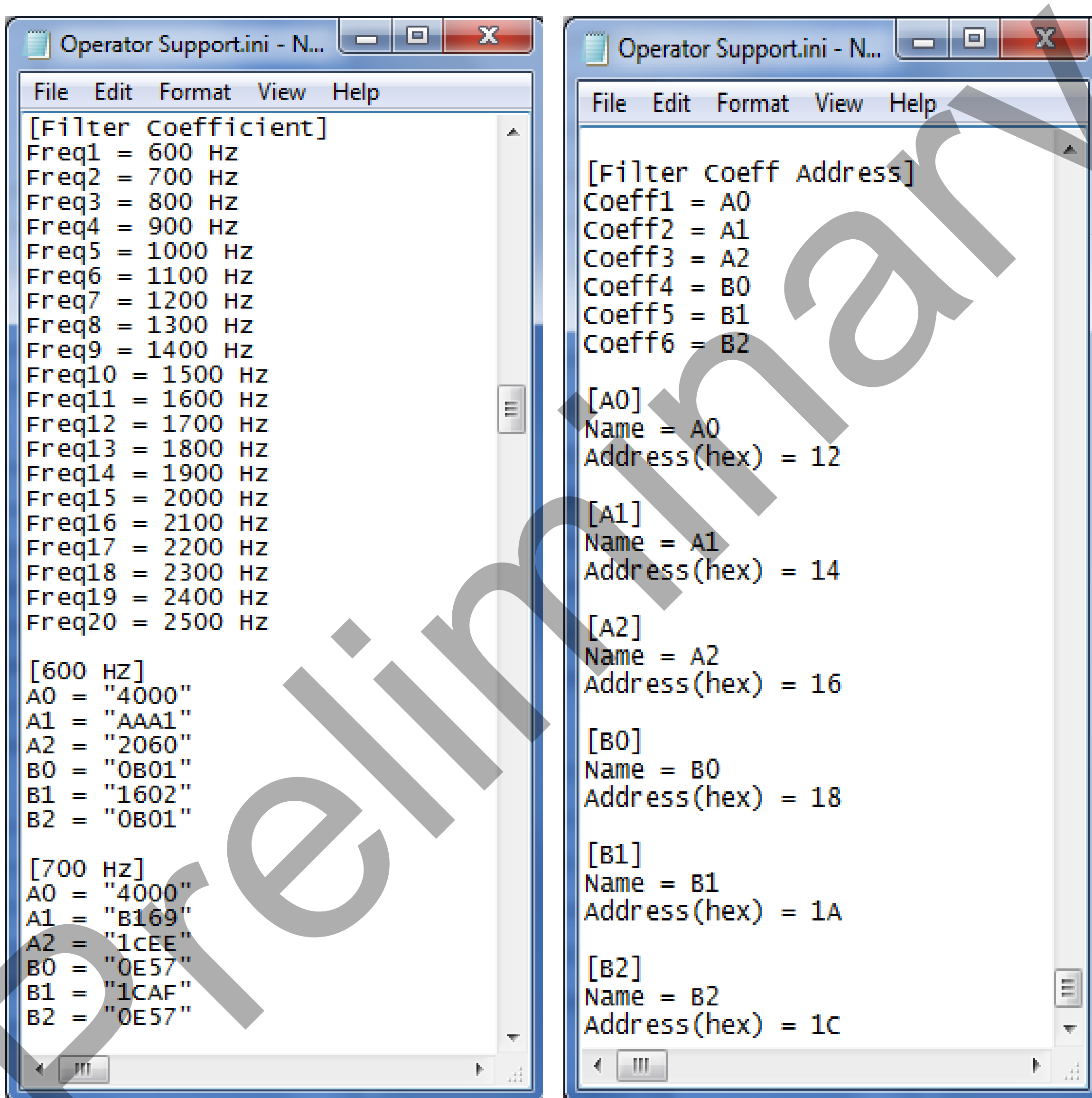
Fig 116: Configuration File – Gain and Offset Settings

- The Offset and Gain values to be used for the coefficient calculation is configured from the above mentioned sections (image) of the configuration file.
- The offset and gain value to be used for the calculation are read from specific EEPROM addresses (bit positions defined by Start and Stop bits) which has been mentioned in the configuration file as displayed in the above image.
- If either of the Addresses is mentioned as 'NA', then the default values of offset and gain will be considered and taken for calculation.
  - Offset Default Value = 0
  - Gain Default Value = 1

Preliminary

3.3. Calibration Coefficients

[The Filter Coefficient] section is used to add the filter coefficients for respective cut-off frequencies. The EEPROM addresses for each of the Coefficients are mentioned in a separate section.



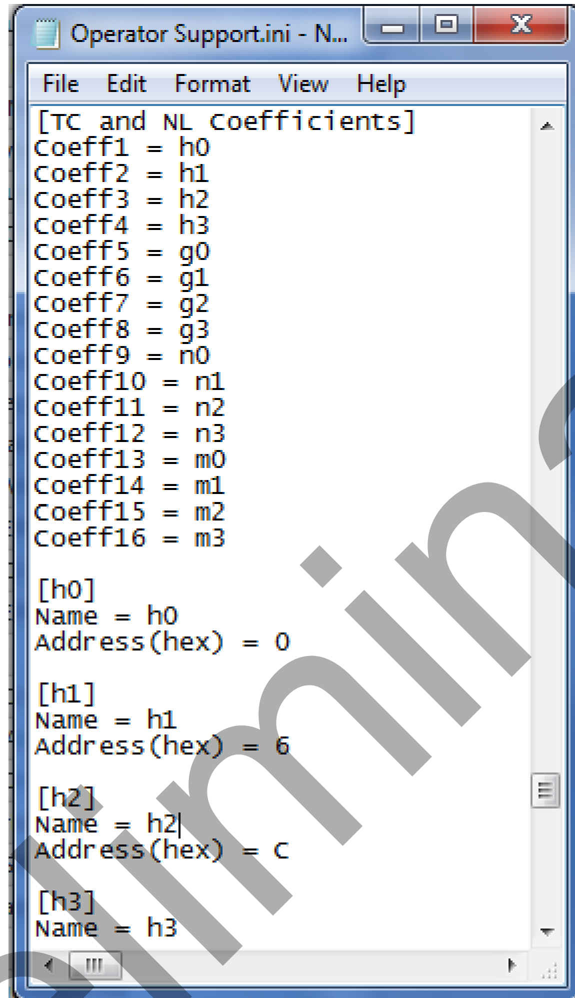
```
[Filter Coefficient]
Freq1 = 600 Hz
Freq2 = 700 Hz
Freq3 = 800 Hz
Freq4 = 900 Hz
Freq5 = 1000 Hz
Freq6 = 1100 Hz
Freq7 = 1200 Hz
Freq8 = 1300 Hz
Freq9 = 1400 Hz
Freq10 = 1500 Hz
Freq11 = 1600 Hz
Freq12 = 1700 Hz
Freq13 = 1800 Hz
Freq14 = 1900 Hz
Freq15 = 2000 Hz
Freq16 = 2100 Hz
Freq17 = 2200 Hz
Freq18 = 2300 Hz
Freq19 = 2400 Hz
Freq20 = 2500 Hz

[600 Hz]
A0 = "4000"
A1 = "AAA1"
A2 = "2060"
B0 = "0B01"
B1 = "1602"
B2 = "0B01"

[700 Hz]
A0 = "4000"
A1 = "B169"
A2 = "1CEE"
B0 = "0E57"
B1 = "1CAF"
B2 = "0E57"
```

```
[Filter Coeff Address]
Coeff1 = A0
Coeff2 = A1
Coeff3 = A2
Coeff4 = B0
Coeff5 = B1
Coeff6 = B2

[A0]
Name = A0
Address(hex) = 12

[A1]
Name = A1
Address(hex) = 14

[A2]
Name = A2
Address(hex) = 16

[B0]
Name = B0
Address(hex) = 18

[B1]
Name = B1
Address(hex) = 1A

[B2]
Name = B2
Address(hex) = 1C
```

Fig 117: Configuration File – Filter Coefficients

- To change the value of the coefficients in any of the cut-off frequencies, edit the corresponding values in the sections.

Preliminary


Operator Support.ini - N...
File Edit Format View Help

```
[TC and NL Coefficients]
Coeff1 = h0
Coeff2 = h1
Coeff3 = h2
Coeff4 = h3
Coeff5 = g0
Coeff6 = g1
Coeff7 = g2
Coeff8 = g3
Coeff9 = n0
Coeff10 = n1
Coeff11 = n2
Coeff12 = n3
Coeff13 = m0
Coeff14 = m1
Coeff15 = m2
Coeff16 = m3

[h0]
Name = h0
Address(hex) = 0

[h1]
Name = h1
Address(hex) = 6

[h2]
Name = h2
Address(hex) = C

[h3]
Name = h3
```

Fig 118: Configuration File – TC and NL Coefficients

- All the TC and NL Coefficients required for the available calibration mode have to be added in this section of the configuration file.
- The Address location for each of the coefficient can be added as displayed in the above image.

Preliminary

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2016, Texas Instruments Incorporated