

MAX11311PMB1 Peripheral Module and Munich (USB2PMB1#) Adapter Board Quick Start Guide

UG6233; Rev 0; 4/16

Abstract

This document explains how the MAX11311PMB1, which features the MAX11311 12-port programmable mixed-signal I/O device, interfaces with the Munich (USB2PMB1#) adapter board, and how to use the GUI.

Table of Contents

1. Required Equipment	2
2. Overview	2
3. Included Files	5
4. Procedure	5
5. Munich GUI.....	8
6. Examples	12
7. References	25
8. Trademarks.....	26

1. Required Equipment

- PC with Windows® OS (Windows XP®, Windows Vista®, Windows 7, Windows 8 and Windows 8.1) with one USB port
- MAX11311PMB1 board
- Munich (USB2PMB1#) board
- USB Type A to Mini B Cable
- Munich2 Software GUI

2. Overview

This document explains how the MAX11311PMB1# featuring the MAX11311 12-port programmable mixed-signal I/O device interfaces with the Munich (USB2PMB1#) adapter board and how to navigate through the tool.

The MAX11311PMB1 peripheral module provides the necessary hardware to interface the MAX11311 12-port device to any system that utilizes Pmod™-compatible expansion ports configurable for SPI communication (**Figure 1**). MAX11311/MAX11312 configuration software is a design tool to create the configuration bit stream, which is downloaded to the MAX11311 via the Munich adapter board (**Figure 2**).

Refer to the MAX11311 IC data sheet for detailed information regarding operation of the IC and to the USB2PMB1# adapter board data sheet for detailed information regarding the Munich board and GUI. Refer to the MAX11311/MAX11312 PIXI™ Configuration Software User Guide for detailed information using the design tool.

The Munich GUI and the USB2PMB1# adapter board are referred to as Munich throughout this document.

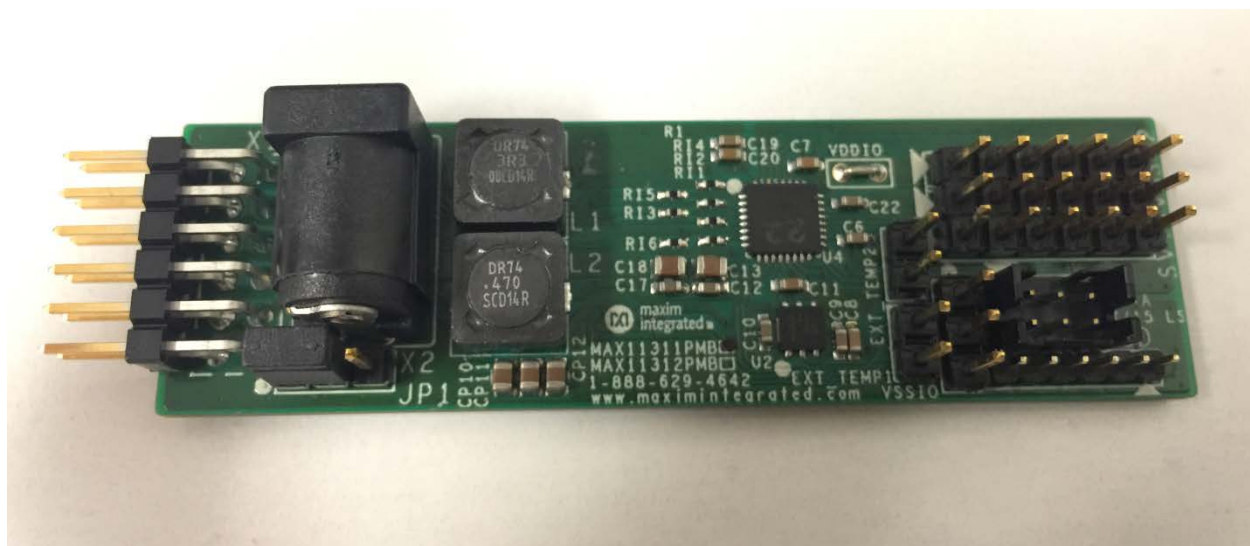
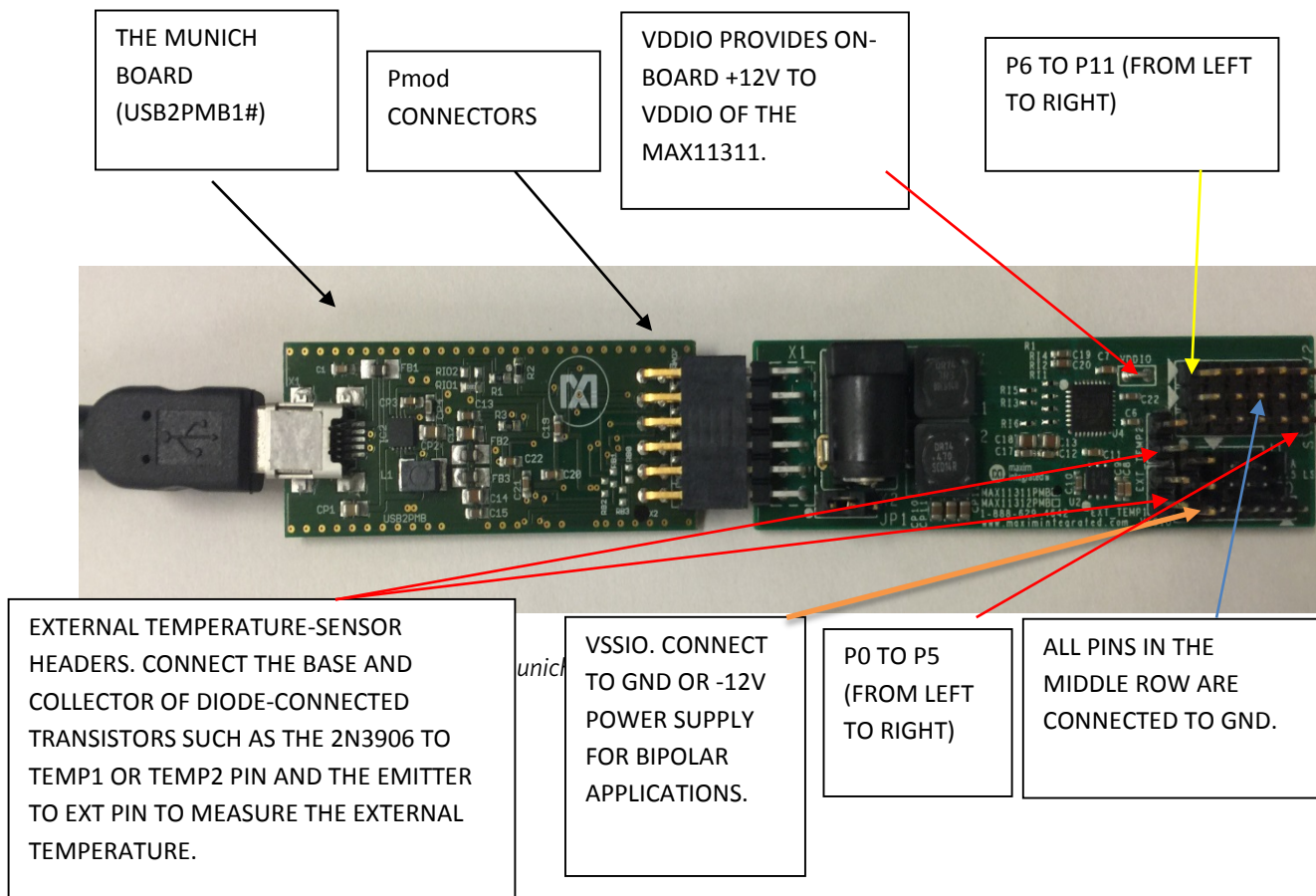


Figure 2. MAX11311PMB1 board.

The Munich adapter board is designed to supply power to external boards (i.e., MAX11300PMB1, MAX11311PMB1#, Santa Fe, and many more) through the connector X2. It is intended to provide power supply to the interface circuitry present in the connected board.

Figure 3 shows the pin configuration for the SPI-compatible connector at the USB2PMB1# board. The MAX11311PMB1 board itself generates positive voltages from 3.3V on pins 6 and 12 with an on-board DC-DC converter. The 3.3V supply is limited to ~1.5W, limiting power available for the MAX11311PMB1# if a higher-watt power supply is required and external power supply is used. Refer to the MAX11311PMB1# IC data sheet for details.

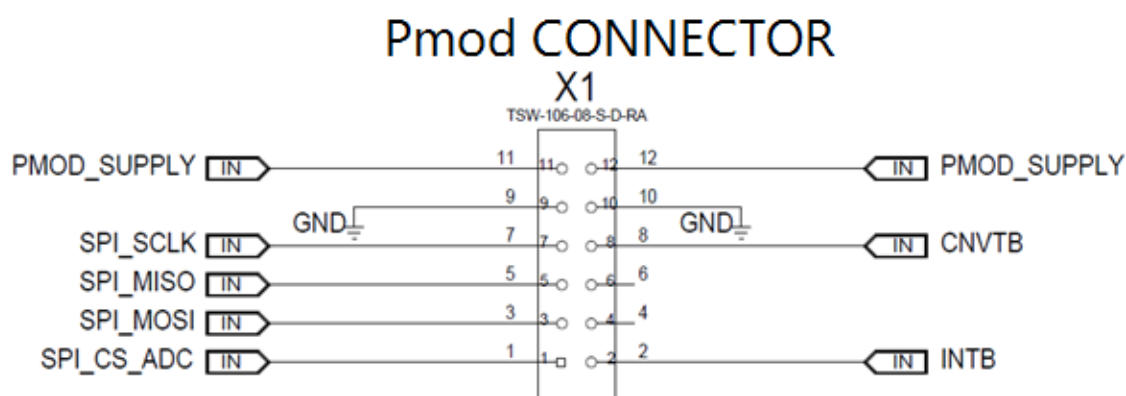


Figure 3. SPI connector inputs found on the MAX11311PMB1#.

Included files are the MAX11311 configuration software and Munich GUI (see **Figure 4**). The Munich GUI evaluates this MAX11311PMB1# and other Pmod designs from Maxim Integrated.

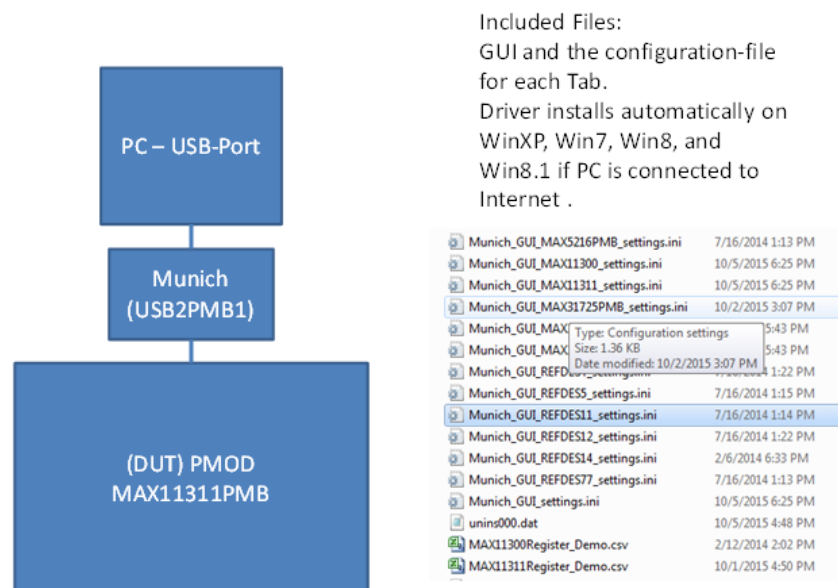


Figure 4. MAX11311PMB1#/Munich included files.

4. Procedure

- 1) Go to www.maximintegrated.com/evkitsoftware to download the most recent version of the Munich board software, **Munich GUI**. Double-click on the installation file **Munich GUISetupV2.11.exe** to install the GUI.
- 2) Connect the USB cable between the Munich board and the PC; the USB driver is installed automatically (internet connection required). Alternately, install the FTDI driver by double-clicking on the **CDM V2.08.30 WHQL Certified.exe** application file in the **FTDI Driver.zip** folder. See **Figure 5**, **Figure 6**, and **Figure 7**.

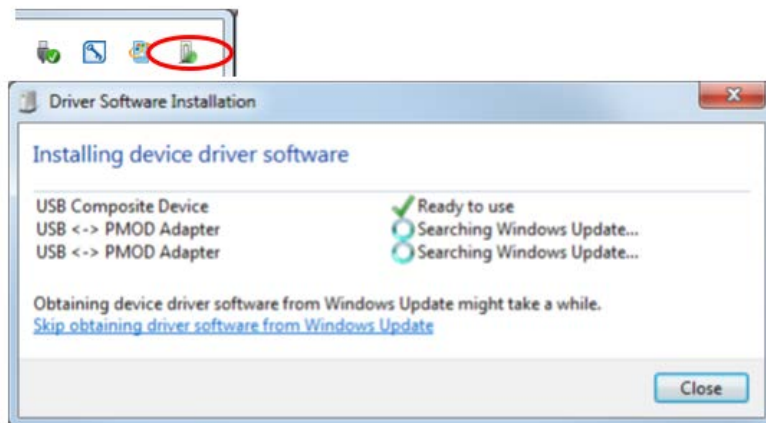


Figure 5. Driver installation for Munich board.



Figure 6. Munich GUI setup wizard.



Figure 7. Munich GUI at startup.

- 3) Ensure that the jumper settings for JP1 and VSSI0 power supplies are correct (JP1 and VSSI0 jumpers are connected from pin 1 to pin 2). Refer to the MAX11311PMB1# data sheet for further details.
- 4) Connect the Munich board's 2x6 pin right-angle connector to the MAX11311PMB1#'s 2x6 pin right-angle header.
- 5) Once the Munich board is connected with the MAX11311PMB1#, open the Munich GUI.exe (double-click) software.
- 6) Click on Device tab and select the MAX11311PMB1# as shown in **Figure 8**, the Munich GUI.

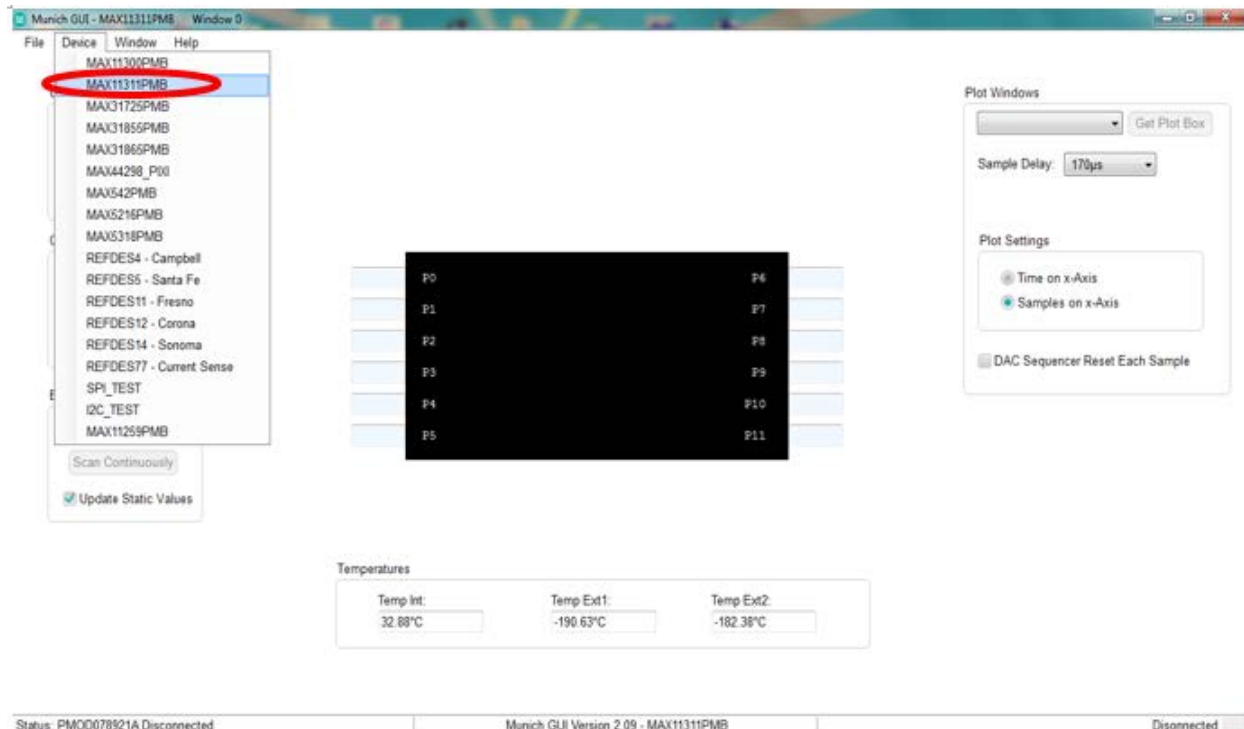


Figure 8. The Munich GUI.

- 7) Press the **Scan Adapters** pushbutton to scan the available Munich boards connected to the computer. This allows the user to test multiple companion driver boards at the same time. Each Munich board has a unique ID that the software determines.

Scan and Select:

Figure 9 depicts the scanning and available Munich devices connected. The software will automatically identify the device: PMOD078921A.

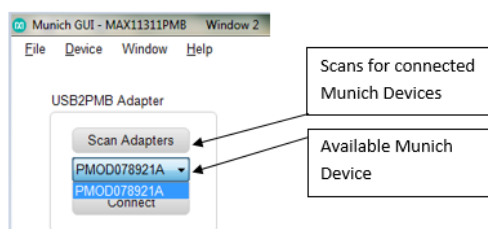


Figure 9. Connect Pmod adapter.

- 8) Within the tab sheet, press the **Connect** button and verify that the button changes its text to **Disconnect** and the status bar at the bottom indicates that the companion board is connected. See Figure 10.



Figure 10. MAX11311 connected to Munich GUI.

5. Munich GUI

The Munich GUI contains the internal block diagram of the MAX11311 device showing the 12 programmable I/Os with their programmed setting.

At startup, the MAX11311 device is un-programmed. To program the device, click on **Chip Configuration File | Read from File** and select a .csv file that was created with the MAX11311 Configuration Software tool. In the example provided, the **MAX11311Register_Demo.csv** was selected. Next, write this 'configuration bitstream' to the MAX11311 device by clicking on **Chip Configuration File | Write to Device** as shown in Figure 11.



Figure 11. Configuration example.

5.1 Write to Device

Click this button to transfer the configuration bit stream to the MAX11311. After updating the device, use the Scan command to read the status and the port name is updated to reflect the configuration. In this case, P5_ADC shown in **Figure 11** denotes that Port 5 is configured as an ADC.

The functions supported by the MAX11311 GUI are as follows:

- Configuration
- Get Predefined Waveform Files
- Choose Sample Delay and Single or Continue Scan Mode
- View/Save Plots
- Plot View Options

Once the MAX11311 device is configured, click on **Scan Continuously**. The GUI then exercises each port with the programmed function.

5.2 Part Configuration

If the port is configured as:

- General-Purpose Input Port (GPI):** The user-defined logic level is read from the port and written to the appropriate port box.
- General-Purpose Output Port (GPO):** The user-defined logic level defined in the port text box is written to the port (use either "0" / "1" or "high" / "low").
- DAC Analog Output:** The value written to the text box is applied to the port. For DAC analog outputs, it is also possible to apply a complete waveform. To do that, select a target DAC-port, and to apply a waveform, click on **Get Plot Box** on the top right. A separate window opens, and a comma separated file (.adc) that contains hex or decimal numbers can be read with up to 1024

samples (see **Figure 12**). There are a few waveform examples provided for sawtooth and sine and triangle waveforms; these files have the suffix .adc*. Note that once a plot box for the port is open, the text box in the GUI main window is ignored.

- d. **ADC Analog Input:** the hexadecimal number in the text box corresponds to the voltage applied to the pin. Note that the ADC input type and the input range is selected in the configuration tool. As with the DAC configuration, you can also open a plot box by selecting the appropriate pin and clicking on **Get Plot Box** on the top right. If a plot box is open, the plot box shows the waveform applied to the ADC input pin.
- e. **Differential ADC Input:** The MAX11311 can be configured for differential ADC mode using two ports. The operation of the GUI is the same as in step d.
- f. **Level Translator:** The IC acts correctly, but the GUI would only show static values for the input and the output.

**Although the files have suffix .adc, the content is a simple hex value that can either be a file saved from an ADC analog input or as a file that can be opened and written to a DAC to provide an analog output.*

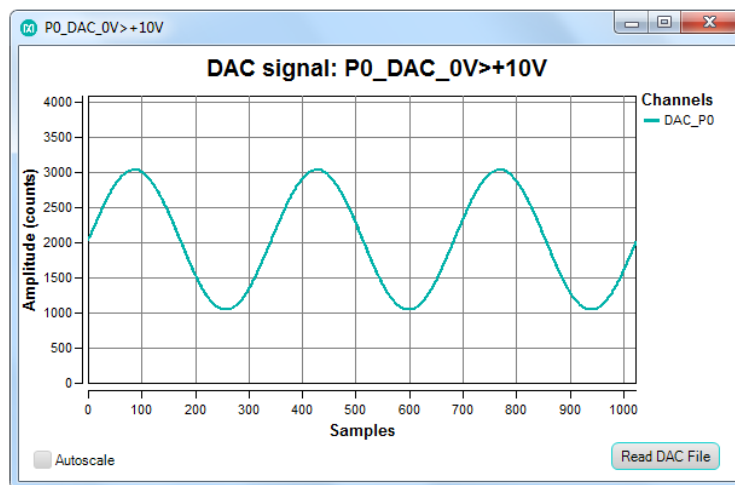


Figure 12. DAC plot box.

The GUI also displays the three temperature-sensor measurements in degrees Celsius (°C). The internal device temperature measures the IC die temperature. For the external temperature measurements, add diode-connected transistor sensors such as a 2N3906 to each channel.

5.3 Sampling

The flexibility of MAX11311 supports different modes of operation including different sampling schemes:

- a. **Scan Once:** In this mode the software reads the status of each input port a single time and displays the value of each port.
- b. **Scan Continuously:** In this mode the software continually loops, reading all input ports and updating the display to show the value of each port.
- c. **Sample Delay:** This drop-down menu allows the user to select up to 1030μs delay. This delay is enabled by the GUI between each complete sequence of ADC read. See **Figure 13**.

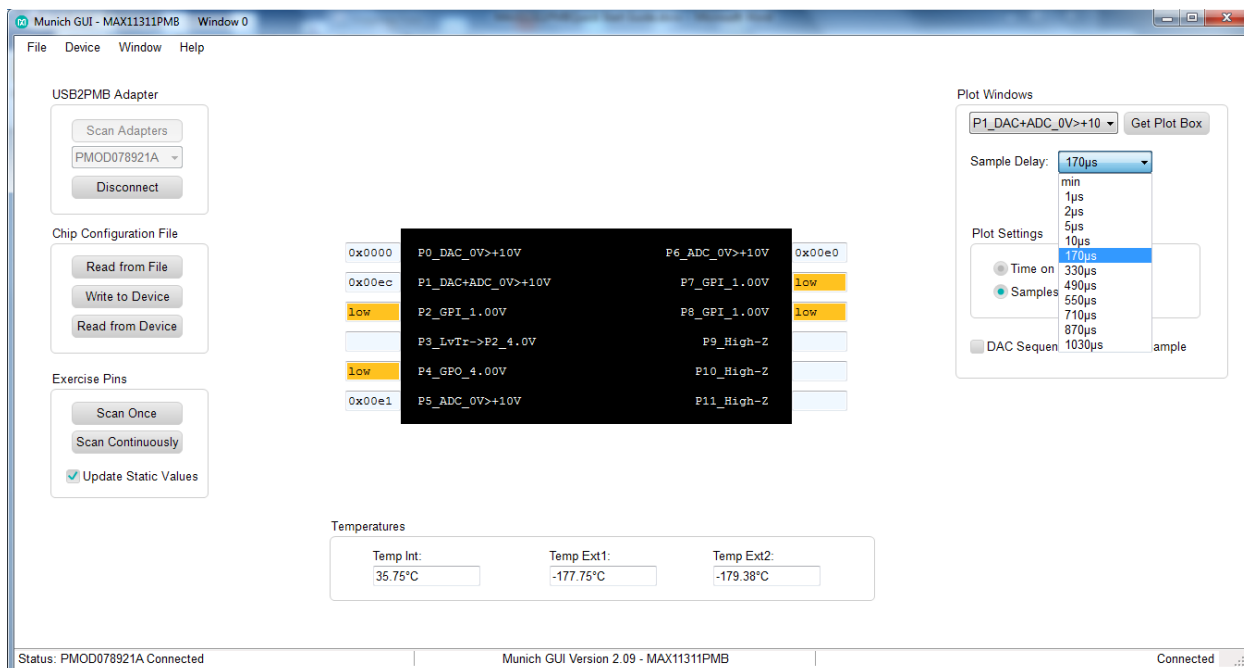


Figure 13. Sample delay drop-down menu.

5.4 DAC Sequencer Reset Each Sample

The DAC ports can be updated in different ways, either following a default sequence, or as soon as new data is received, or using one of two data values (which can be programmed to all DAC ports). This button when selected enables the DAC to be updated as soon as new data is received at the DAC Data register. See **Figure 14**.

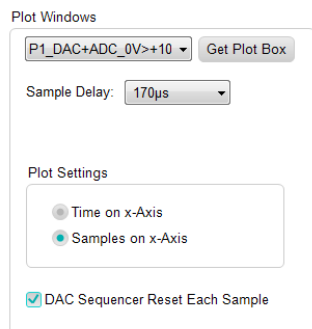


Figure 14. DAC sequencer reset.

6. Examples

The MAX11311PMB1# software includes a number of waveform plots and examples for configuring and testing the MAX11311.

Figure 15 lists the files provided with the software tool. The files with suffix .adc are predefined waveform plots that can be used as input values to ports configured in DAC mode. The files with suffix .csv are example configurations showing different ways the MAX11311 can be configured. The files with suffix .mpix are design files from MAX11311 configuration software.

Name	Date modified	Type	Size
 ideal_sine_1000_4x.adc	2/3/2014 11:26 AM	ADC File	8 KB
 ideal_sine_1000_3x.adc	2/3/2014 11:27 AM	ADC File	8 KB
 ideal_sine_1000_2x.adc	2/3/2014 11:27 AM	ADC File	8 KB
 ideal_sine_1000_1x.adc	2/3/2014 11:27 AM	ADC File	8 KB
 ideal_sine_1500_1x.adc	2/3/2014 11:27 AM	ADC File	8 KB
 ideal_sine_1500_2x.adc	2/3/2014 11:28 AM	ADC File	8 KB
 ideal_sine_1500_3x.adc	2/3/2014 11:28 AM	ADC File	8 KB
 ideal_sine_1500_4x.adc	2/3/2014 11:28 AM	ADC File	8 KB
 ideal_sawtooth_left_1000_1x.adc	2/3/2014 4:09 PM	ADC File	8 KB
 ideal_sawtooth_left_1000_2x.adc	2/3/2014 4:10 PM	ADC File	8 KB
 ideal_sawtooth_left_1000_3x.adc	2/3/2014 4:10 PM	ADC File	8 KB
 ideal_sawtooth_left_1000_4x.adc	2/3/2014 4:10 PM	ADC File	8 KB

Figure 15. MAX11311PMB1# files.

The Munich GUI is provided with an example configuration file for the examples in **Figure 15**. To create a configuration file, a separate tool is needed: MAX11311/12 Configuration Software.

A step-by-step guide to use example register file is as follows:

1. Ensure the Munich board is connected to the MAX11311PMB1# board and the USB is connected to a PC through a USB Type A to Mini B Cable as shown in **Figure 16**.
2. Ensure the MAX11311PMB tab is opened in the Munich GUI and the adapter number is shown in top-left corner. If you have multiple USB2PMB1# adapters, verify the serial numbers.

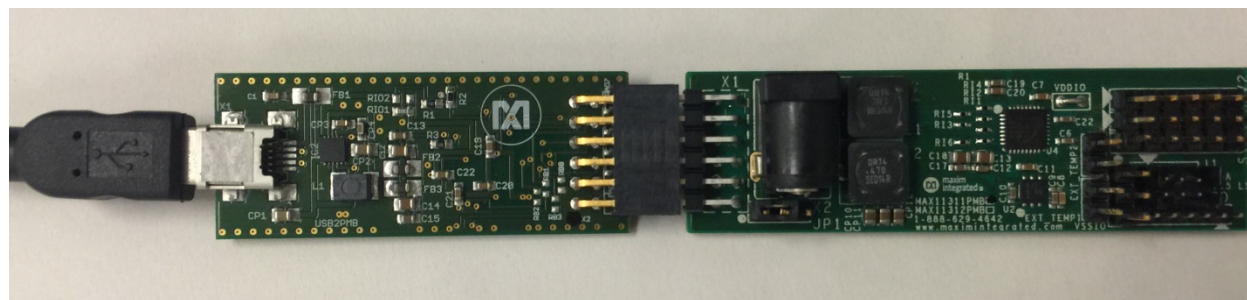


Figure 16. MAX11311PMB1#/Munich connected to PC through a USB cable.

3. Click the Connect button. The Munich GUI then reads the current configuration from the connected MAX11311PMB1# and displays the pin description for each pin. If the device is powered up, every pin is high-Z (high impedance) as shown in **Figure 17**.



Figure 17. Munich GUI at startup.

4. Under **Configuration**, click on **Read from File** and select the register file MAX11311Register_Demo.csv as shown in **Figure 18**.

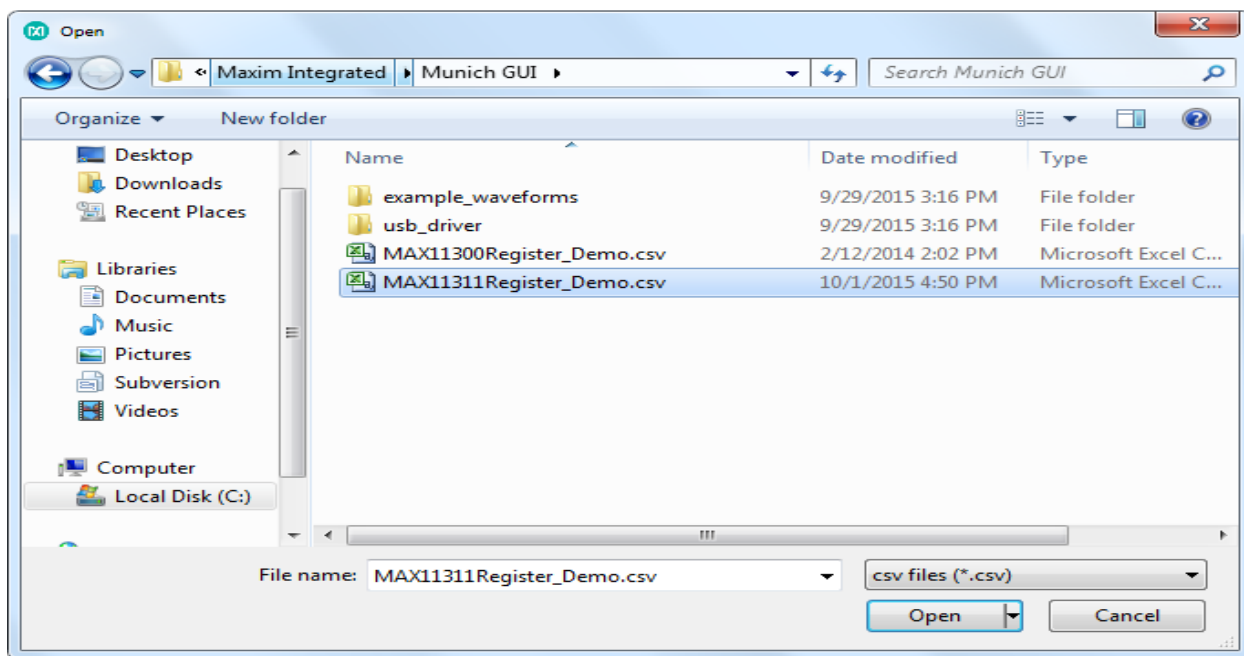


Figure 18. Register files.

5. The Munich GUI now shows all pin descriptions in the GUI according to the register file. But the GUI does not immediately write it to the device, allowing the user opportunity to double-check. This step is only important if something is already connected to the MAX11311 ports as a DAC can provide up to $\pm 10V$ and potentially destroy the external circuitry. Also, do not connect 2-DAC ports together (shorted) and set to different voltage levels. In addition, GPI ports can only support levels between 0V and 5V. Therefore, do not connect a DAC to a GPI with a voltage outside of the allowed range. A Confirmation window appears as depicted in **Figure 19**.

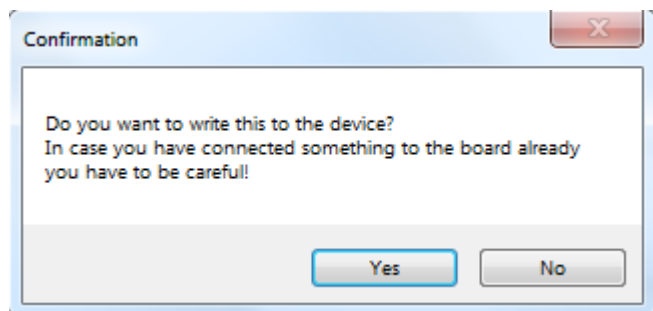


Figure 19. Confirmation window.

Select **Yes** to write the new configuration to the MAX11311. If No is chosen, the GUI does not write to the device, but reads the configuration from the device and displays that again.

6. After the Example Register File is written, the screen on the GUI appears as shown in **Figure 20**.



Figure 20. Pin configuration.

Pin configurations in this example are as follows:

- P0: DAC Range 0V to +10V
- P1: DAC + ADC Monitoring Range 0V to +10V
- P3: Unidirectional Level Translator Input from P2, Output Level 5.0V
- P4: GPO General Purpose Digital Output, Level 4.0V
- P5: ADC (Single-Ended ADC) Range 0V to +10V
- P6: ADC (Single-Ended ADC) Range 0V to +10V
- P7: GPI General-Purpose Digital Input, Threshold 1.0V
- P8: GPI General-Purpose Digital Input, Threshold 1.0V
- P9, P10, P11: High-Z (not used)

7. Hover the mouse cursor over the abbreviated pin description to obtain the detailed description as shown in **Figure 21**.

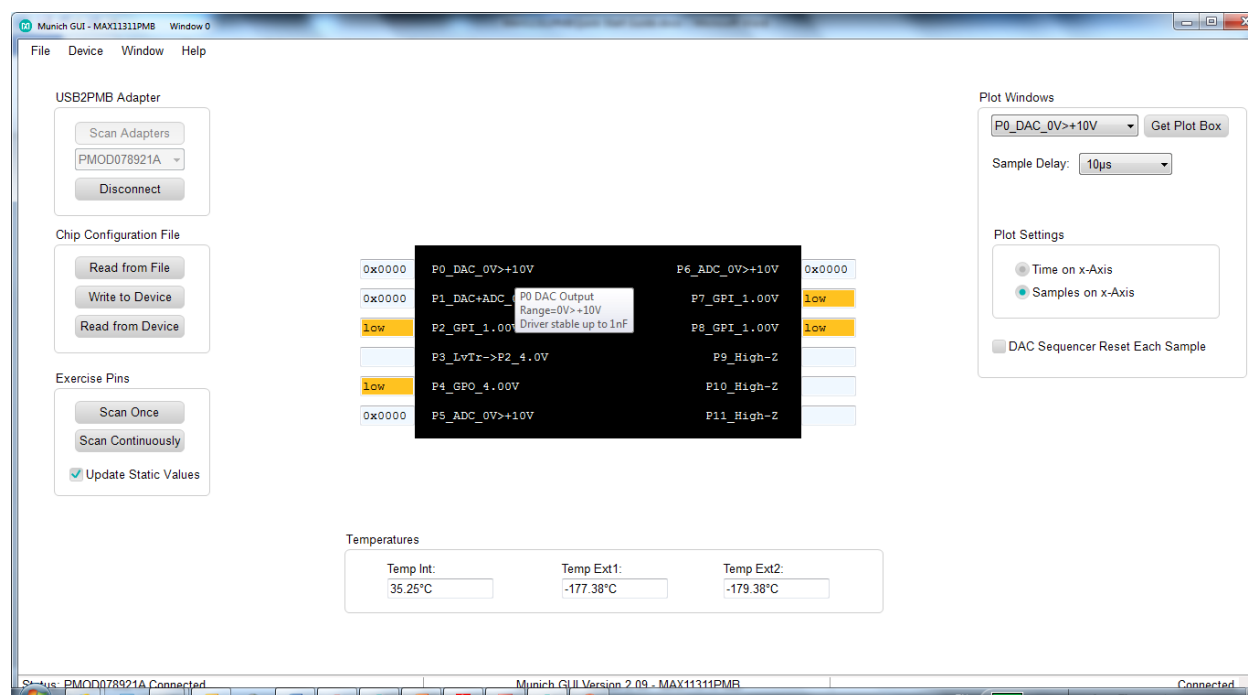


Figure 21. Detailed port description.

8. Select the **Scan Once** button under Exercise Pins. The Munich GUI then scans through all 20 PIXI ports and executes its function. For DAC, it writes the value in the port box to the DAC; for ADC, it reads the ADC and displays the received value into the port box, similar to the GPIs and GPOs. Only switches and level translators are independent, and the Munich GUI does not make any changes to them. The **Scan Continuously** button performs the same function as the **Scan Once** button, but it repeats the scanning process until Stop Scanning is pressed.

To see results in real time, click on **Scan Continuously**. The Scan Continuously button then changes to the Stop Scanning button, and the Scanning Progress bar appears. All inputs are continuously updated (i.e., GPIs, ADCs, temperature sensors). See **Figure 22**.

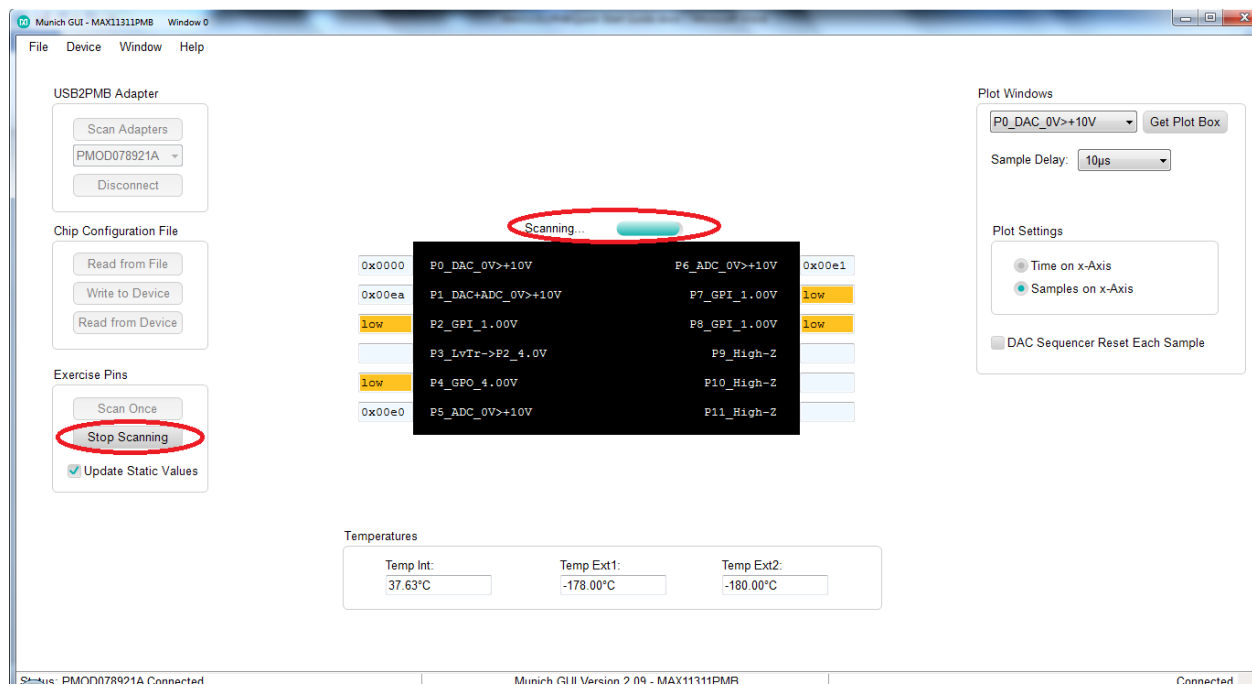


Figure 22. Continuous scanning mode.

The following examples illustrate step-by-step instructions to:

- Write a waveform to a DAC and sample it back with an ADC.
- Demonstrate the behavior of a DAC with ADC monitoring.
- Generate PWM with potentiometer-adjustable pulse width.

6.1 Writing a Waveform to a DAC and Sampling It Back with an ADC

1. Under Plot Windows in the drop-down menu at the top right, select **P0_DAC_0V>+10V** and Get plot box. A small plot window appears on the screen as shown in **Figure 23**. Click **Read DAC File** and select an analog data point file called **ideal_sine_1000_3x.adc**.

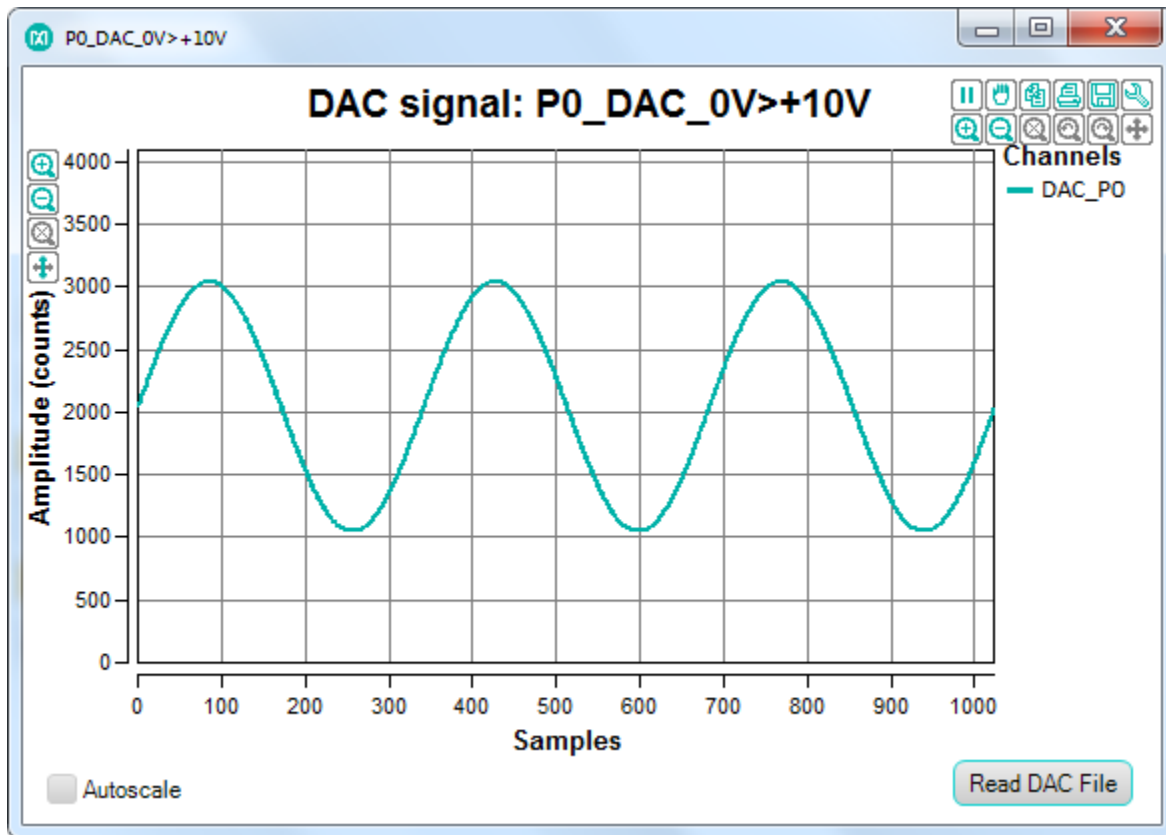


Figure 23. P0_DAC_0V>+10V plot window.

The plot window now shows a sine wave, and the entire plot has 1024 data points. While **Continuous Scan** is in progress, this waveform is written to the DAC and the speed depends on how many DAC and ADC pins are available and how many plot windows are open.

2. In the drop-down menu at the top right, select **P5_ADC_0>+10V** and click the **Get plot box** button. A plot window opens and shows the analog level on the ADC as shown in **Figure 24**.

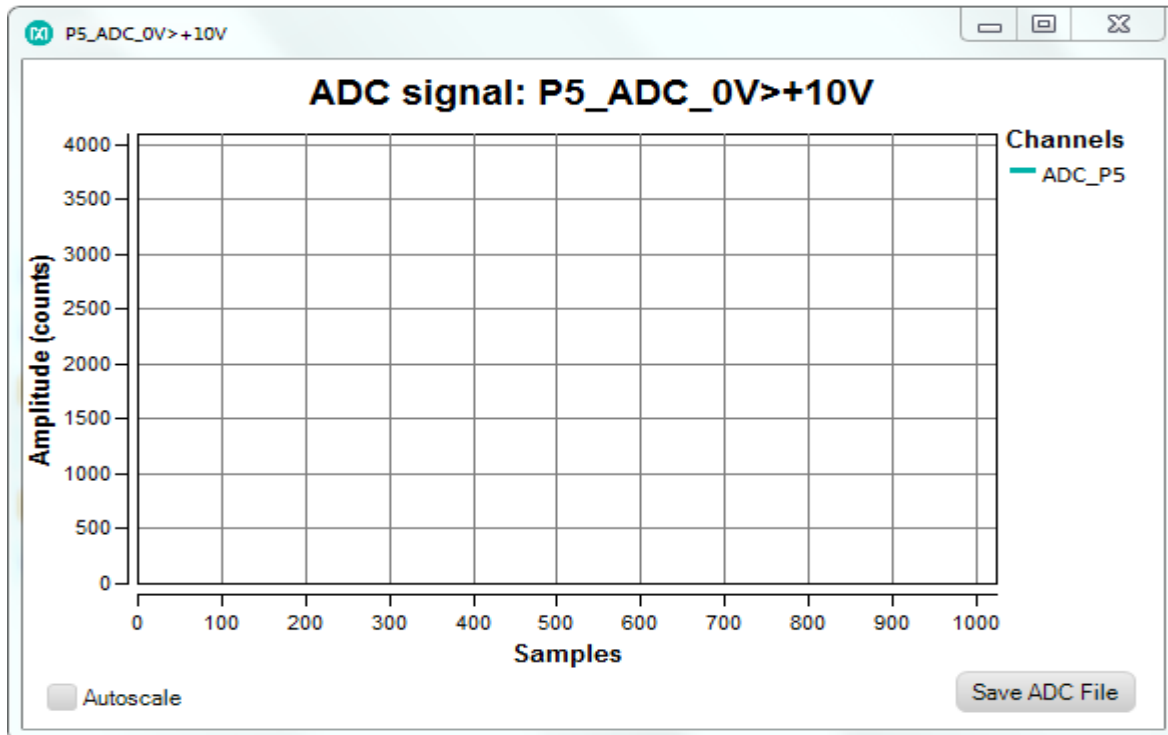


Figure 24. P5_ADC_0>+10V plot window.

3. Use a jumper wire to connect P0 to P5 as shown in **Figure 25**.

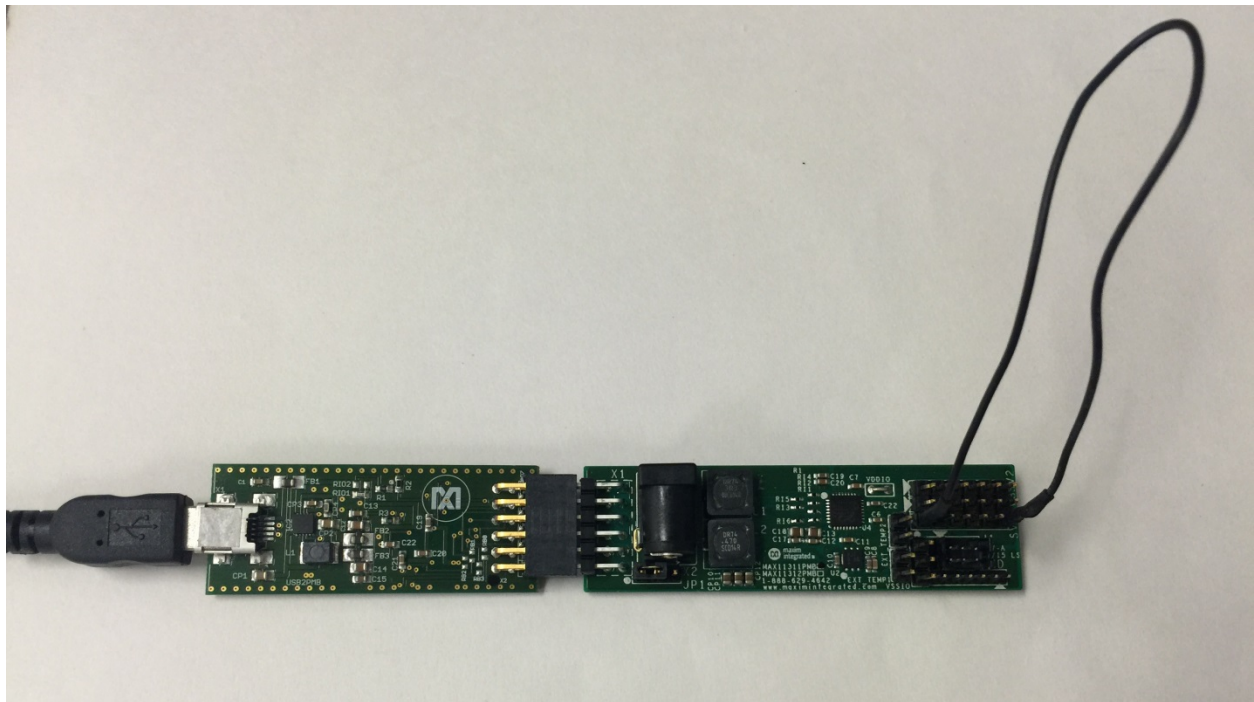


Figure 25. P0 connected to P5.

4. After connecting the DAC to the ADC, observe a sine-wave-like-shaped waveform in the ADC plot window as shown in **Figure 26**.

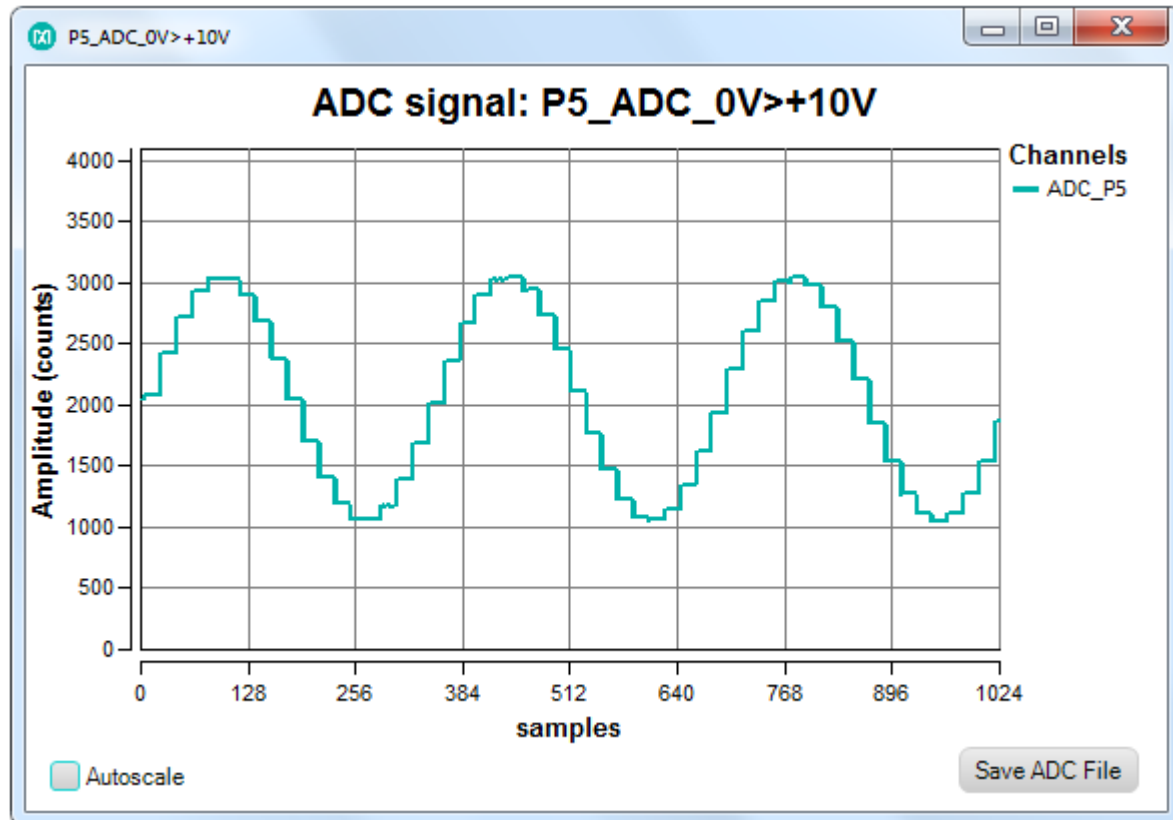


Figure 26. Sine-wave-like waveform without sample delay.

The waveform does not appear as smooth as the waveform applied in the DAC window. The reason is that the DAC can sample 20ksps only, but the ADC can sample 400ksps. Therefore, multiple data points are written to the DAC before the DAC can accept a new data point. To allow ample time for the DAC to settle before writing a new value, insert a sample delay between each sample by selecting 10 μ s or more in the Sample delay drop-down menu.

5. After adding 170 μ s sample delay, the ADC shows the same waveform as we apply to the DAC. See **Figure 27**.

6. Keep the plot windows open.

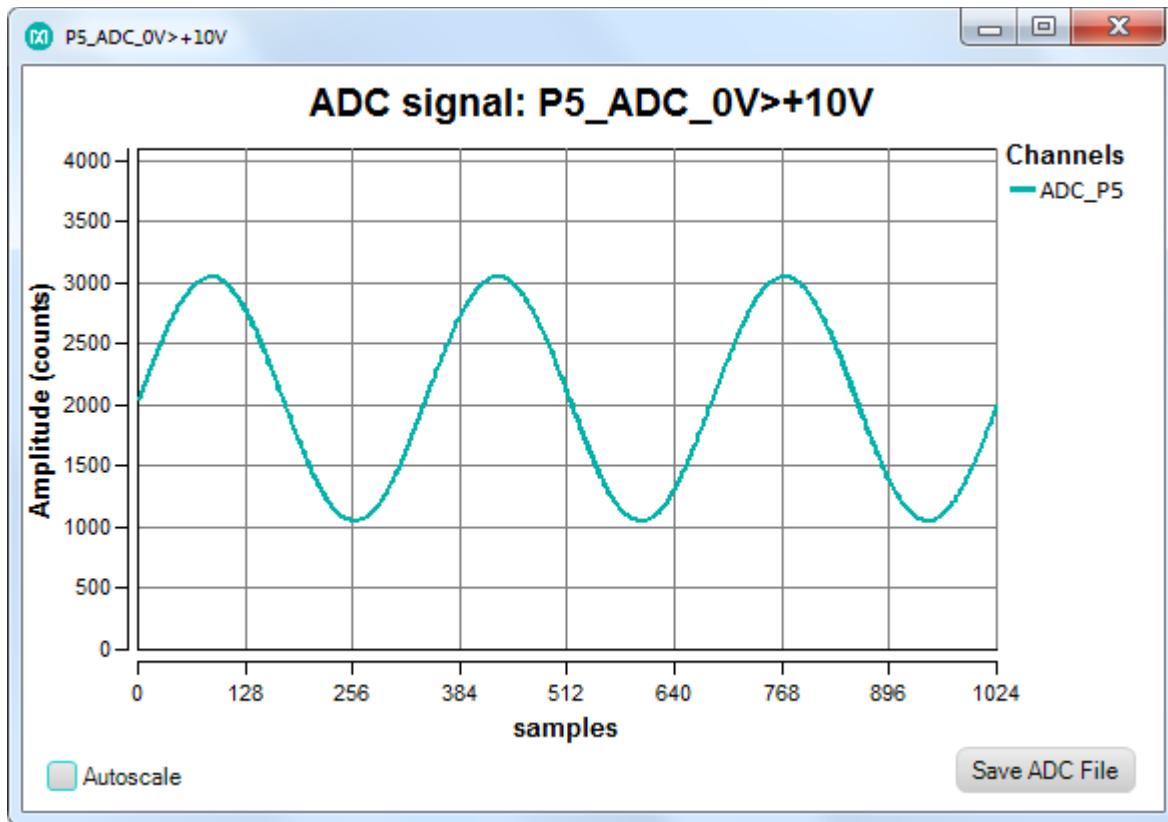


Figure 27. Sine wave with 170 μ s sample delay.

6.2 Demonstrate the Behavior of a DAC with ADC Monitoring

In this example, a potentiometer is connected between ADC and DAC to adjust the amplitude.

See **Figure 28** for the potentiometer schematic connection. Make sure that (1) is connected to the DAC output of the MAX11311 Port 0, (2) is connected to GND (middle row), and (3) is connected to the ADC MAX11311 Port 5. See Figure 29 for the setup.

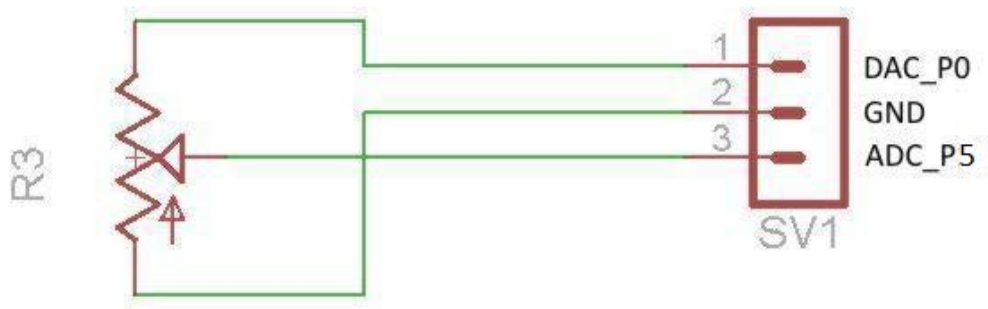


Figure 28. Potentiometer schematic connection to MAX11311PMB1# ports.

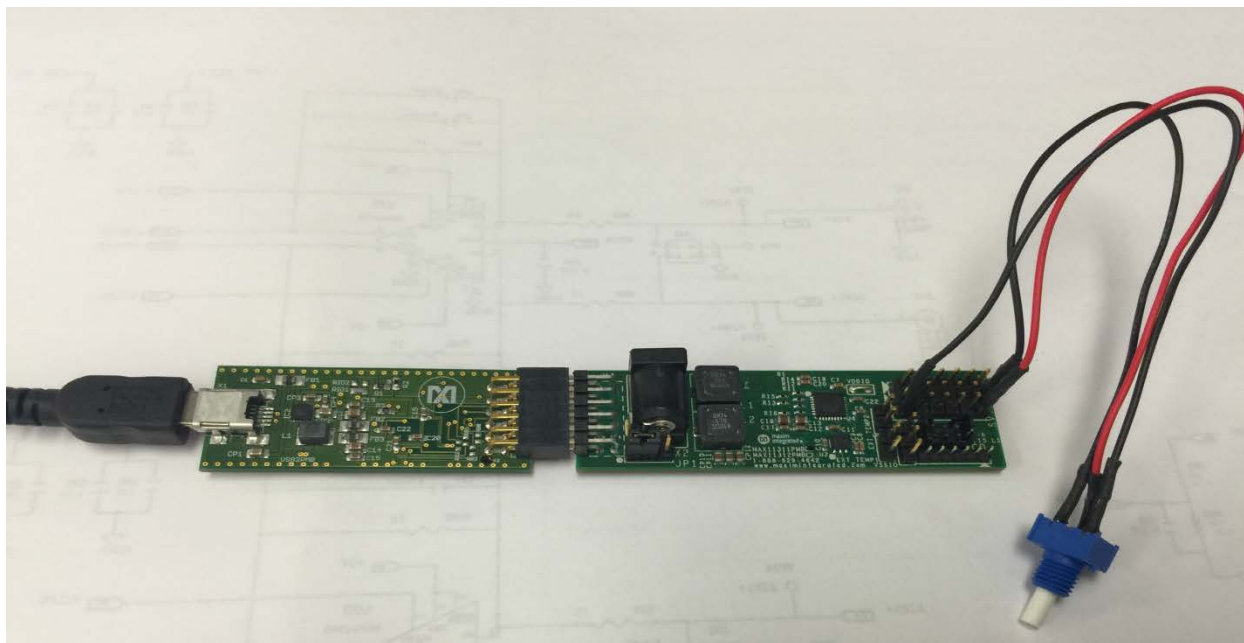


Figure 29. Potentiometer setup for demonstrating the behavior of a DAC with ADC monitoring.

1. Adjust the potentiometer and observe the amplitude received at the ADC changes as the potentiometer is varied.
2. In drop-down menu on the top right, select **P1_DAC+ADC_0>+10V** and select the Get plot box button. Another plot window opens and displays two waveforms: one in teal and one in purple. The teal waveform is for ADC, and the purple waveform is for DAC. Click on **Read DAC File** and open the **ideal_triangle_1000_3x.adc** file. The purple DAC waveform shows a triangle as driven into the DAC. The teal plot shows the waveform as sampled with the ADC on that same pin. Zoom in or make the plot window full screen to see the difference between the ADC and DAC plots, which is mostly due to timing. DAC allows current up to ~20mA; if more current is drawn, the DAC voltage drops. This happens if the DAC has an output voltage of 10V with a load resistor of 500 Ω or less.

6.3 Generate PWM with Potentiometer-Adjustable Pulse-Width

In this example, a sawtooth waveform is generated and fed to a comparator (we will use a level translator). The output of the comparator is high if the signal is above the threshold or low if the signal is below the threshold (same case for a level translator). As the position of the sawtooth waveform is moved up or down when the potentiometer is adjusted, the width of the curve above the threshold is wider or thinner. See **Figure 30** and **Figure 31**.

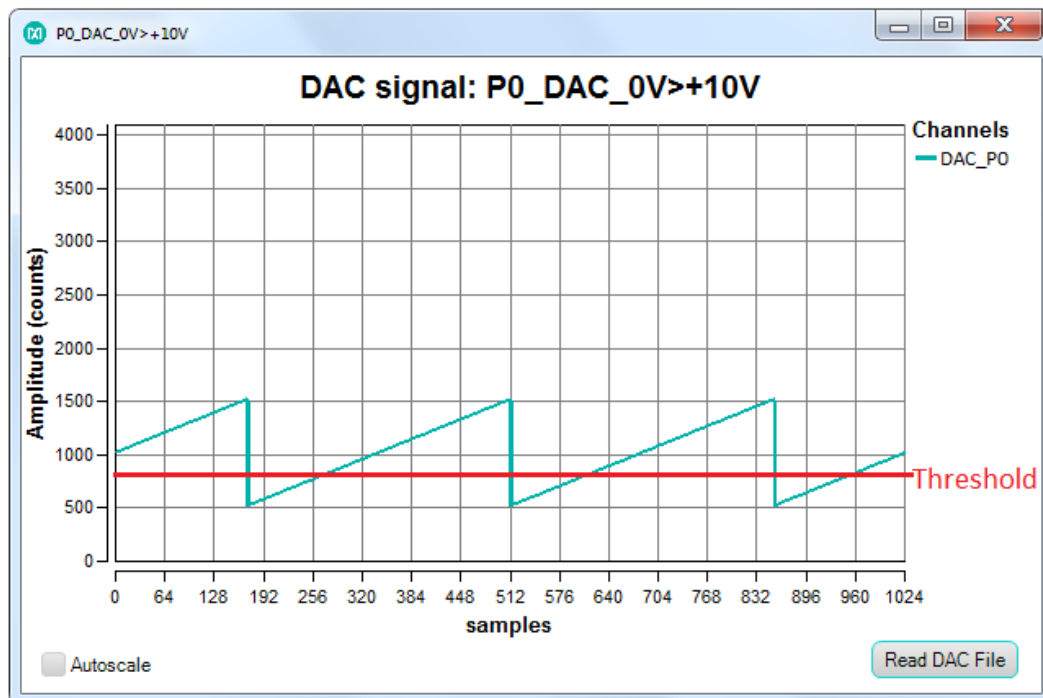


Figure 30. A sawtooth waveform shown is generated and applied to a level translator, resulting in a PWM as depicted in Figure 31.

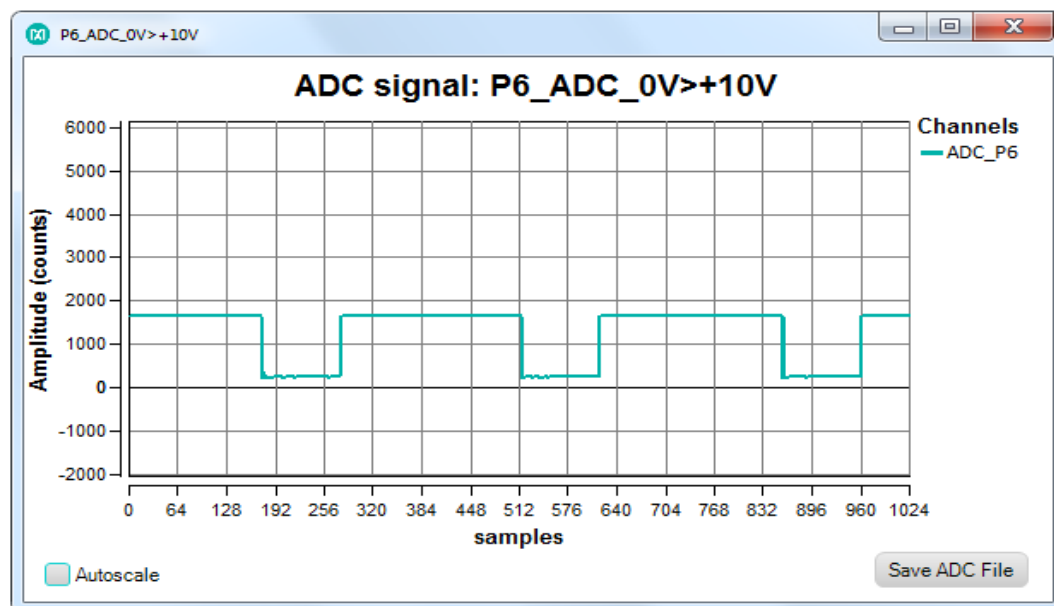


Figure 31. The width of a PWM is adjusted by a potentiometer.

1. Connect a potentiometer cable assembly to Port 0 (black jumper wire), Port 2 (red jumper wire), and GND (black jumper wire). Connect Port 3 to Port 7 (red jumper wire) as shown in **Figure 32**.

In the drop-down menu on the top right, select **P0_DAC_0V>+10V** and click the Get plot box button. Another plot window opens for DAC on P0. Click on **Read DAC File** in the newly opened P0_DAC window and open the analog data point file **ideal_sawtooth_right_1000_3x.adc**.

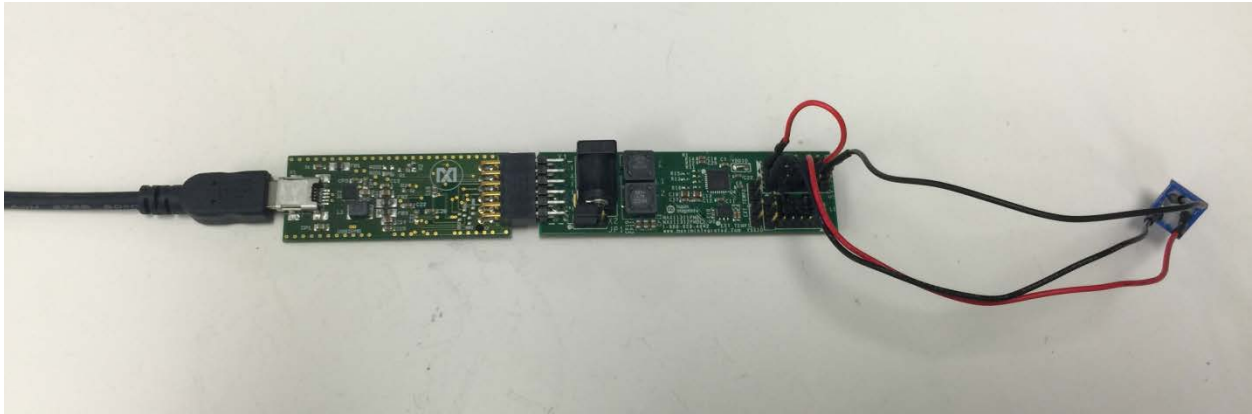


Figure 32. A potentiometer is connected to ports 0, 2, and GND.

2. Click P3_LvTr->P4_4.0V as shown in **Figure 33** below, a separate window of pin description for Pin3 will pop up as shown in **Figure 34**, under the associated pin tap, choose P4 (this is due to the limitation of GUI). The real pin number is listed on the pin description.

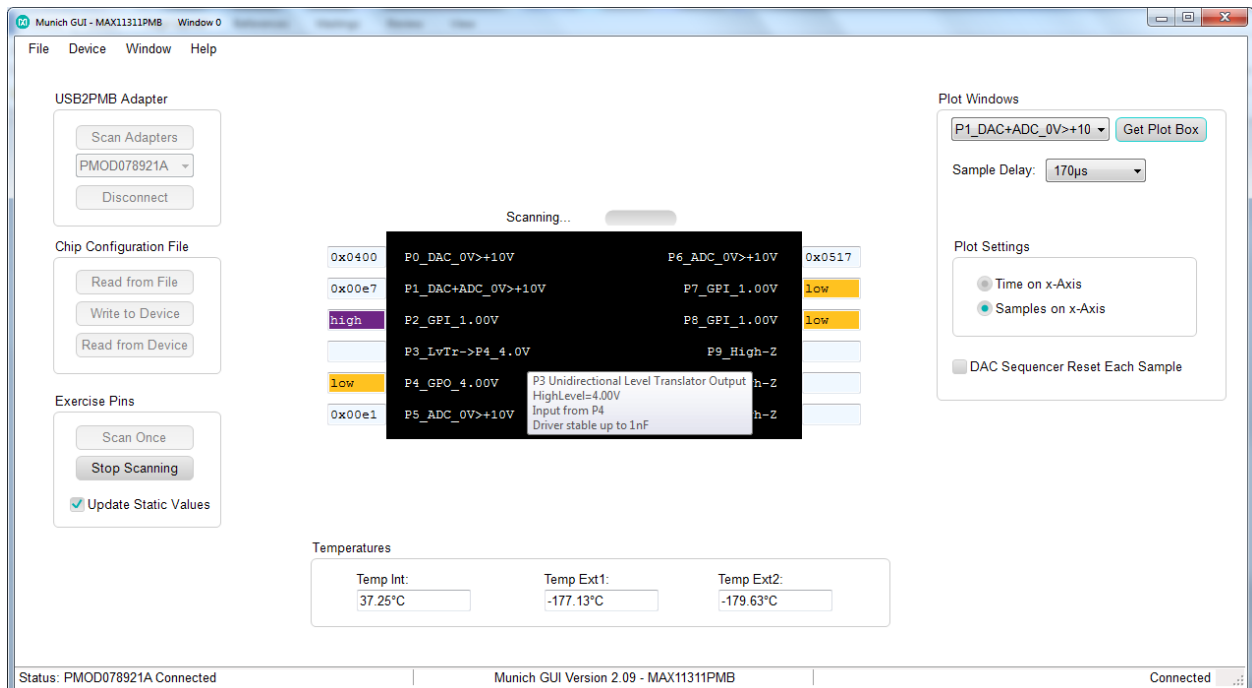


Figure 33. Click on Output of level translator.

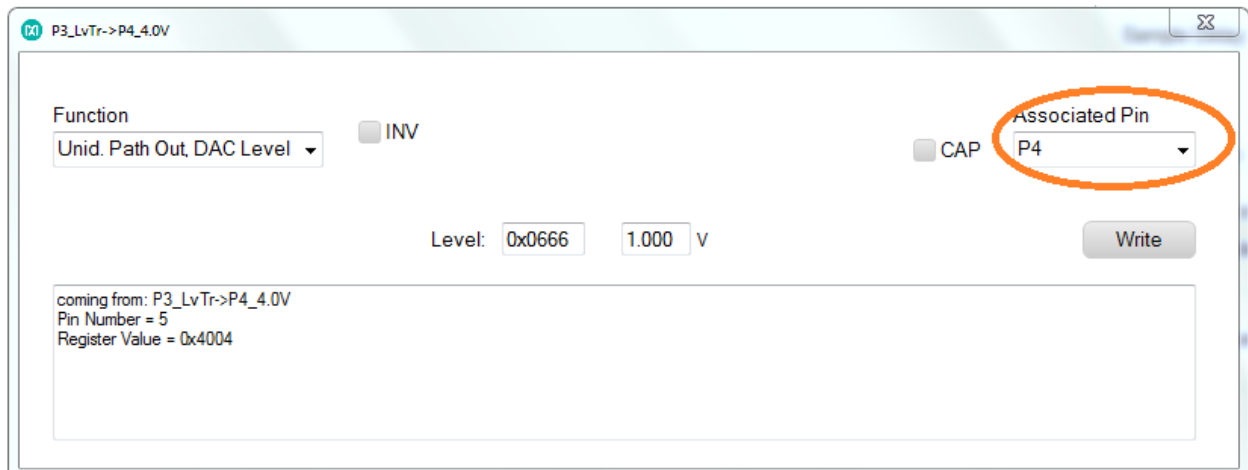


Figure 34. Pin description of P3.

3. In the drop-down menu on the top right, select **P6_ADC_0V>+10V** and click the Get plot box button. Another plot window opens for ADC on P6. Observe PWM signal in the ADC plot window. Adjust the pulse-width by turning the potentiometer to the low and high positions and observe the width of the PWM waveform as shown in **Figure 35** and **Figure 36**.

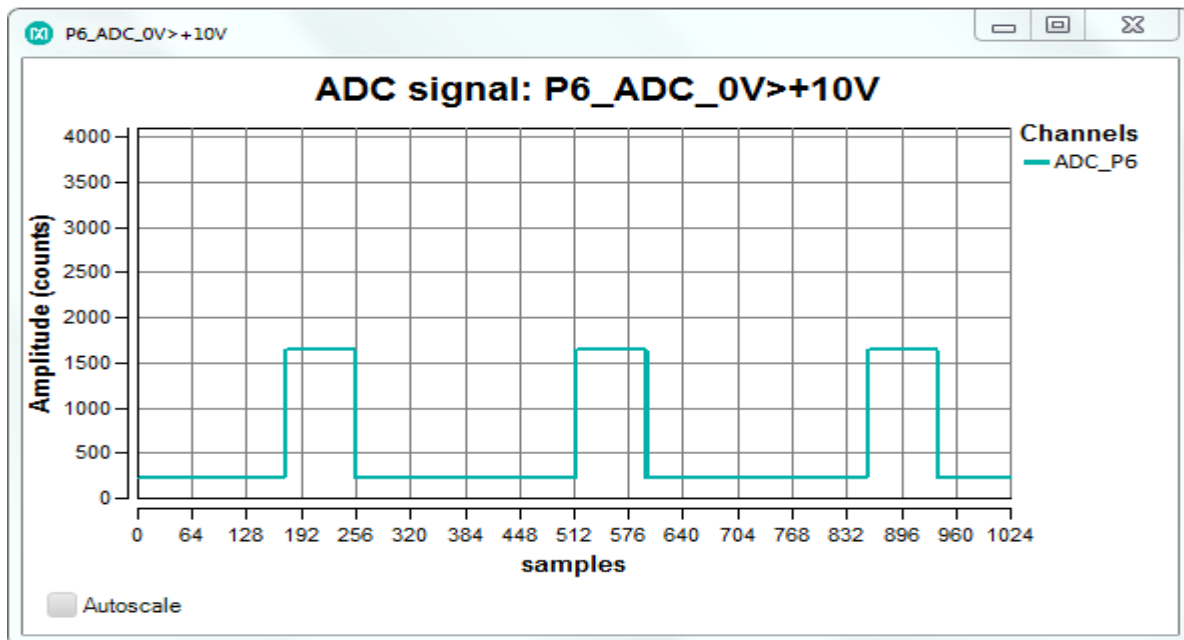


Figure 35. PWM width as the potentiometer set to a low position.

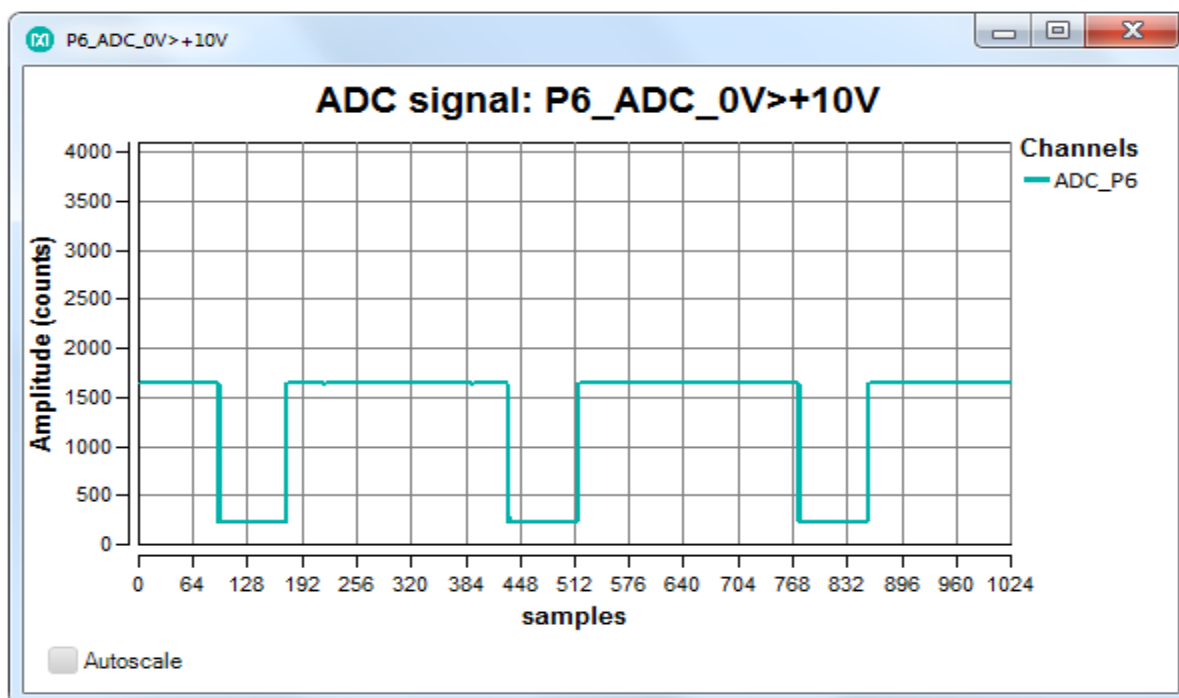


Figure 36. PWM width as the potentiometer set to a high position.

Note that the input of the level translator can only handle up to 5V. The provided configuration file is set to deliver a max of 5V in 0V to 10V mode. If other files are used, be sure not to apply more than 5V at any digital inputs.

7. References

For further detailed information, refer to the following documents available online:

1. MAX11311 data sheet: <https://datasheets.maximintegrated.com/en/ds/MAX11311.pdf>
2. MAX11311PMB1# data sheet: www.maximintegrated.com/MAX11311PMB1
3. MAX11300 Configuration Software: <https://www.maximintegrated.com/en/design/tools/applications/evkit-software/>
4. USB2PMB1# Adapter Board(Munich) data sheet: <https://datasheets.maximintegrated.com/en/ds/USB2PMB1.pdf>
5. USB2PMB1(Munich) Adapter Board GUI: www.maximintegrated.com/en/design/tools/applications/evkit-software/index.mvp/id/1161
6. MAX11300PMB1 and Munich Adapter Board Quick Start Guide: <http://pdfserv.maximintegrated.com/en/an/UG5795.pdf>

8. Trademarks

PIXI is a trademark of Maxim Integrated Products, Inc.

Pmod is a trademark of Digilent Inc.

Windows, Windows Vista, and Windows XP are registered trademarks and registered service marks of Microsoft Corporation.

©2016 by Maxim Integrated Products, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. MAXIM INTEGRATED PRODUCTS, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. MAXIM ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering or registered trademarks of Maxim Integrated Products, Inc. All other product or service names are the property of their respective owners.