MPLAB® Data Visualizer User's Guide

MPLAB® Data Visualizer

Notice to Development Tools Customers

Important:
All documentation becomes dated, and Development Tools manuals are no exception. Our tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our website (www.microchip.com) to obtain the latest version of the document.

Documents are identified with a DS number located on the bottom of each page. The DS format is DS<DocumentNumber><Version>, where <DocumentNumber> is an 8-digit number and <Version> is an uppercase letter.

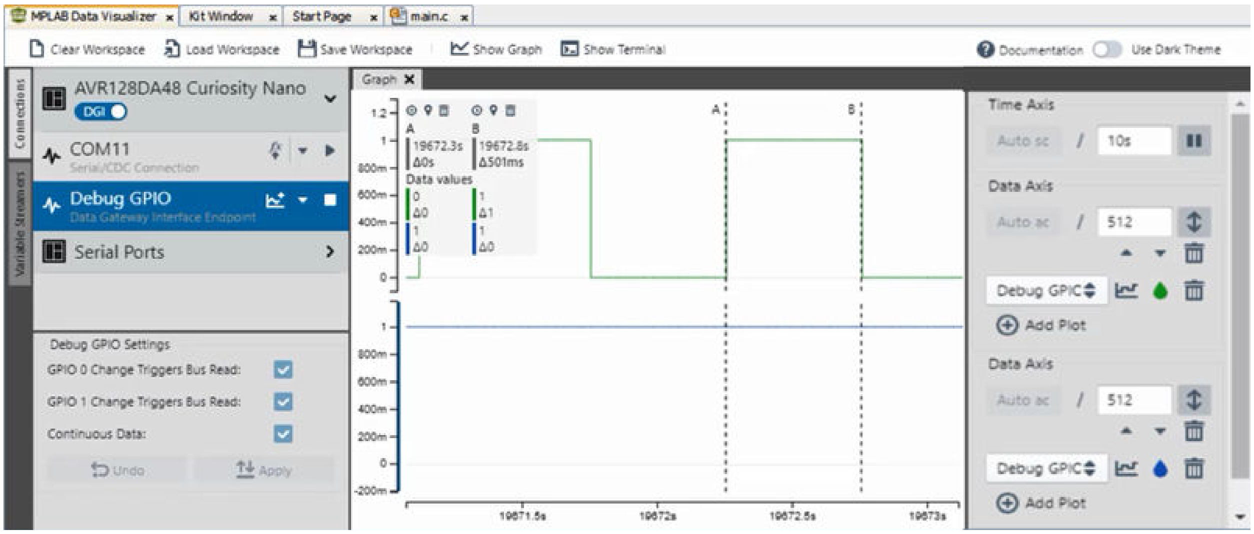
For the most up-to-date information, see the MPLAB® Data Visualizer help at onlinedocs.microchip.com/.

Table of Contents

1. Overview

The MPLAB Data Visualizer is a program used to process and visualize data from a running embedded target. The program may be accessed as an MPLAB X IDE plugin or standalone program.

Figure 1-1. Output of Streaming Data



1.1 MPLAB Data Visualizer Features

It can be difficult to troubleshoot data on an embedded target while your application is running. In the same way a debugger helps you debug your application code, MPLAB Data Visualizer helps you debug your data. With MPLAB Data Visualizer, you can see how key data points in your application change during runtime, e.g., visualize values captured by a sensor on your development board.

MPLAB Data Visualizer has the following features:

- Capture data streamed from a running embedded target via serial port (CDC) or the Data Gateway Interface (DGI)
- Decode data fields at runtime using the Data Stream Protocol format
- Visualize the raw or decoded data in a Graph as a time series or display the data in a terminal.
- Concurrently stream data and debug target code using the MPLAB® X IDE
- Analyze plotted data using cursors to measure bandwidth, pulse width and more

1.2 How the MPLAB Data Visualizer Works

The MPLAB Data Visualizer captures and displays data coming from a running embedded target (see figure).

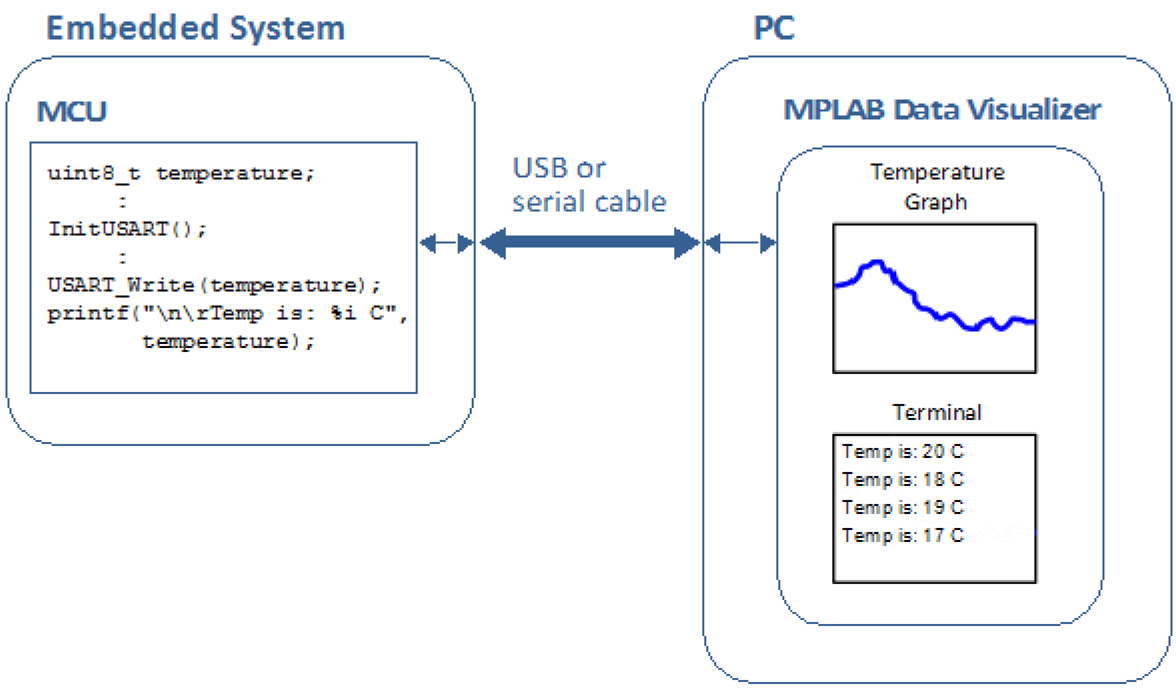
The Data Visualizer operates as an MPLAB X IDE plugin or as a standalone program. As a **plugin**, you can debug your code while using the Data Visualizer functions at the same time. As a **standalone program**, you can't debug your code. However, you CAN debug in MPLAB X IDE while streaming in standalone from the same kit.

Get started by viewing 2. Visualization Examples.

See the details of the visualizer and terminal interface, and the related controls, in 3. User Interface.

Find out how to connect the embedded target to your PC in 4. External Connections. For an understanding of how variable values are plotted, see 5. Variable Streamers.

Figure 1-2. Operational Overview



1.3 MPLAB Data Visualizer Installation

The visualizer operates in two ways: as an MPLAB X IDE plugin and as a standalone program.

MPLAB X IDE Plugin

To install the visualizer as a plugin:

- In MPLAB X IDE v5.30 and above, select *Tools>Plugins*, **Available Plugins** tab.
- From the list, check the "Install" box next to "MPLAB Data Visualizer" and then click **Install**.
- Follow the wizard dialogs to install the plugin.

For more information on installing plugins, see the MPLAB X IDE documentation, "**Add Plugin Tools**."

Once installed, activate the plugin from *Tools>Embedded*.

Standalone Program

To install the visualizer as a standalone program, go to the following "Microchip Gallery" link:

MPLAB-Data-Visualizer-Standalone

Launch-and-install or download-and-install. Follow the install wizard screens. Once the visualizer is installed, launch it from the install location.

2. Visualization Examples

You may find the examples in this section to be a helpful, hands-on way of understanding the features of the MPLAB Data Visualizer.

Details about the visualizer operation are found in the other sections.

2.1 Example of Streaming Raw Data

An MPLAB Xpress Evaluation Board is used to generate data based on poteniometer values. This board supports a serial/CDC connection, which is a valid connection for the visualizer.

One MPLAB Xpress code example is used to quickly develop an MPLAB X IDE application that will generate data. A second code example demonstrates how to set up a USART to transmit the data, via USB CDC connection, to the visualizer.

Tools used for this example are:

- MPLAB X IDE v5.40
- MPLAB XC8 C Compiler v2.20
- MPLAB Xpress Evaluation Board (PIC16F18855) - DM164140

2.1.1 Get Example Software and Hardware

Complete the following instructions to set up example software and hardware.

MPLAB X IDE

Download and install MPLAB X IDE 5.40 (or later) for free from the link below.

www.microchip.com/mplab/mplab-x-ide

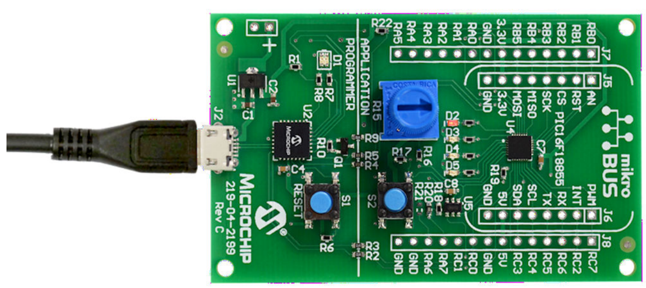
MPLAB XC8 C Compiler

Download and install the MPLAB XC8 C Compiler v2.20 (or later) from the link below. The free compiler is used for this example. A PRO compiler with additional optimizations and features is available for purchase.

www.microchip.com/mplab/compilers

MPLAB Xpress Evaluation Board - DM164140

Acquire this evaluation board from microchipDirect or a distributor.



2.1.2 Setup the MPLAB Xpress Evaluation Board

First, connect the MPLAB Xpress Evaluation Board to your computer via the enclosed USB cable to install the drivers. Next, you will need to determine:

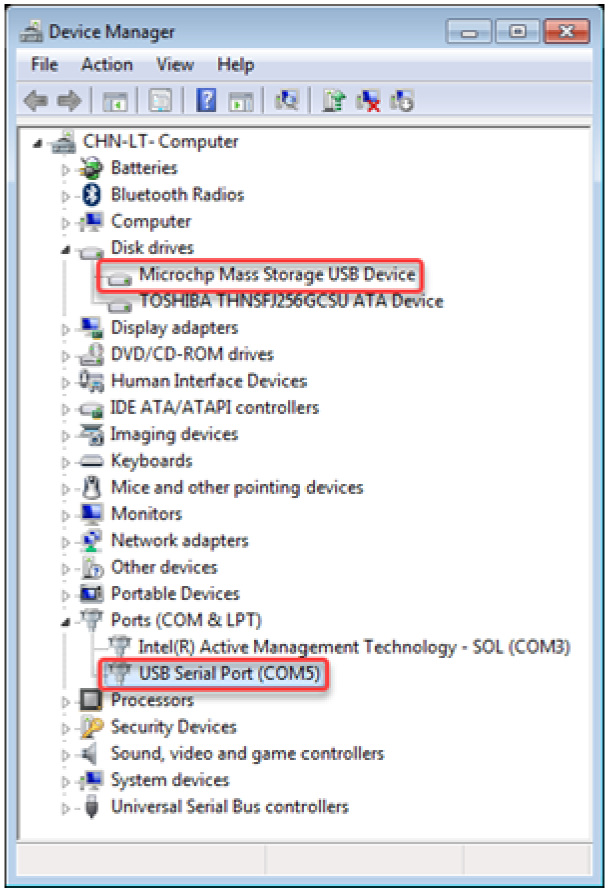
- If you need to use the USB CDC Driver for serial communications support. This driver is available on the MPLAB XPress webpage under the “Downloads” tab:

www.microchip.com/mplab/mplab-xpress

- The COM number of the USB port. This information will be needed later for data display.

Proceed by following the steps below.

1. Open the Device Manager (Windows) or System Information/Profiler (Mac) or lshw, etc (Linux).
2. Find the evaluation board to ensure it is connected.
   Note: An MPLAB Xpress evaluation board acts like a mass storage USB device.
3. View the ports. Take note of the COM number, e.g., USB Serial Port (COMx).
4. Unplug and replug the evaluation board. If a serial port vanishes and reappears, a serial connection is already available. If not, you will need to install the USB CDC Driver.

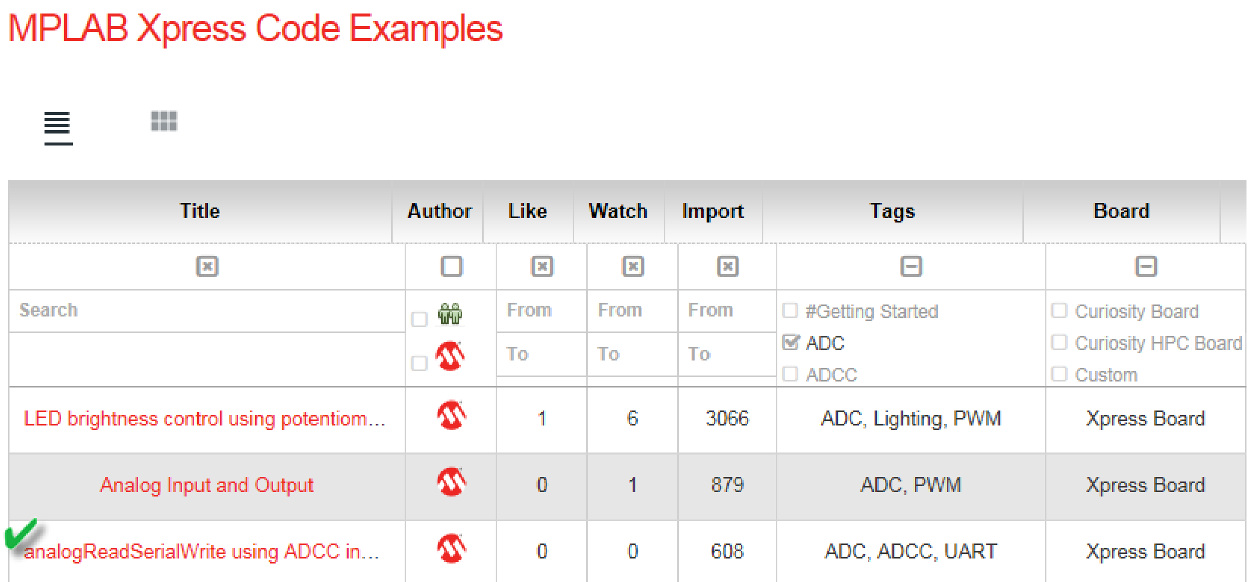


2.1.3 Get Example Code

Find code examples for the MPLAB Xpress evaluation board at the link below:

mplabxpress.microchip.com/mplabcloud/example

For this example, code that uses the potentiometer (pot) and the device ADC is desirable. To find this type of project, under “Tags”, check “ADC” and under “Boards”, check “Xpress Board”. The project used for this example will be “**analogReadSerialWrite using ADCC in Basic Mode**”.



MPLAB Xpress Code Examples

Title	Author	Like	Watch	Import	Tags	Board
Search		From / To	From / To	From / To	#Getting Started, ADC, ADCC	Curiosity Board, Curiosity HPC Board, Custom
LED brightness control using potentiom...		1	6	3066	ADC, Lighting, PWM	Xpress Board
Analog Input and Output		0	1	879	ADC, PWM	Xpress Board
analogReadSerialWrite using ADCC in...		0	0	608	ADC, ADCC, UART	Xpress Board

Click on the project name under “Title” to go to a detailed web page. On this page there is a link to Developer Help concerning this project, as well as device data sheet and evaluation board documentation links. After reviewing the page, click on the **Download** button to download the project and code.

Once you have downloaded the project, unzipped it into the `MPLABXProjects` folder on your computer. `MPLABXProjects` may be found in your User directory, where the path is shown in the ***Help>About*** window.

2.1.4 Set Project Properties and Build

Open the project in MPLAB X IDE. Right click on the project name and select “Properties.” In the Project Properties window (see figure below), select your configuration (free or pro compiler).

1. Choose “Simulator” under “Connected Hardware Tool.” The MPLAB Xpress development board will not show in this window as it is not a debug tool. It can be programmed as described in 2.1.5 Program Example Code.
2. Choose the highest-number pack under “Packs.” This will correspond to the pack version installed with MPLAB X IDE.
3. Choose the highest-number XC8 compiler version. Find the MPLAB XC8 compiler under www.microchip.com/mplab/compilers.
4. Click **OK** to close the window. Then debug the project to ensure it builds and runs.

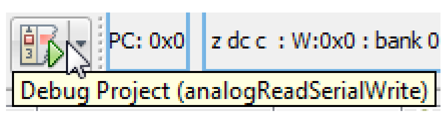
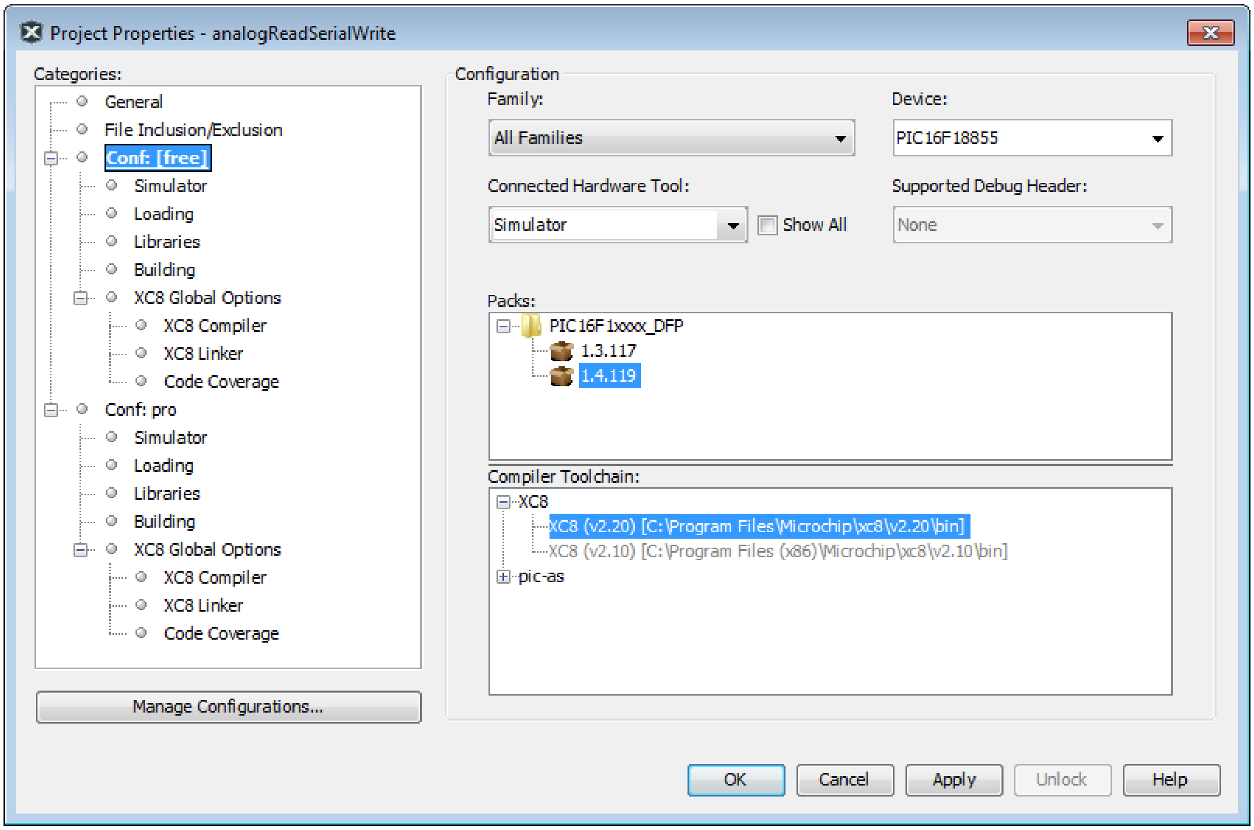


Figure 2-1. Project Properties Window



2.1.5 Program Example Code

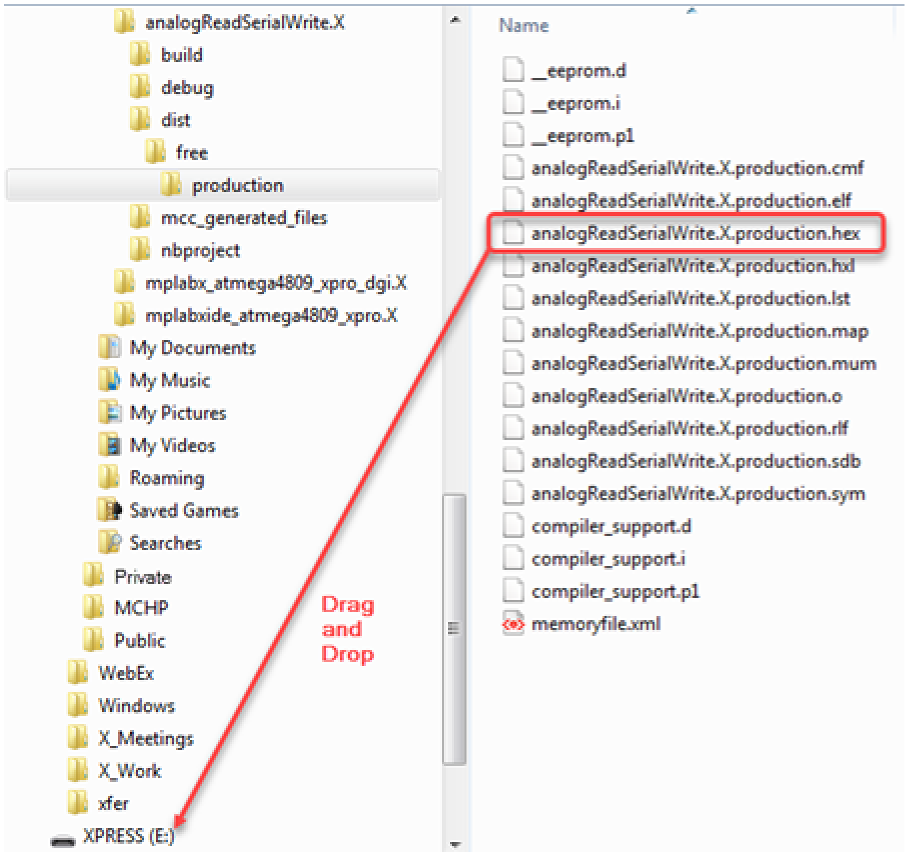
The MPLAB Xpress development board acts like a USB drive when connected to the PC. Therefore, to program the PIC16F18855 device on the board, you simply need to drag and drop the project executable (Hex) file onto the board.

Find the Hex file in the Project Folder

Locate the “analogReadSerialWrite.X” project on your PC. `MPLABXProjects` may be found in your User directory, where the path is shown in the *Help>About* window. Then find the Hex file under `dist/free/production/analogReadSerialWrite.X.production.hex`.

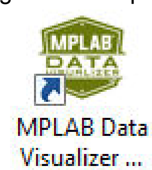
Program the Device

Find “XPress” as a USB connected device. Drag and drop the Hex file to “Xpress” to program the board.



2.1.6 Open MPLAB Data Visualizer

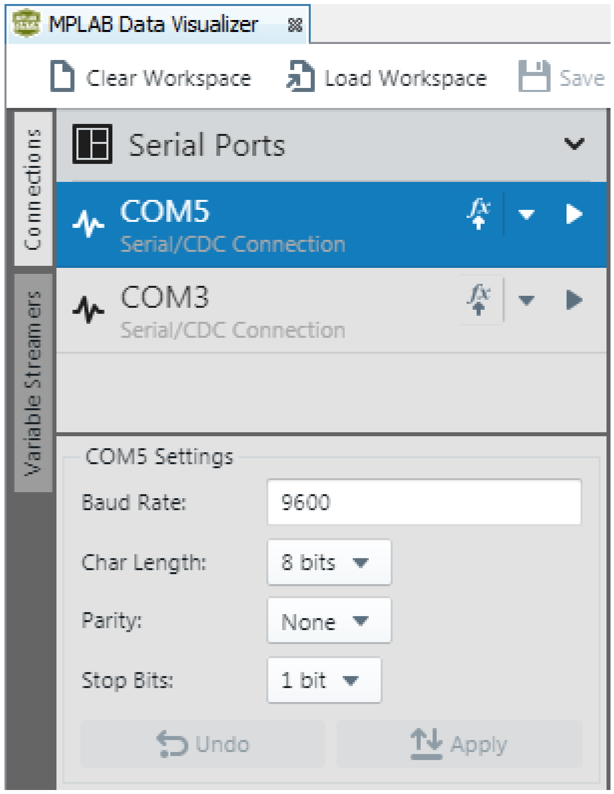
Open the MPLAB Data Visualizer by double clicking the desktop icon.



In the Data Source pane, on the **Connections** tab, you should see “Serial Ports.” The available COM ports are shown underneath. From 2.1.2 Setup the MPLAB Xpress Evaluation Board, we know the COM port for the MPLAB Xpress board. Click to select that COM connection.

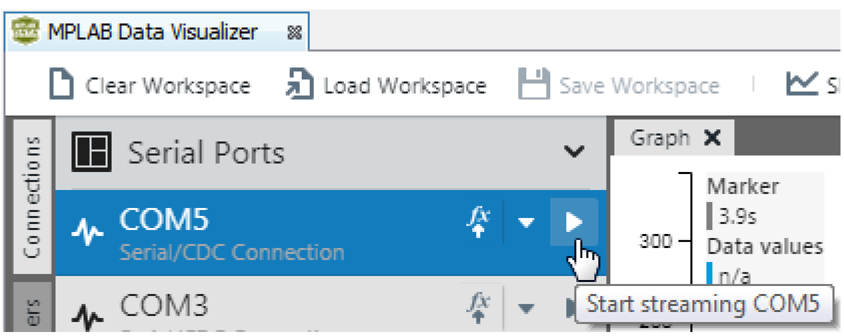
For more on the COM Settings, see 4.1 Serial Port.

Figure 2-2. Click to Highlight COM Connection



Click the arrow on the right to start data streaming. No output will be displayed in the Graph or Terminal yet.

Figure 2-3. Click to Start Streaming



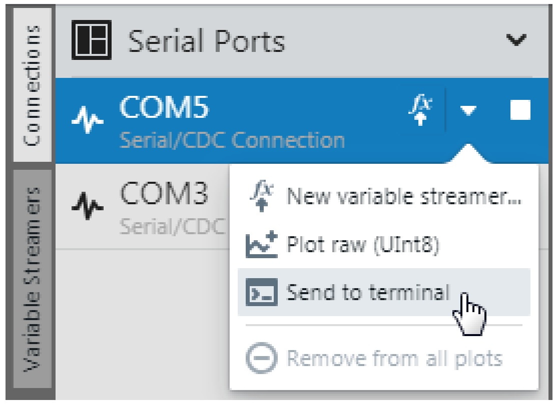
2.1.7 View Example Output in Terminal

In `main.c`, in the `while` loop, is one line of code:

```
printf("\n\rADCC Value is: %i    ", ADCC_GetSingleConversion(POT));
```

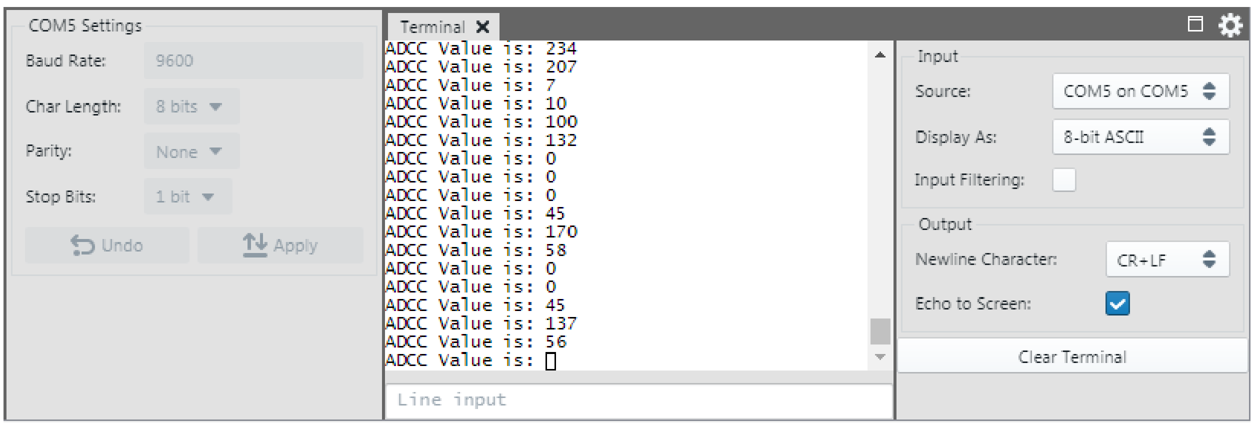
This will output data that can be viewed in the Terminal window. To view this data, click on the down arrow and select "Send to Terminal."

Figure 2-4. Select Send to Terminal



You should see the output from the `printf` statement. Change the pot on the board to see the values change.

Figure 2-5. Output In Terminal Window

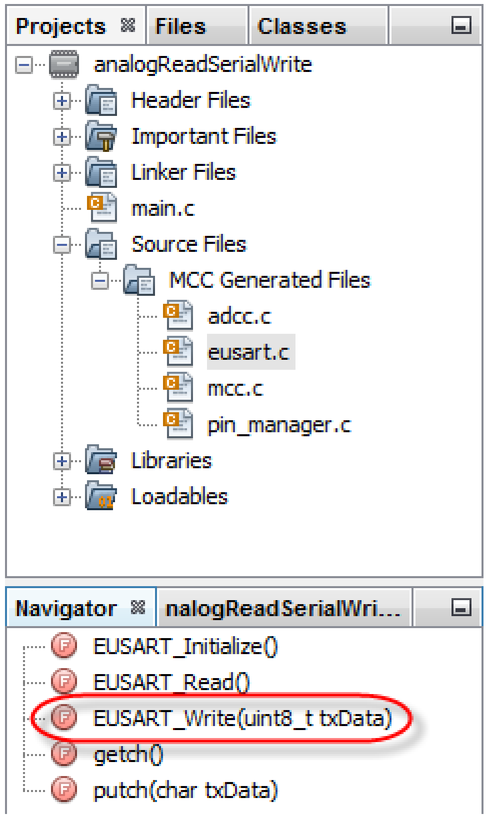


2.1.8 Modify Example Code

To graph the ADC data from the potentiometer in the download MPLAB Xpress code example, the data will be transmitted by byte using the EUSART on the PIC16F18855, as USART is a supported format in the MPLAB Data Visualizer.

Using EUSART Functions

In the example project, there is code for an EUSART that can be used for this purpose. In the Projects tree, click on `eusart.c`. In this file, the `EUSART_Write()` function will be used.



In `main.c` in the `while` loop, add this `EUSART_Write()` line of code:

```
EUSART_Write(ADCC_GetSingleConversion(POT));
```

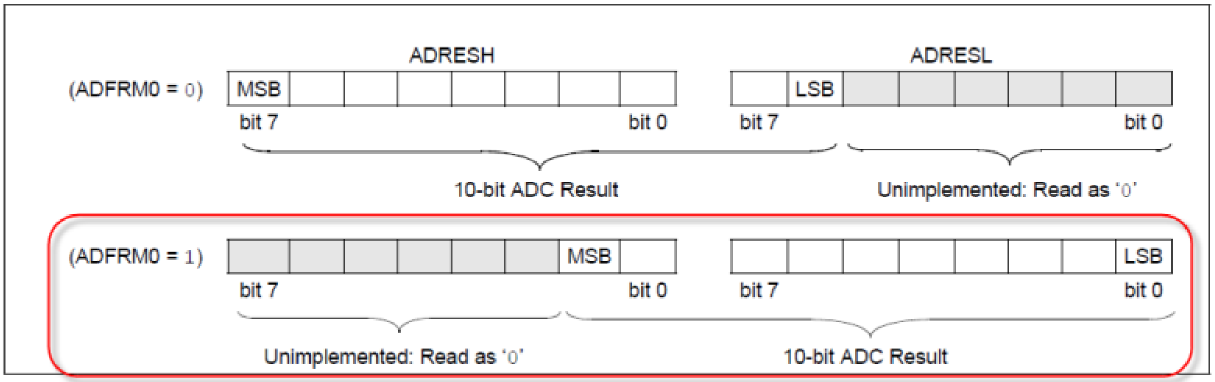
Unfortunately this will not work as the ADC produces a 10-bit result and the EUSART write function is looking for an 8-bit value. Therefore, this value needs to be adjusted.

In the PIC16F18855 data sheet, two formats for the ADC result are shown (see figure). The format used depends on the ADCON0 register ADFRM0 bit value, where '`1`' means data are right-justified and '`0`' means data are left-justified, zero-filled. Inspect the `adcc.c` file to find the following information concerning the ADCON0 register, where ADFM = ADFRM0:

```
// ADGO stop; ADFM right; ADON enabled; ADCONT disabled; ADCS FOSC/ADCLK;
ADCON0 = 0x84;
```

Therefore, the second ADC format is the one used in this example.

FIGURE 23-3: 10-BIT ADC CONVERSION RESULT FORMAT



So to modify the input to the EUSART write function so that the most-significant 8 bits are used:

```
EUSART_Write((ADCC_GetSingleConversion(POT) & 0x03FC)>>2);
```

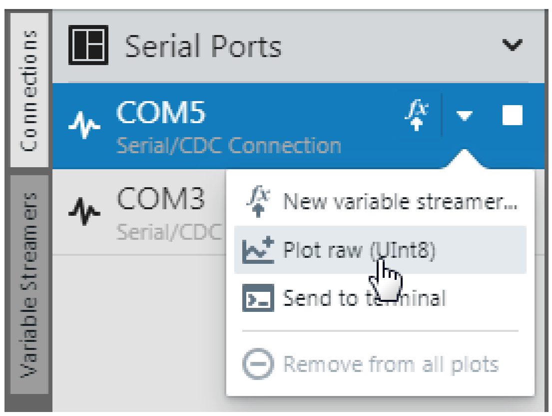
Debugging the Modified Code

Debug Run the new code to ensure it executes. If there are errors, review the instructions above.

2.1.9 Example Visualization

Reprogram the MPLAB Xpress board for EUSART data. To view this data, click on the down arrow and select "Plot raw (Uint8)."

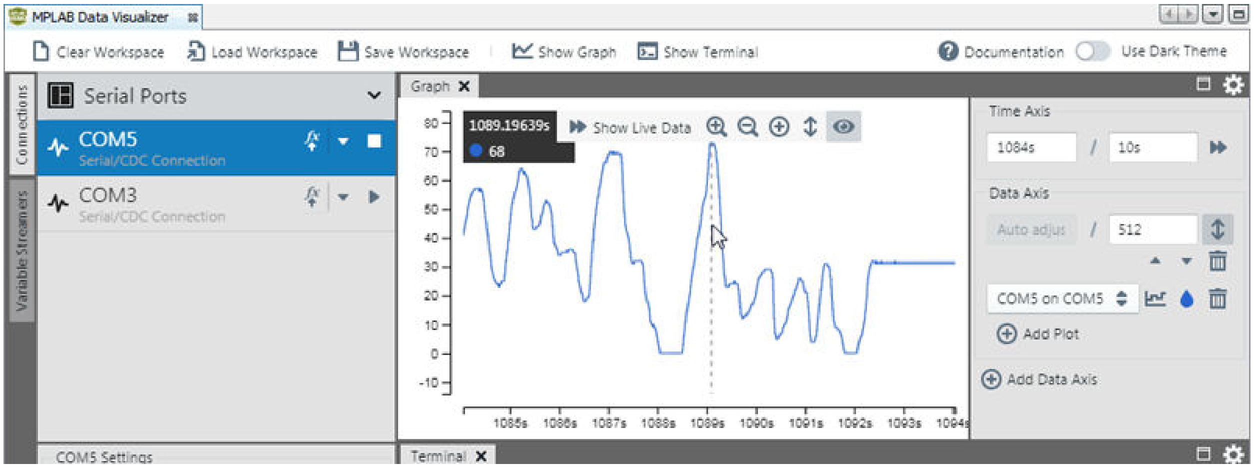
Figure 2-6. Select to Plot Raw Data



You should now see incoming data displayed on the graph in the Center Pane. Turn the potentiometer back and forth on the MPLAB Xpress board to see the data graph change.

Other options are available on Time Axis for setting the data range, an offset, or other graph formatting. For more information, see 3. User Interface.

Figure 2-7. Graph of Raw Data



Related Links

3.1 Data Sources (Left) Pane

3.3 Graph Visualization Controls (Right) Pane

3.2 Graph Window

2.2 Example of Plotting Data

An ATmega4809 Xplained Pro Board is used to generate data that is output via the DGI/SPI to the visualizer and plotted using variable streamers.

Tools used for this example are:

- MPLAB X IDE v5.40
- MPLAB XC8 C Compiler v2.20
- ATmega4809 Xplained Pro Board - ATMEGA4809-XPRO

2.2.1 Example Setup

Complete the following instructions to set up the example software and hardware.

MPLAB X IDE

Download and install MPLAB X IDE 5.40 (or later) for free from the link below.

www.microchip.com/mplab/mplab-x-ide

MPLAB XC8 C Compiler

Download and install the MPLAB XC8 C compiler v2.20 (or later) for free from the link below. A PRO version of the compiler with additional optimizations and features is also available for purchase.

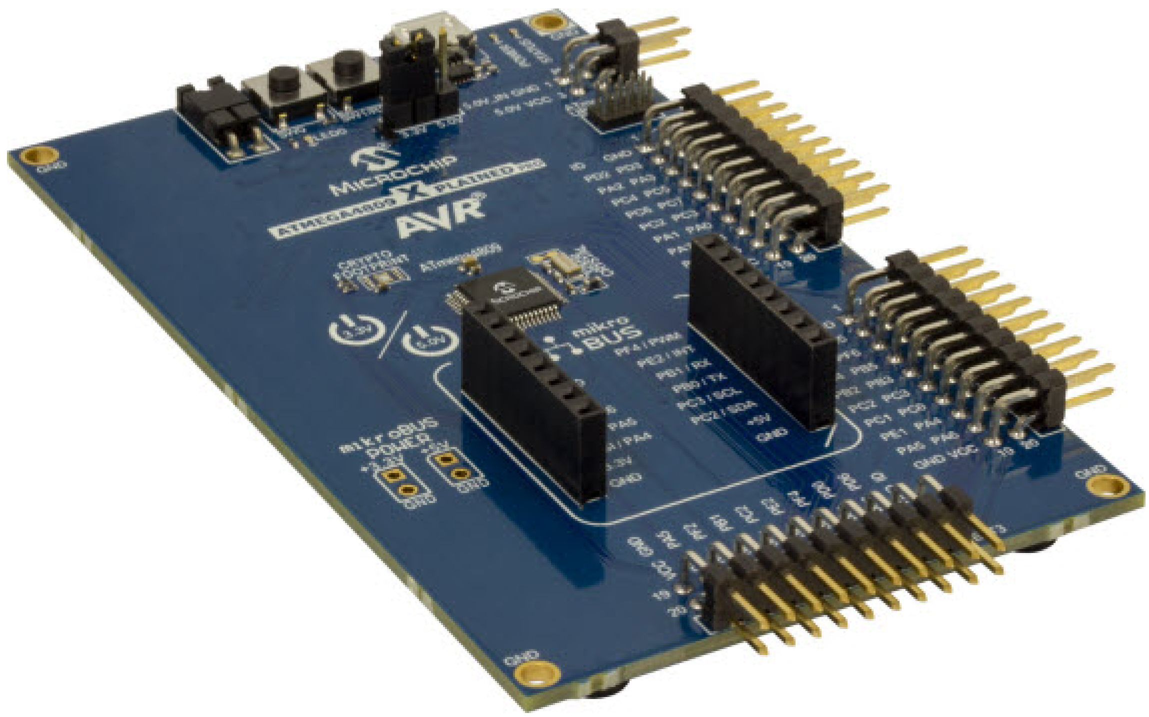
www.microchip.com/mplab/compilers

ATmega4809 Xplained Pro Board - ATMEGA4809-XPRO

Acquire this evaluation board (see image below) from microchipDirect or a distributor. Then connect the board to your computer via the enclosed USB cable to install the drivers.

For more information about this board, go to:

www.microchip.com/developmenttools/ProductDetails/PartNo/ATMEGA4809-XPRO



2.2.2 Create Example Project

MPLAB X IDE requires a project for development of application code.

Preliminaries

Before creating the project ensure:

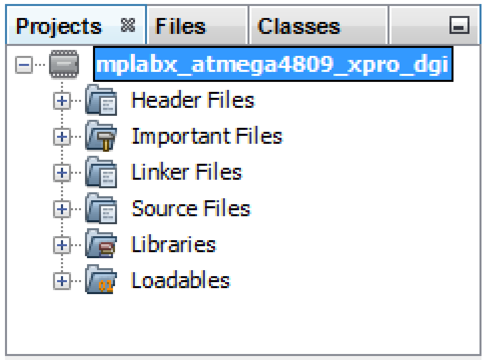
- You have installed the compiler and MPLAB X IDE can detect it. If not, go to *Tools>Options>Embedded>Build Tools* to view the Toolchain list. If the compiler is not there, click **Add** to browse and add it.
- You have plugged the Xplained Pro board into your computer with the USB cable.

Create Project

Select *File>New Project* or the **New Project** icon  to open the Project wizard. Follow the steps below to create your project. Click **Next** to move to the next step.

1. **Choose Project**: Click on the “Microchip Embedded” category and then the “Standalone Project” project.
2. **Select Device (and Tool)**: Enter the device “ATmega4809”. Then enter the tool “ATmega4809 Xplained Pro-SN: ATML#” where the tool serial number (SN) contains the prefix “ATML” followed by a multi-digit number.
3. **Select Compiler**: Under *Compiler Toolchains>XC8*, Select the most-current compiler version.
4. **Select Project Name and Folder**: Name your project. For example, “mplabx_atmega4809_xpro_dgi”. For Windows OS, the default project folder is `C:\Users\<UserName>\MPLABXProjects`.

After clicking **Finish**, the project tree should appear in the Projects window.



2.2.3 Add Files to Project

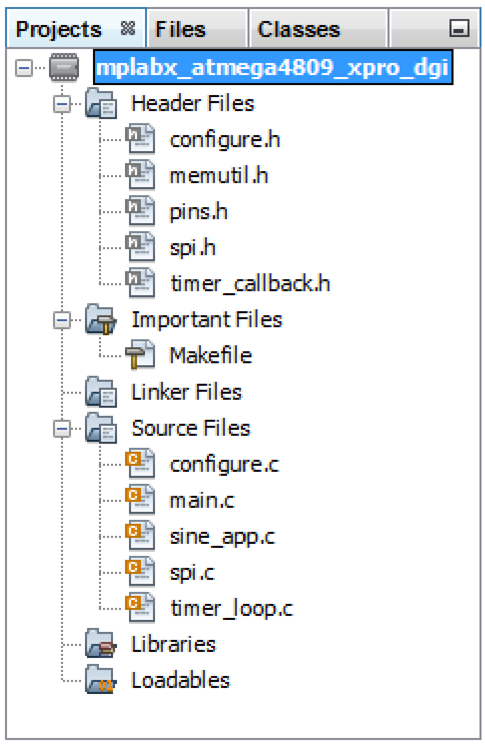
To add C header files to the project, right click on the "Header Files" folder and select *New>C Header File*.

To add C source code to the project, right click on the "Source Files" folder and select *New>C Main File* (once, for `main.c`) or *New>C Source File* for all other files.

Example code for this project is found in 8. Example of Plotting Data - Code Listing.

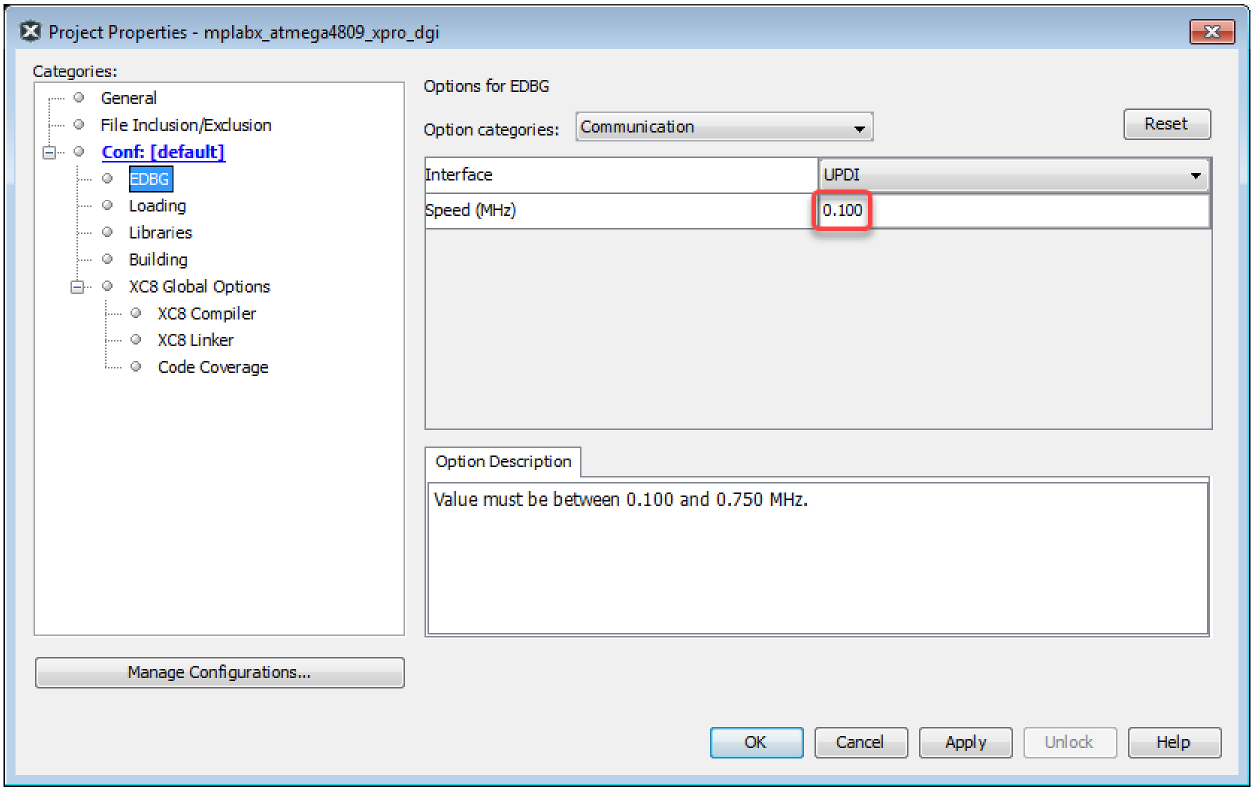
The completed project should look like the figure below.

Figure 2-8. Project with Files



2.2.4 Set Project Communications Options

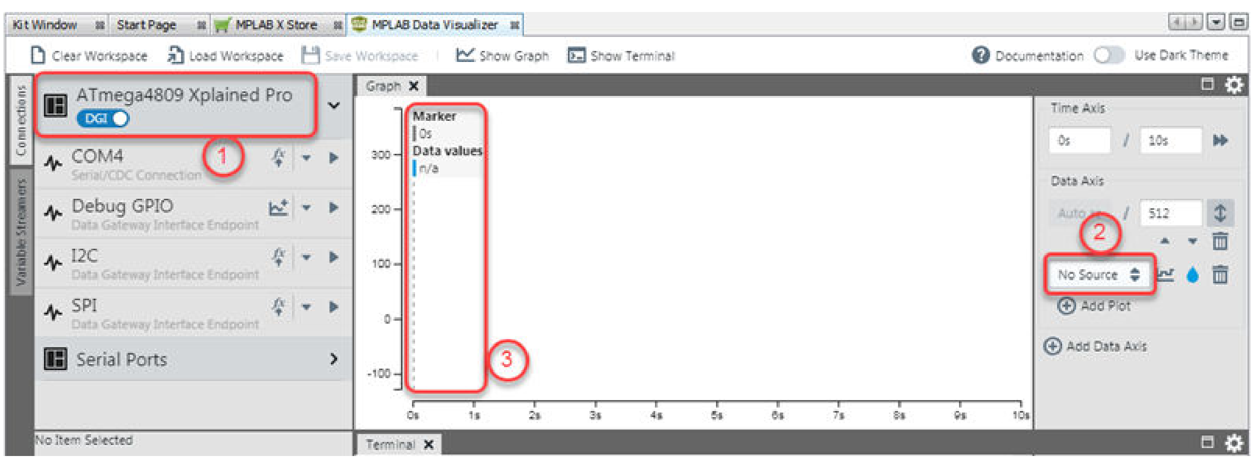
Right click on the project name in the Projects window to open the Project Properties. Under EDBG Communication options, make the Speed "0.100" to match the settings in code.



2.2.5 Open MPLAB Data Visualizer

Open the MPLAB Data Visualizer plug-in by clicking the toolbar icon or selecting *Tools>Embedded>Data Visualizer*.

1. In the Data Source pane, on the **Connections** tab, you should see “ATmega4809 XPlained Pro” with “DGI” enabled.
2. In the Visualization pane, the “Source” box for the Time Axis should say “No Source.”
3. In the Graph tab, the sliding marker (dashed gray line) will be at zero and there will be no data values (if available would be blue to match source color).

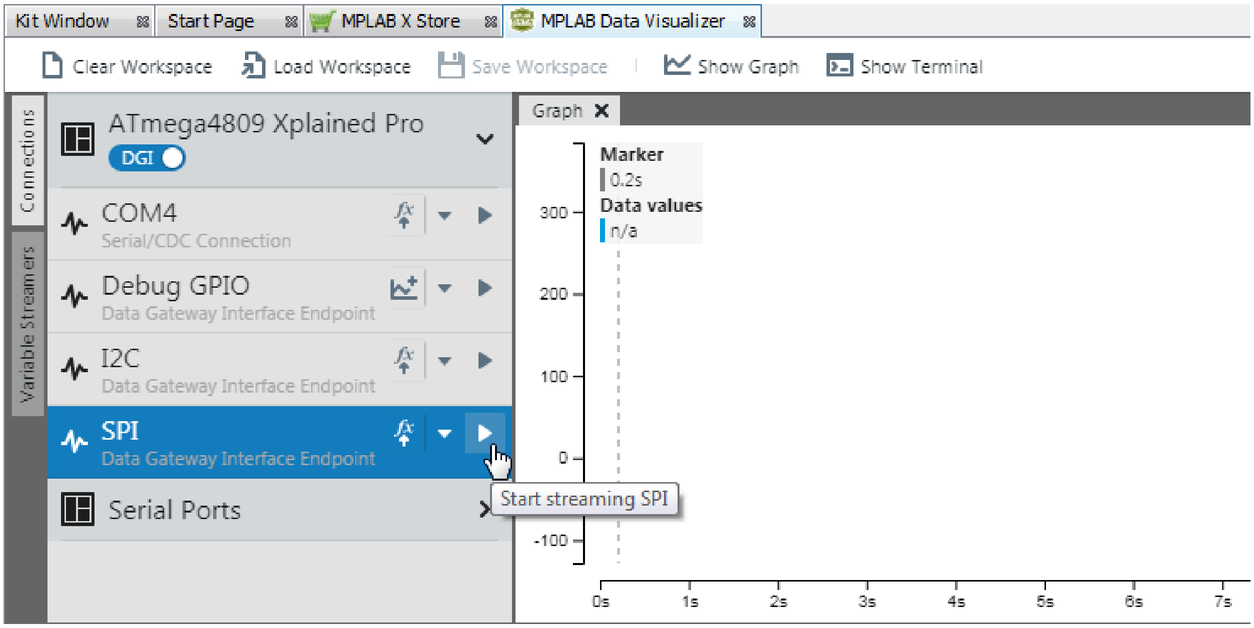


2.2.6 Debug Project and Visualize Output

To begin debugging the project, click on the "Debug Project" icon .

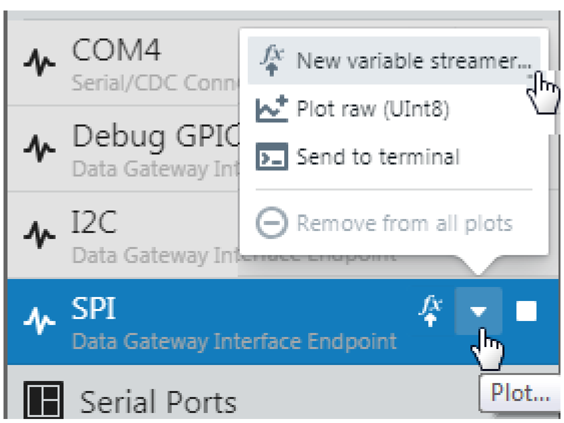
In MPLAB Data Visualizer, under "ATmega4809 Xplained Pro" (DGI enabled), click on the SPI button right-arrow control to enable SPI streaming.

Note: When you click on the SPI connection, you will see "SPI Settings" below. For details on what these settings mean, see 4.2.4 SPI Interface.

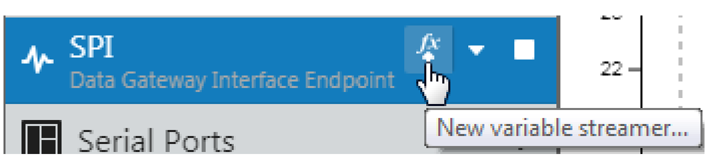


Once you start streaming, a banner will appear on the top of the graph. You can click on "Plot" to start setting up the plot. However, this banner is only visible for a few seconds.

You can click on the arrow down (Plot) control on the SPI button. Then select the type of plot to set up.



If you are going to set up a variable streamer, you can click the fx (New Variable Streamer) control instead.



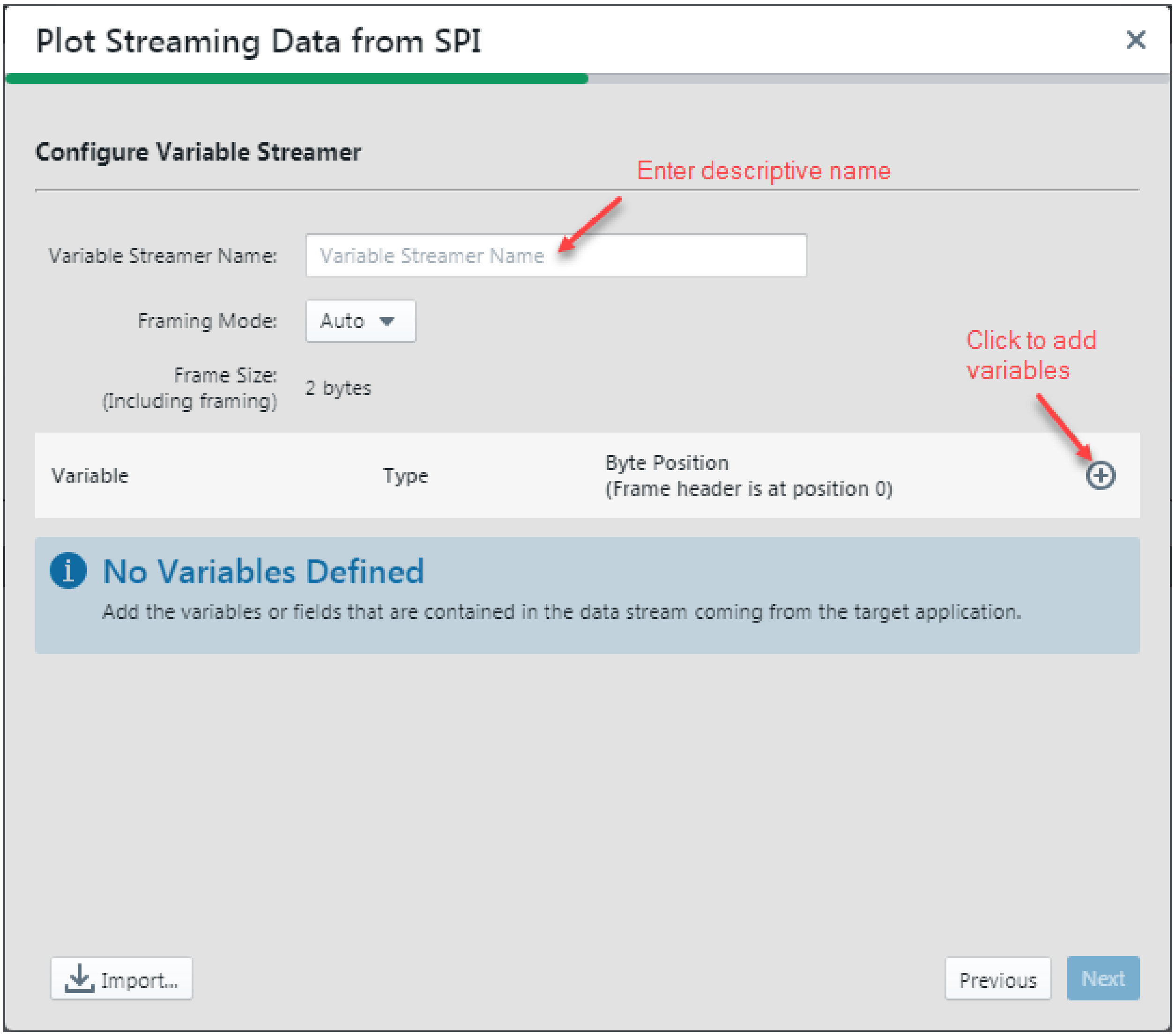
2.2.6.1 Plot Streaming Data for SPI

Configure Variable Streamer - Initial View

For this example, the values of a variable will be streamed and plotted. The first dialog of the plotting wizard will look as below. Name the variable streamer to identify the setup later. Then add the variable by clicking on the "+" as shown to display text boxes for entry.

Note: If a previous setup had been saved, you could load it by clicking the **Import** button on the bottom of the dialog.

Also, using the "?" key opens a keyboard shortcuts dialog, and using the "Esc" key closes the dialog.

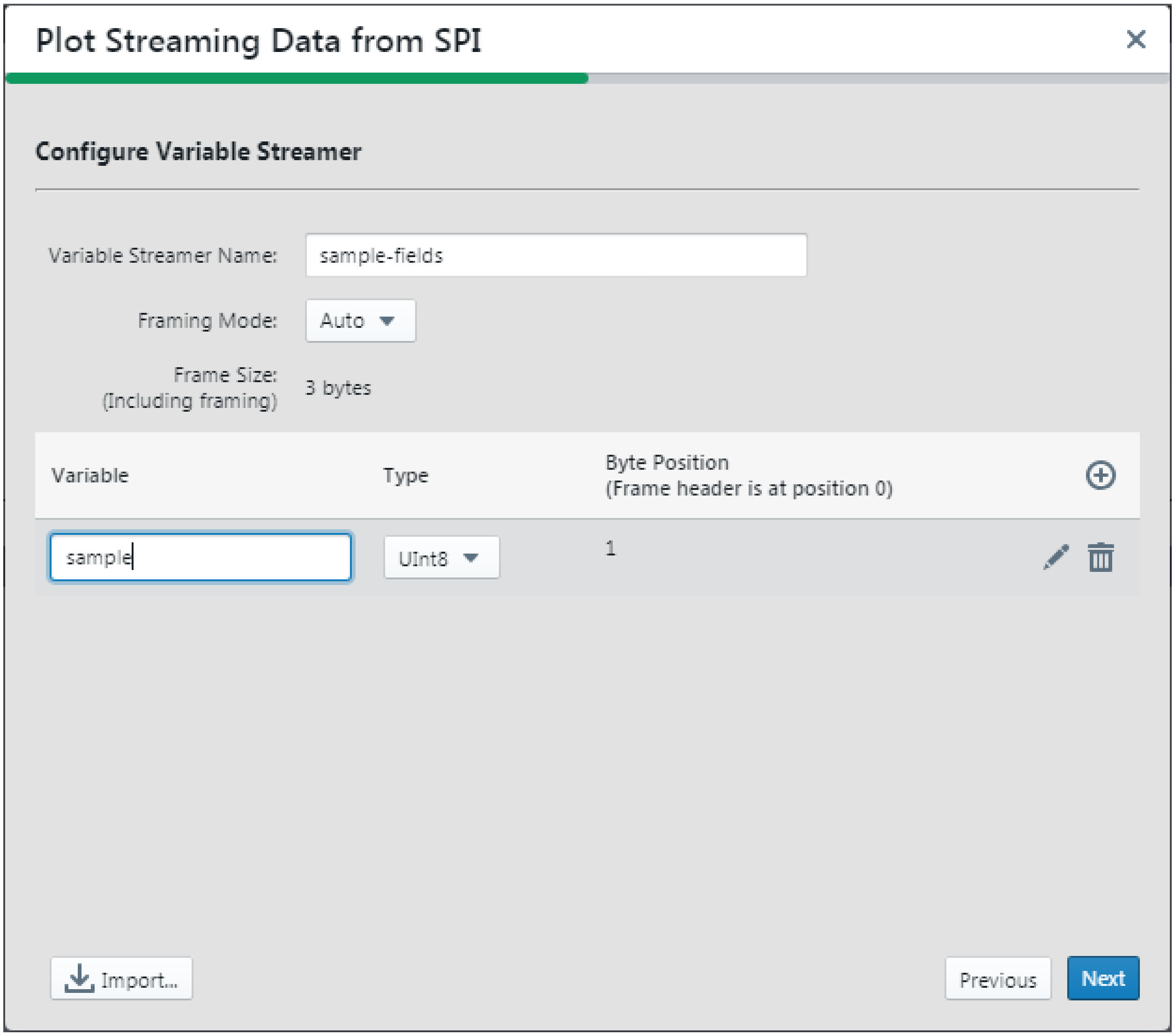


Configure Variable Streamer - Enter Data

The dialog below shows the previous dialog will data entered. Specified information on the variable has been provided.

In order to decode a data stream, the variables (or fields in the data stream) must be defined. The data streamed will be of the format shown in 5.2 Stream Format.

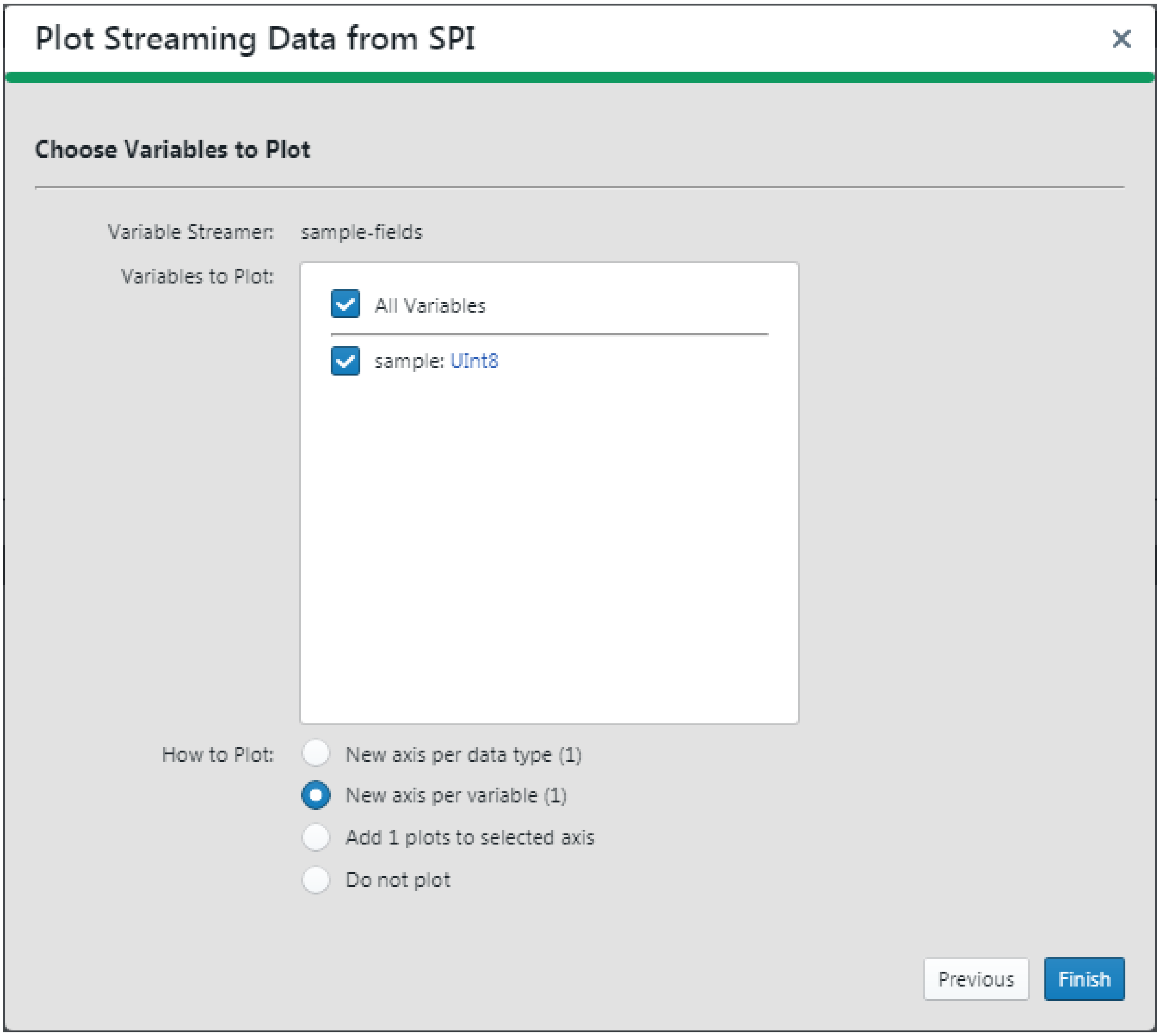
Click **Next** to proceed with setup.



Configure Variable Streamer - How to Plot

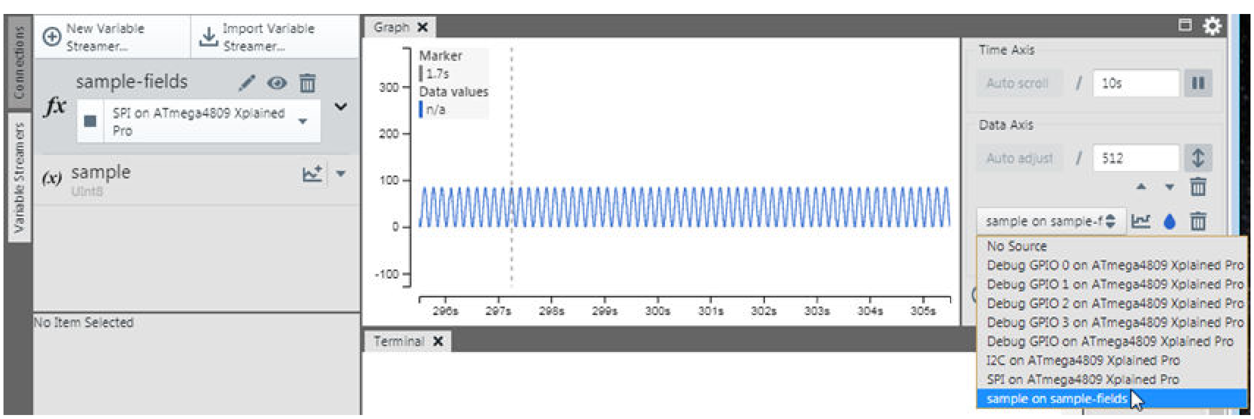
This dialog shows a summary of the previous one and a selection list of how to plot the data. For this example, "New axis per variable (1)" has been selected.

Click **Finish** to proceed to plot.

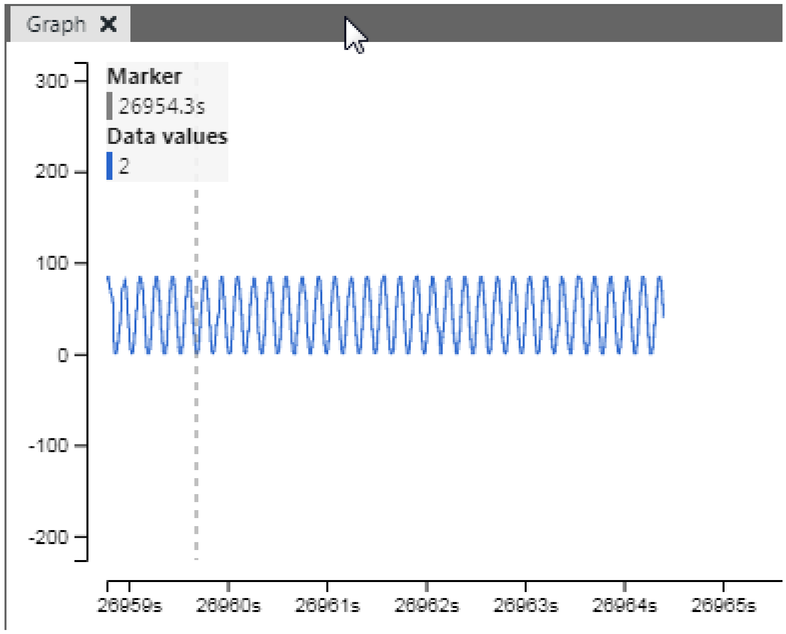


2.2.6.2 View Data based on Plot Setup

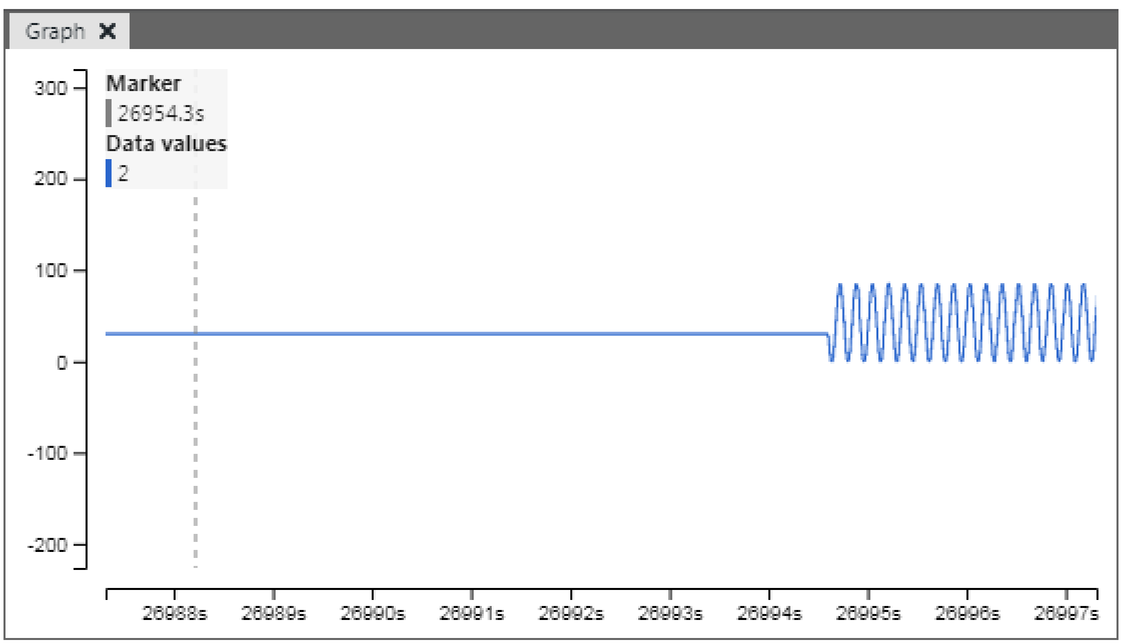
The Time Axis Source should now be set to “sample on sample-fields” and data should be plotted on the graph. In order to toggle the data scrolling in the graph, press "Pause Scrolling/Show Live Data" on the Graph banner, or use the Space key.



When debug is paused , data output is stopped.



When debug is continued, , data is again shown on the graph.

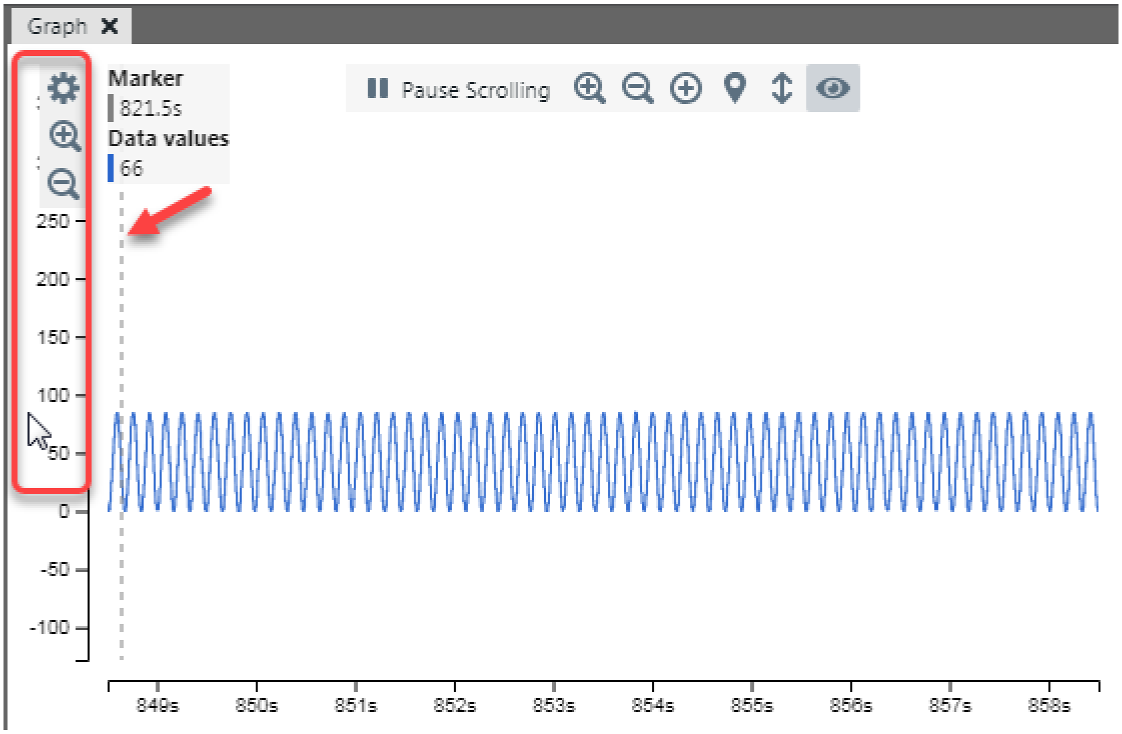


2.2.6.3 Analyzing Plot Data with Tools

Graph tools may be used to change the view of and analyze plot data.

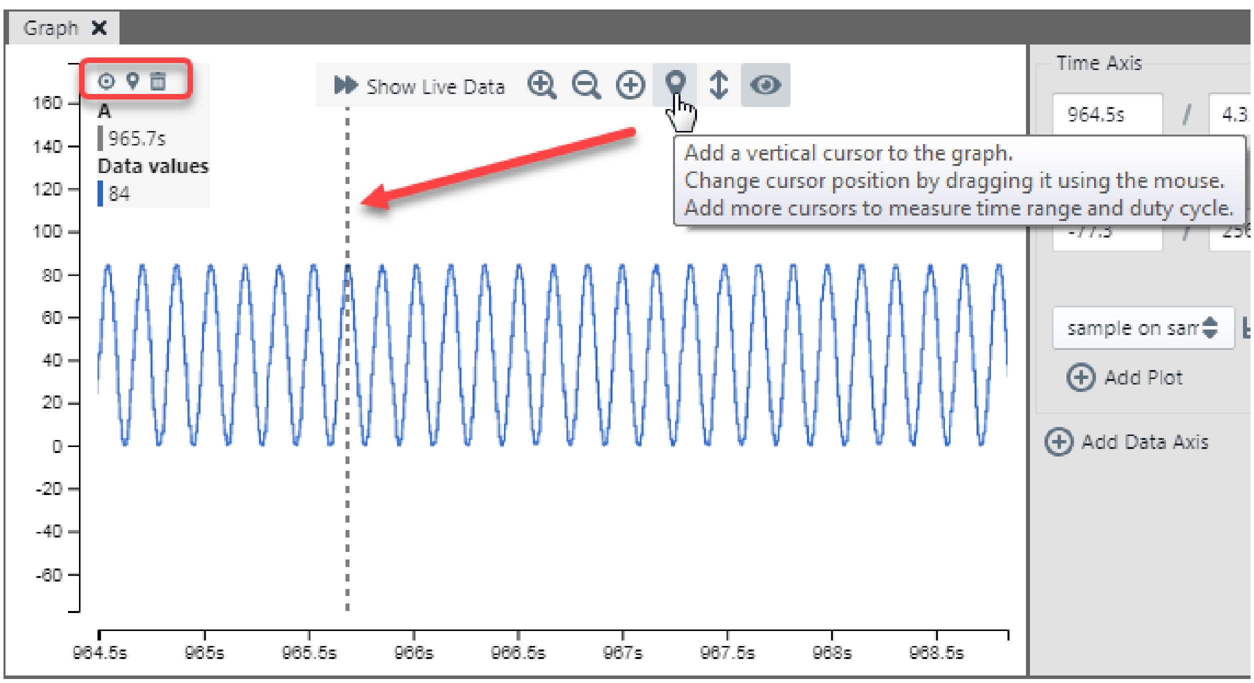
Adjusting an axis range and plot location can make viewing data easier. In the image below, the Data Value (vertical) axis is circled for example. Click on or near the axis you want to adjust. Then use the mouse wheel to zoom in or out on the axis range. You can also click and hold to drag the axis one way or the other, thus moving the plot accordingly. Also. controls at one end of the axis can be used to zoom in or out, and set plot characteristics, as on the Visualization pane.

On the graph there is a rolling vertical marker that will follow mouse movements and show the corresponding Time (horizontal axis) and Data Value (vertical axis).

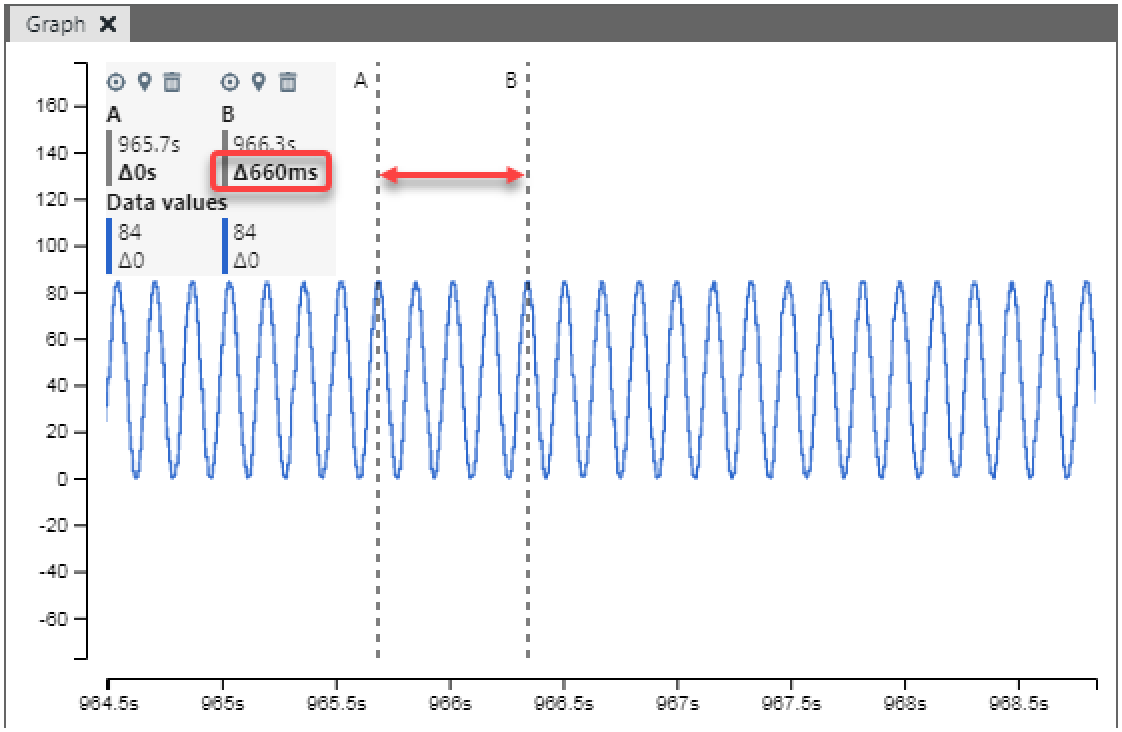


To enable a vertical cursor that may be dragged and dropped at a plot location, click as shown below to enable cursor A. This will disable the rolling marker.

Controls for the cursor will appear where the rolling marker information had been located.



To determine time between plot points, enable another vertical cursor, set its location, and then view the time delta.



2.3 Example of Multiple Data Plots

An AVR128DA48 Curiosity Nano board is used to demonstrate how to use GPIO pins to generate multiple data plots, either on the same axis or different axes.

Tools used for this example are:

- MPLAB X IDE v5.40
- MPLAB XC8 C Compiler v2.20
- AVR128DA48 Curiosity Nano Evaluation Kit - DM164151

2.3.1 Example Setup

Example Setup Follow the instructions in the following sections to set up example software and hardware.

MPLAB X IDE

Download and install MPLAB X IDE 5.40 or later for free from the link below.

www.microchip.com/mplab/mplab-x-ide

MPLAB XC8 C Compiler

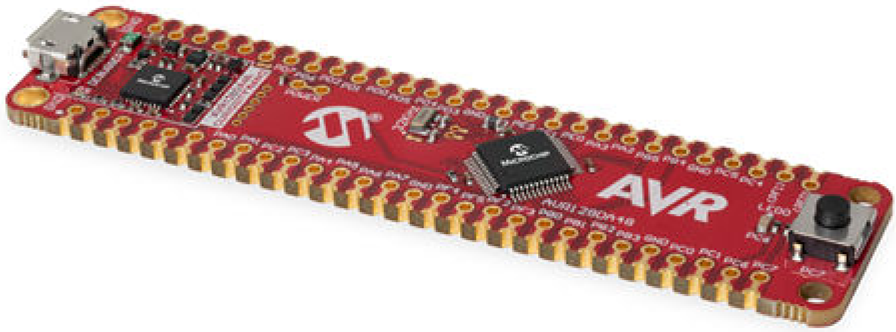
Download and install the MPLAB XC8 C compiler v2.20 or later for free from the link below. A PRO version of the compiler with additional optimizations and features is also available for purchase.

www.microchip.com/mplab/compilers

AVR128DA48 Curiosity Nano Evaluation Kit - DM164151

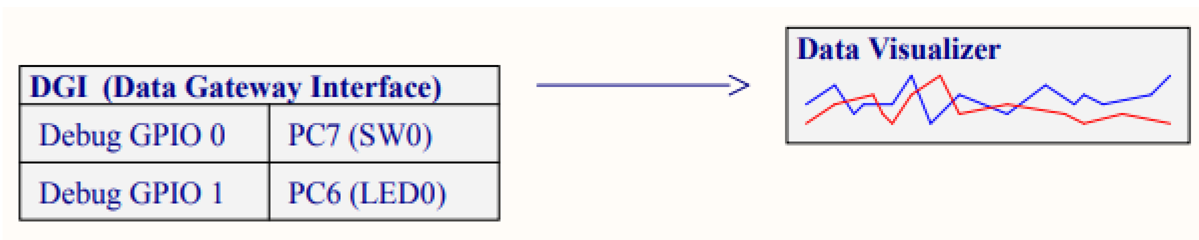
Acquire this evaluation board (see image below) from microchipDirect or a distributor. Then connect the board to your computer via the enclosed USB cable to install the drivers. For more information about this board, go to:

www.microchip.com/DevelopmentTools/ProductDetails/PartNO/DM164151

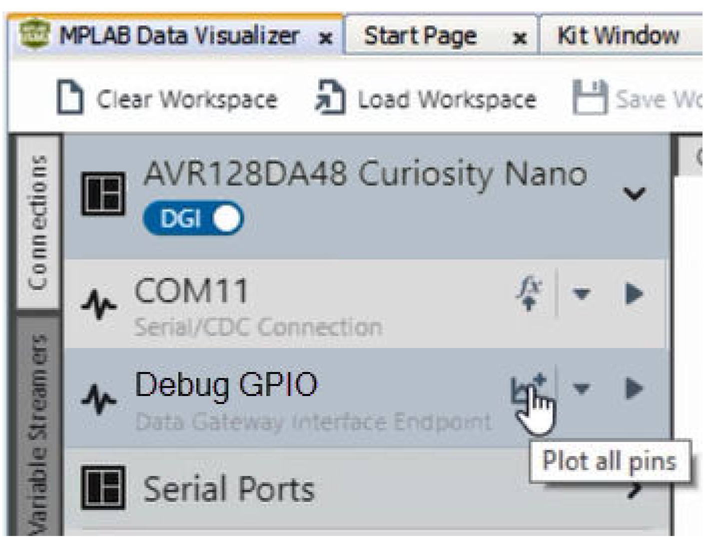


2.3.2 Plug and Play with the Curiosity Nano

The AVR128DA48 Curiosity Nano Evaluation board is designed to be plug-and-play. Therefore plug the board into the PC using the USB cable and then launch MPLAB X IDE. When the IDE opens, you should see a **Kit Window** tab with information about the Curiosity Nano. Click on the board schematics link and on the first page of the schematics find GPIO pin references that can be used with the MPLAB Data Visualizer. PC7 is attached to the board pushbutton switch and PC6 is attached to LED0.

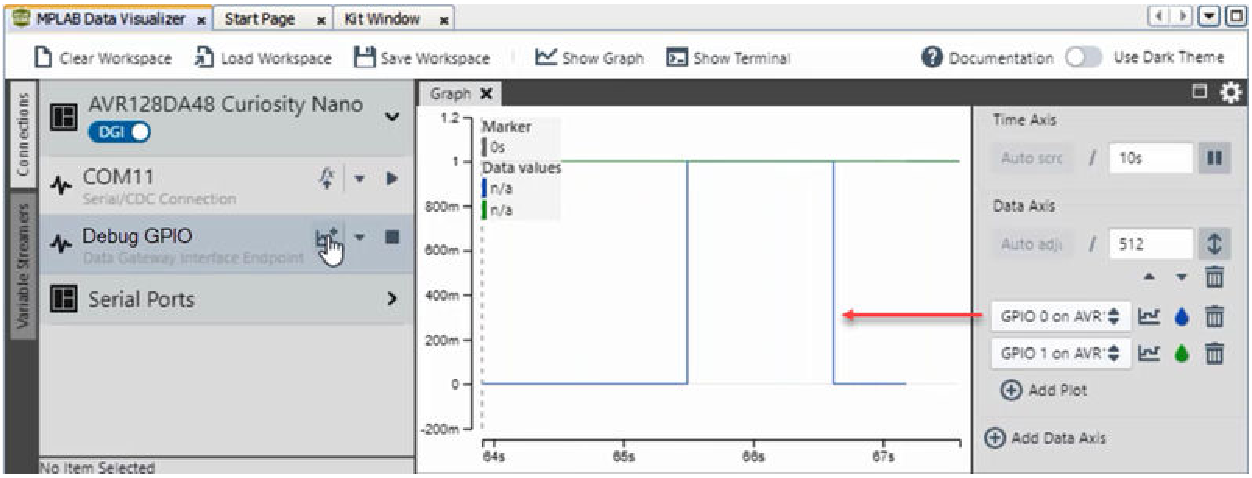


Open the MPLAB Data Visualizer plugin. The visualizer will display available data sources, including Debug GPIO for the GPIO pins. Click on the plot icon to plot all pins.



On the Graph, both GPIO pin outputs are plotted on a single axis. On the right side of the graph you will see information about each plot, including its color-coding.

Now it's time to play with the Curiosity Nano. Press the board switch to see a pulse on GPIO 0. Note that there is lag between when the button is pressed and when the pulse appears. Refer again to the schematic to see that there is no pull-up on the button. However, pin pull-ups can be enabled using software. Also, GPIO 1 is only showing a single-line plot, but a pulse produced from a timer could provide a more interesting plot. Therefore it is time to create a project and add some code.

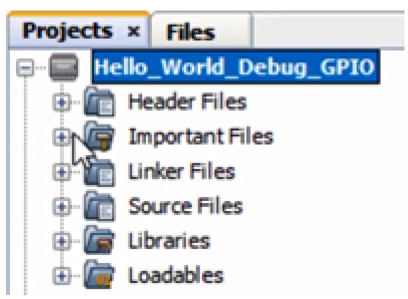


2.3.3 Create Example Project

Select *File>New Project* or the **New Project** icon  to open the Project wizard. Follow the steps below to create your project. Click **Next** to move to the next step.

1. **Choose Project**: Click on the "Microchip Embedded" category and then the "Standalone Project" project.
2. **Select Device (and Tool)**: Enter the device "AVR128DA48." Then enter the tool "AVR128DA48 Curiosity Nano-SN: MCHPL#" where the tool serial number (SN) contains the prefix "MCHP" followed by a multi-digit number.
3. **Select Compiler**: Under *Compiler Toolchains>XC8*, Select the most-current compiler version.
4. **Select Project Name and Folder**: Name your project. For example, "Hello World Debug GPIO." For Windows OS, the default project folder is `C:\Users\<UserName>\MPLABXProjects`.

After clicking **Finish**, the project tree should appear in the Projects window.



2.3.4 Create an Application

Create an application by adding source code to the project.

2.3.4.1 Create a New Main Source File

To add a new source file to the project:

1. Right click on the project source folder and select *New>Other*.
2. In the "New File" wizard, "Choose File Type," select the category *Microchip Embedded>XC8* and the file type `avr-main.c`.
3. Under "Name and Location," change the file name to `main.c`. Then click **Finish**.

The new file will open in an Editor window. By default the code will look like this.

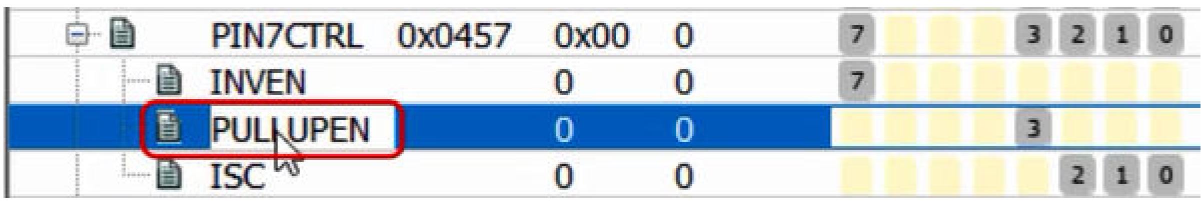
```
/*
 * File:   main.c
 * Author: Microchip Technology Inc.
 *
```

```
 * Created on May 19, 2020, 1:37 PM
 */


#include <avr/io.h>

int main(void) {
    /* Replace with your application code */
    while (1) {
    }
}
```

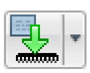
2.3.4.2 Add Code for a Pullup on PC7

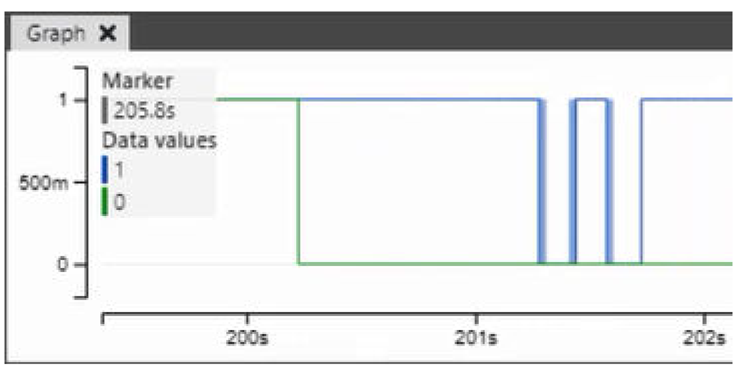
Code for enabling a pull-up on PortC, Pin 7, will be added to the source code. The View IO window is helpful in locating the correct register selection.

Figure 2-9. View IO Window




```
#include <avr/io.h>

int main(void) {

    PORTC.PIN7CTRL = PORT_PULLUPEN_bm; /* Enable PC7 Pullup */
    PORTC.DIR = PIN6_bm; /* Turn on LED */

    while (1) {
    }
}
```

Program the part  to see the better response of the PC7 switch and the turning on of the PC6 LED.



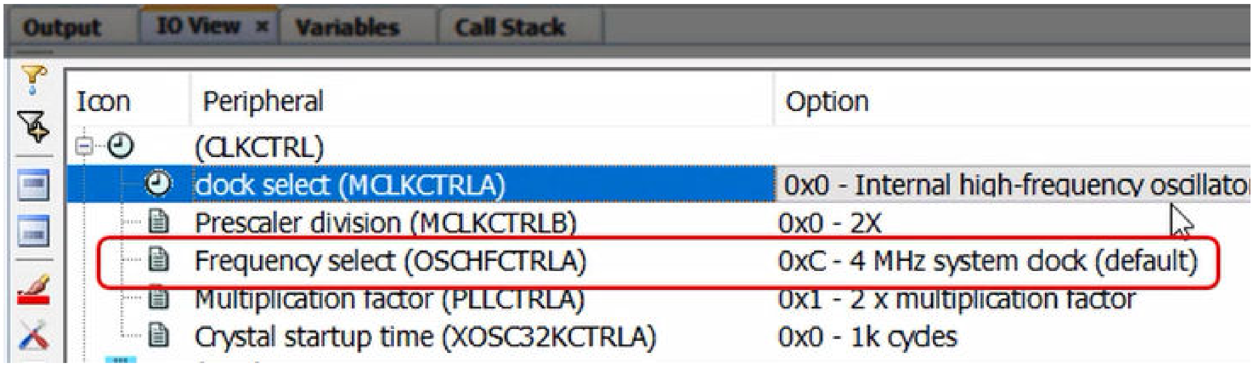
2.3.4.3 Add Code to Toggle LED with a Delay

Previously code was added to the main source file to enable a pull-up on PC7 for better switch response and the user LED was turned on. Now code is added to toggle the LED on and off. Due to the speed of the device this could be imperceptible unless a delay is added to slow down the toggle. Information on the delay used can be found at:

www.nongnu.org/avr-libc/user-manual/group__util__delay.html

A `#define` for the CPU frequency needs to be added as well as a `#include` for delay support.

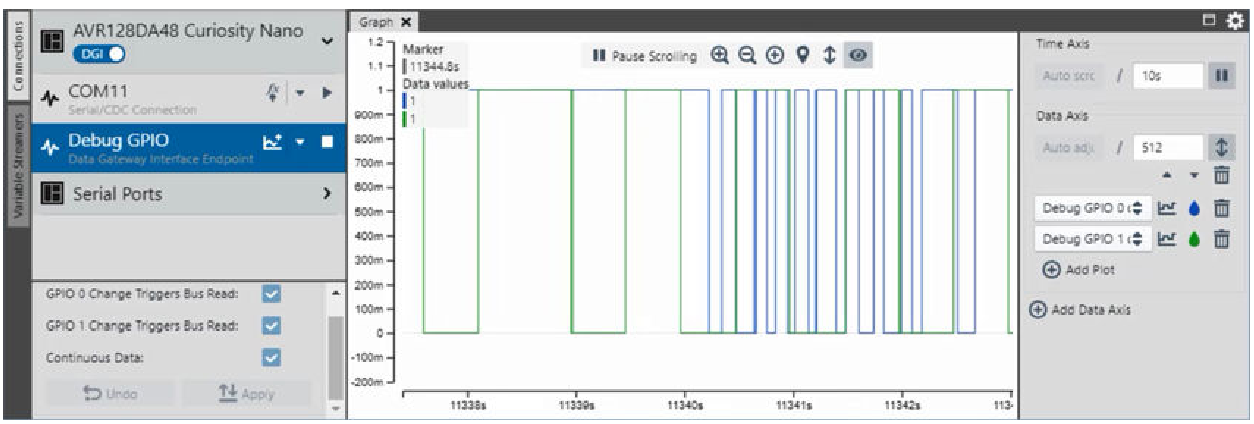
To determine the `F_CPU` for the Curiosity Nano, open the View IO window again and perform a debug run of the code to get live values in the window. Once the code is in the `while(1)` loop, Pause and look at CLKCTRL to find the value.



The updated code will be as shown below.

```
#include <avr/io.h>
#define    F_CPU    4000000UL
#include <util/delay.h>

int main(void) {

    PORTC.PIN7CTRL = PORT_PULLUPEN_bm; /* Enable PC7 Pullup */
    PORTC.DIR = PIN6_bm; /* Turn on LED */

    while (1) {
        PORTC.OUTTGL = PIN6_bm; /* Toggle LED on/off */
        _delay_ms(500); /* wait between toggles */
    }
}
```

Plotting the GPIO pins again, you can see PC6 (GPIO 1) in green with pulses from toggling and PC7 (GPIO 0) in blue with smaller pulses from button presses.



2.3.5 Plot Configurations

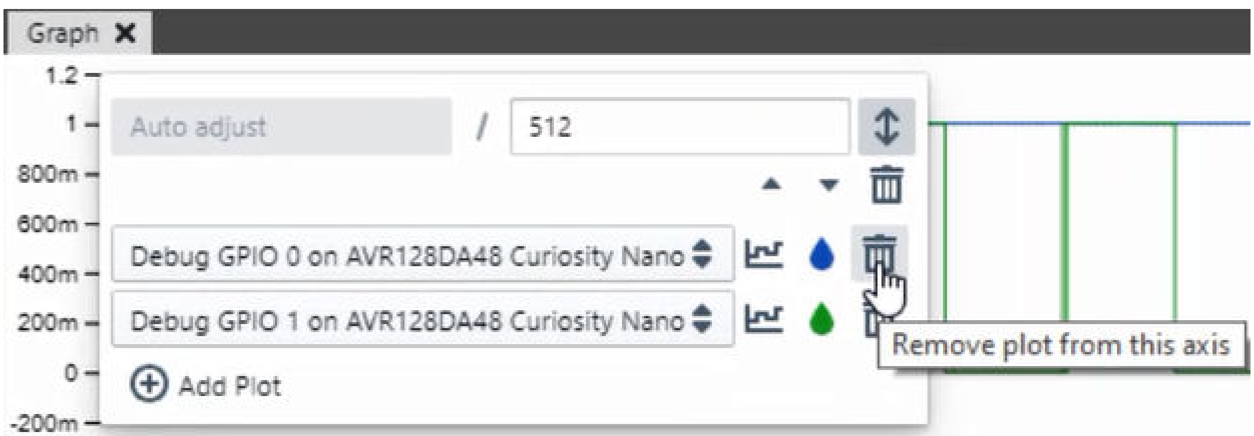
The project code produces outputs on both GPIO pins, which creates a crowded display on a single axis, even with color-coded plots. It would be better to have each plot on its own axes. Once the plots are each on an axis, it is easy to view the effect GPIO options have on the plots.

2.3.5.1 Move Each Plot to an Axis

For this example, there are only two plots on one axis, but the same procedure would work for more.

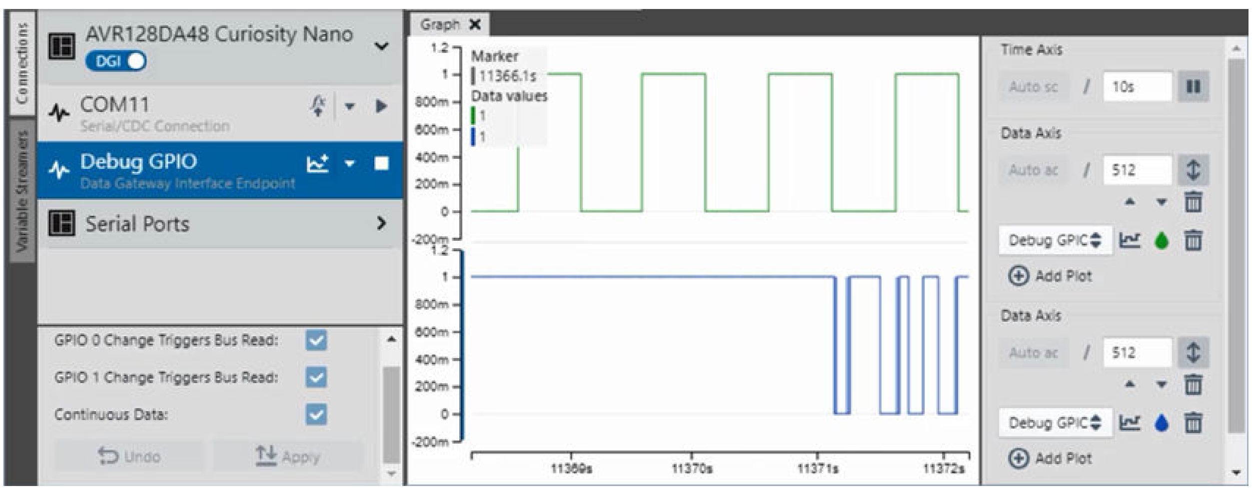
First add another axis to the Graph. An empty axis will appear beneath the original.

Second, delete one plot from the original axis.



Third, add the deleted plot to the new axis.

Finally, each plot will be on its own axis and will be much more visible.



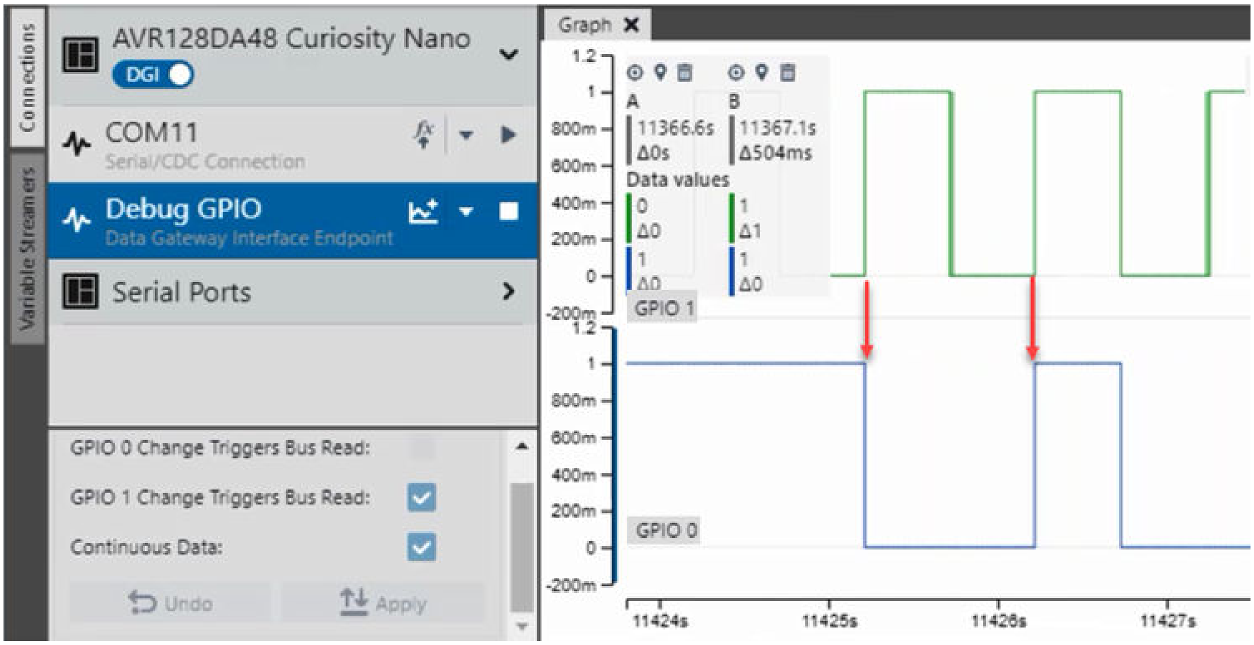
2.3.5.2 Debug GPIO Options

When the Debug GPIO data source is selected, options are visible beneath. All options are enabled by default.

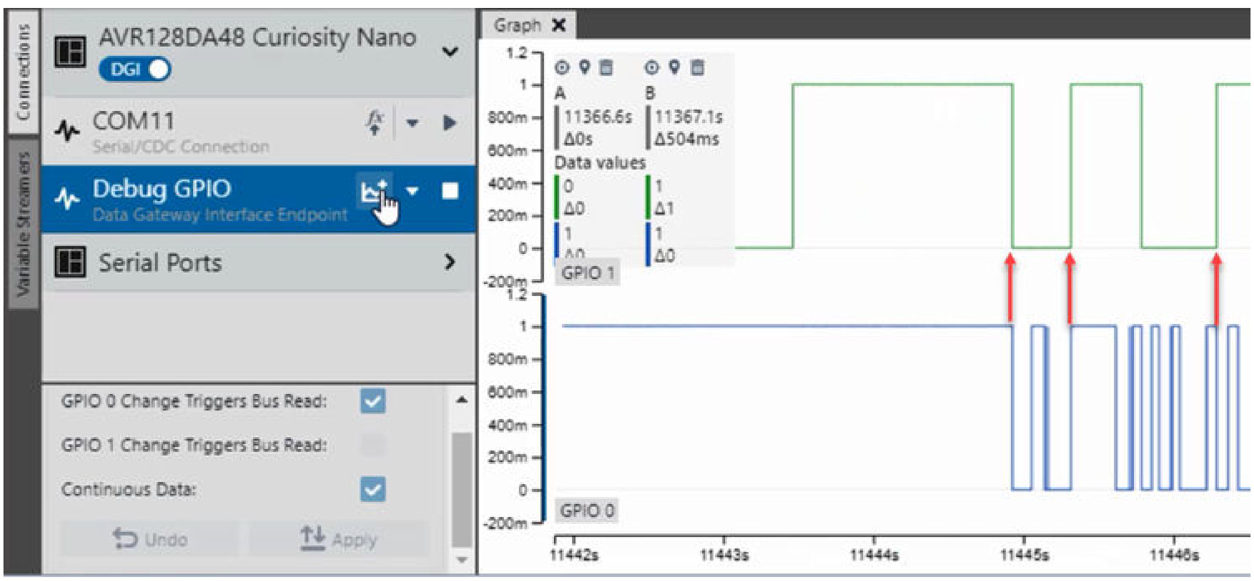
- GPIO 0 Change Triggers Bus Read
- GPIO 1 Change Triggers Bus Read
- Continuous Data

“GPIO x Change Triggers Bus Read” means that whenever there is a change on GPIO x, the GPIO bus is read and data displayed on the plots.

If “GPIO 0 Change Triggers Bus Read” is disabled and “GPIO 1 Change Triggers Bus Read” is enabled, the bus will only be read when GPIO 1 toggles, meaning that even for a quick button press, only changes that occur between GPIO 1 toggles will be displayed.



If "GPIO 0 Change Triggers Bus Read" is enabled and "GPIO 1 Change Triggers Bus Read" is disabled, the bus will only be read when GPIO 0 changes with a button press, meaning that even though GPIO 1 toggles at a consistent rate, only changes that occur when GPIO 0 changes will be displayed.



3. User Interface

In the default configuration the MPLAB Data Visualizer user interface is made up of the areas discussed in the following sections.

3.1 Data Sources (Left) Pane

The Data Sources pane of the MPLAB Data Visualizer is for identifying the sources of data and setting up data display.

Related Links

4. External Connections
5. Variable Streamers

3.1.1 Connections Tab

Click the **Connections** tab to see available connections. After you connect a target to the PC, that connection will be displayed and the on-board data sources will be listed.

To select a data source, such as "SPI," click on it and its settings will be available for editing. If you are connected and wish to change the settings, you can disconnect by clicking the **Stop Streaming** button. Click the button again to reconnect.

For connection details, see 4. External Connections.

Table 3-1.

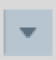
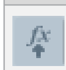
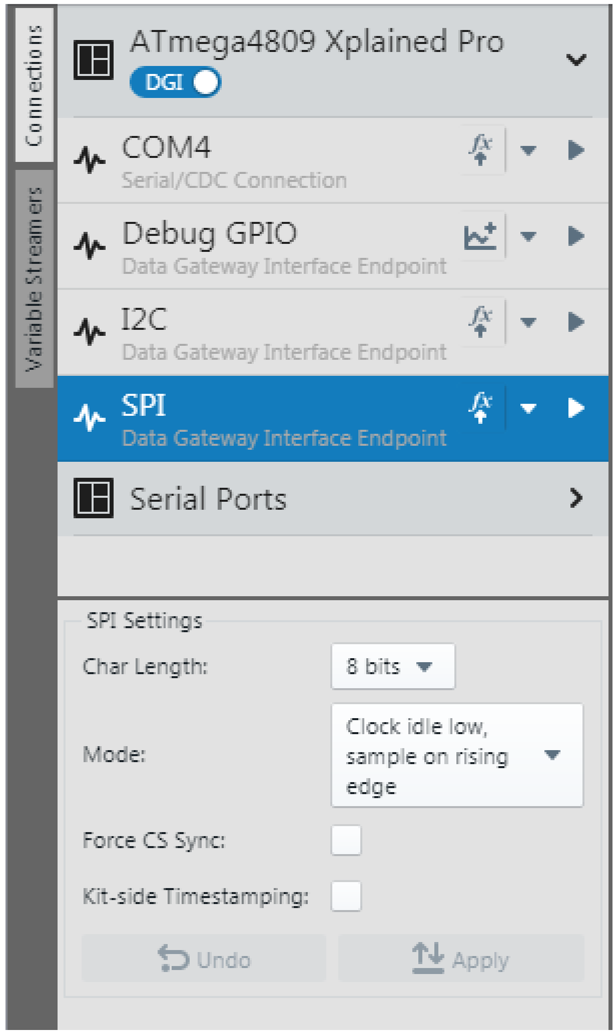
Button	Action
	Start/stop streaming - Toggle button: Begin/end streaming data from the data source. The Graph and Terminal window will continue to scroll.
	Plot - Choose plot type based on data source: Plot raw data, plot all pins, plot a variable streamer, or send data to the terminal. Also, remove data source from all plots.
	New Variable Streamer - Opens wizard to set up a new variable streamer.
	Plot All Pins - Plot all GPIO pins on graph.

Figure 3-1. Connections Tab



3.1.2 Variable Streamers Tab

Click the **Variable Streamers** tab to create, edit or delete variable streamers. **Import Variable Streamer** is a way to import `.ds` or `.txt` files that were used by the Atmel Studio Data Visualizer. More details on file format can be found in the document below, Section 5.1.1 "Configuration Format."

ww1.microchip.com/downloads/Secure/en/DeviceDoc/40001903B.pdf

To view decoding statistics, streaming data information in real-time, click on the eye icon.

When there are variables in a data stream, the stream must be decoded before the variables can be visualized. For details, see 5. Variable Streamers.

Figure 3-2. Variable Streamers Tab

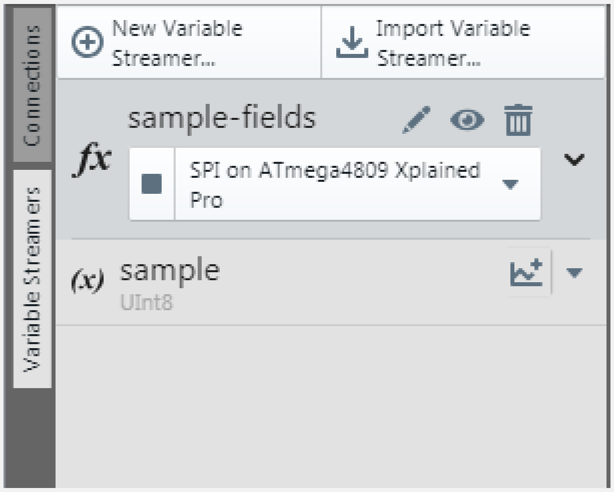
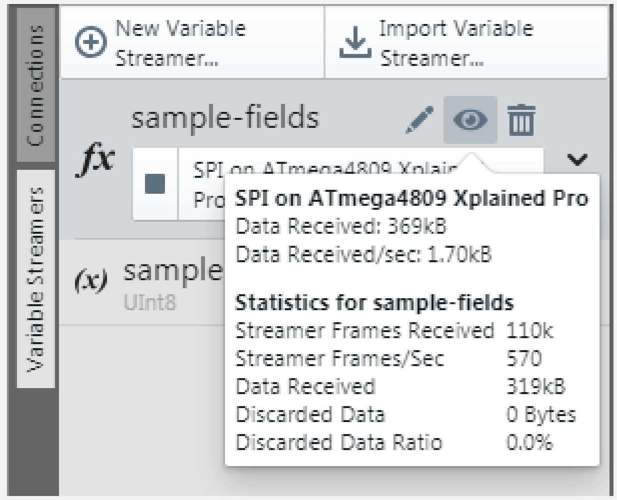


Figure 3-3. View Decoding Statistics

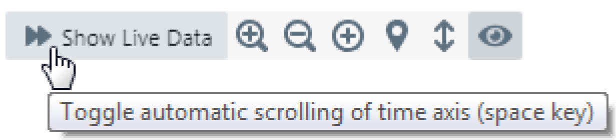


3.2 Graph Window

The Graph window shows the data plot(s) and provides tools for data analysis. Use the Graph Visualization Controls (right) pane to select Axis options and add multiple source axes.

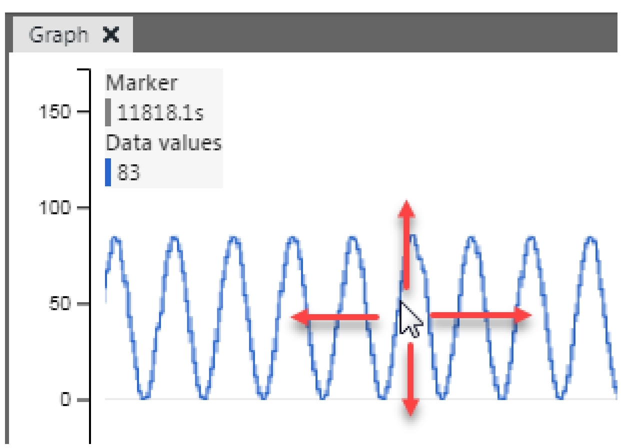
3.2.1 Plot Scrolling

If plot scrolling is not active, data might not be visible even though streaming is enabled. Hover over the top of the graph and click on "Show Live Data" to start plot scrolling. Once selected, the text changes to "Pause Scrolling" to stop plot scrolling.



Alternately, double click or hit the Space bar to stop the data scroll (although data streaming continues in the background). Double click or hit Space again to resume the timeline.

To manually move a plot, click on it and drag. Click on the left side of the plot to drag and pause the data axis. Click on the plot or in the center of the pane to drag and pause the time axis.



3.2.2 Zoom In and Out

Click on an axisto use the following controls:

- Use the mouse wheel to zoom in and out on the axis. The plot will resize accordingly.
- Use the mouse wheel to zoom in and out in the graph area.
- Hover over an axis to see options to zoom in and out.
- Hover over the top of the graph to zoom in and out.

Figure 3-4. Axis Zoom

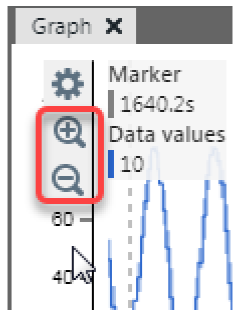


Figure 3-5. Graph Zoom

3.2.3 Add Data Axes

To add another data axis, hover over the top of the graph and select the icon to add an axis.

Set up axis properties on the Visualization Controls (right) pane or hover over the axis and click on “Axis Options.” On Visualization Controls is another selection to add a data axis. Both methods provide a control to add a plot to the axis.

Figure 3-6. Visualization Controls - Time Axis

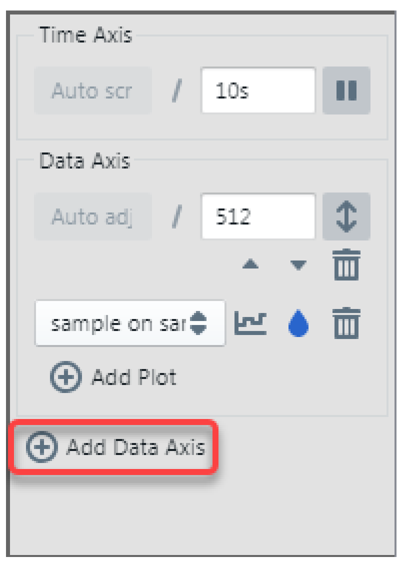
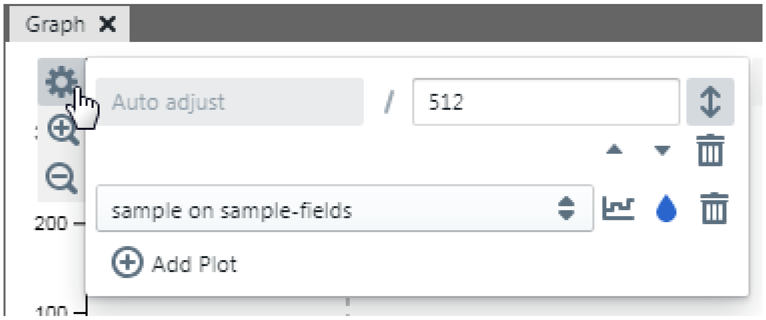
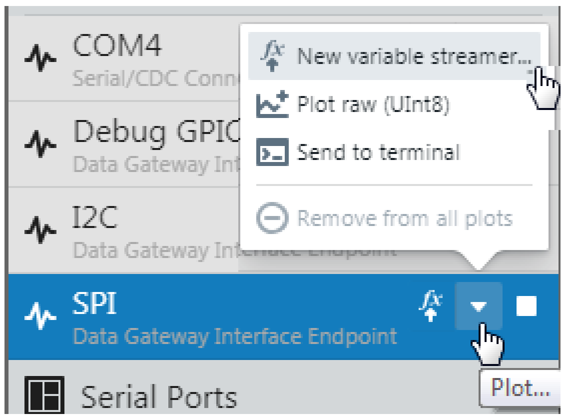


Figure 3-7. Axis Options



In addition, once the axis is set up, data can be plotted from the Data Sources pane by clicking the “plot” button on the selected data source.

Figure 3-8. Data Sources - Plot



3.2.4 View Plot Values

MPLAB Data Visualizer has the following built-in tools for viewing and analyzing plot data.

3.2.4.1 Graph Marker

The graph marker is a dashed vertical line that follows mouse movements along the time axis. Time and data values for the current location of the marker are displayed in the top left corner of the graph. To toggle this display, hover over the top of the graph and click on the “Inspect Values” icon.

Figure 3-9. Marker and Inspect Values Display

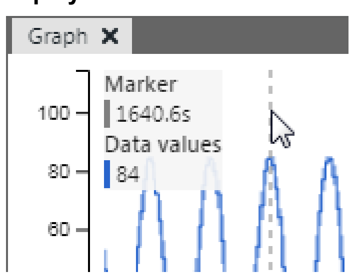


Figure 3-10. Toggle Inspect Values Display

3.2.4.2 Graph Cursors

A vertical cursor can be used in a similar manner to the graph marker, except that the cursor does not follow mouse movements; it must be dragged to a position and it will not move until dragged again. Inspect Values” displays cursor values in place of marker values.

Figure 3-11. Add Vertical Cursor

Figure 3-12. Vertical Cursor A

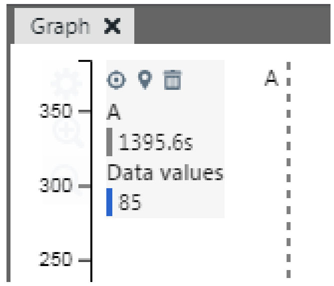


Table 3-2. Inspect Values Display

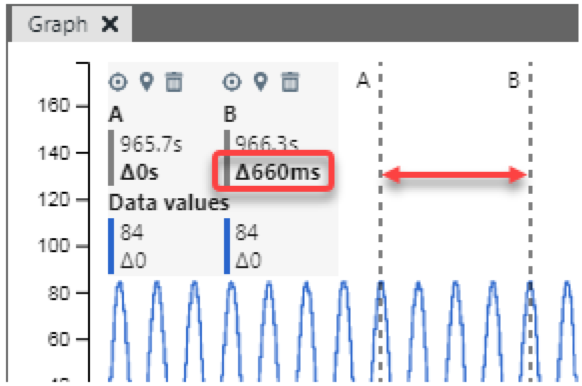
	Jump to cursor. The cursor will be centered on the graph and scrolling is paused. If the cursor has scrolled away with the plot, jump to its location.
	Reposition the cursor to the center of the graph. Hold the shift key when repositioning to move all cursors the same amount.
	Delete the cursor.
A, B, C, etc.	The letter number of the cursor.
Time Values	For each cursor, the value where the cursor intersects the time axis is displayed (timestamp). For two or more cursors, a time difference (delta) is displayed, with the leftmost cursor as reference.
Frequency Value	For two or more cursors, hover over the time values to see a frequency value, with the leftmost cursor as reference.
Data Values	For each cursor, the value where the cursor intersects the data plot is displayed. If there is more than one plot, a color bar corresponding to the plot color will signify the associated data value. For two or more cursors, a data difference (delta) is displayed, with the leftmost cursor as reference.

Use Two Cursors for Bandwidth

Two vertical cursors can be used to determine bandwidth. Using the time delta, for example in the figure below, the time difference between the position of A and of B is 660 ms. Therefore the bandwidth is 660 ms / 4 cycles ≅ 165 ms/cycle.

Note: As there may be variation between cycles, it is usually best to measure time over several cycles to provide an average value.

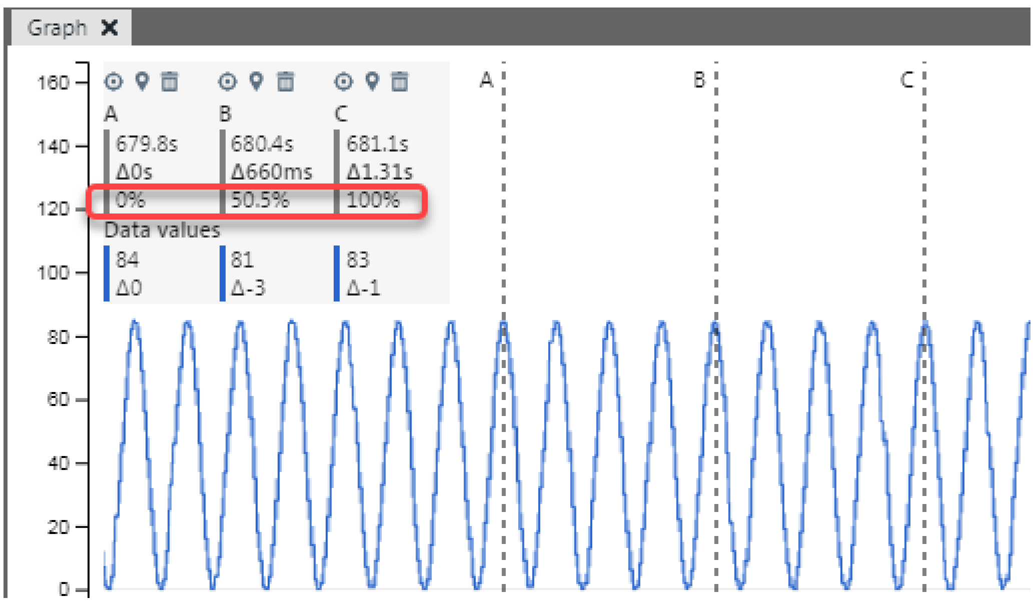
Figure 3-13. Add Two Vertical Cursors - Bandwidth



Use Three Cursors for Duty Cycle

Adding a third cursor allows you to calculate the duty cycle. If A-C is the period, then A-B is shown as a percentage of that (50.5%).

Figure 3-14. Add Three Vertical Cursors - Duty Cycle



Additional cursors may be added to the graph.

3.2.5 Automatically Adjust Data Axes

To toggle an automatic adjustment of all data axes, hover over the top of the graph and select the auto adjust icon. Also, double clicking on an axis will toggle auto adjust for that axis.

Figure 3-15. Before Auto Adjust

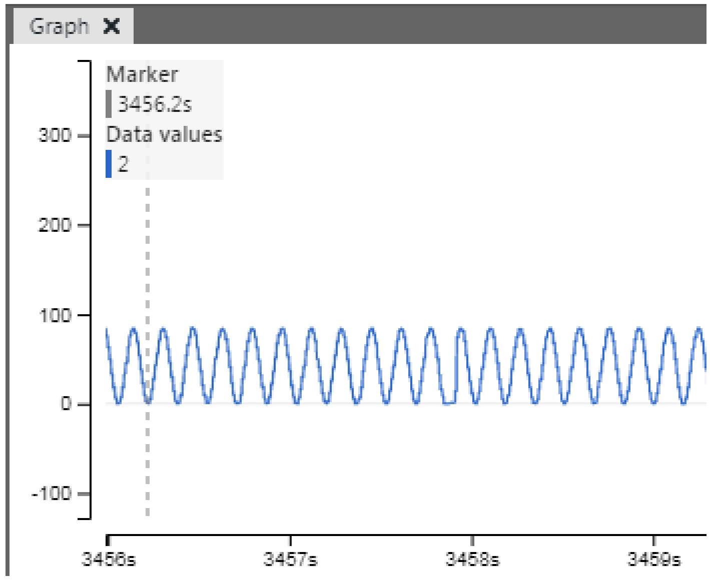
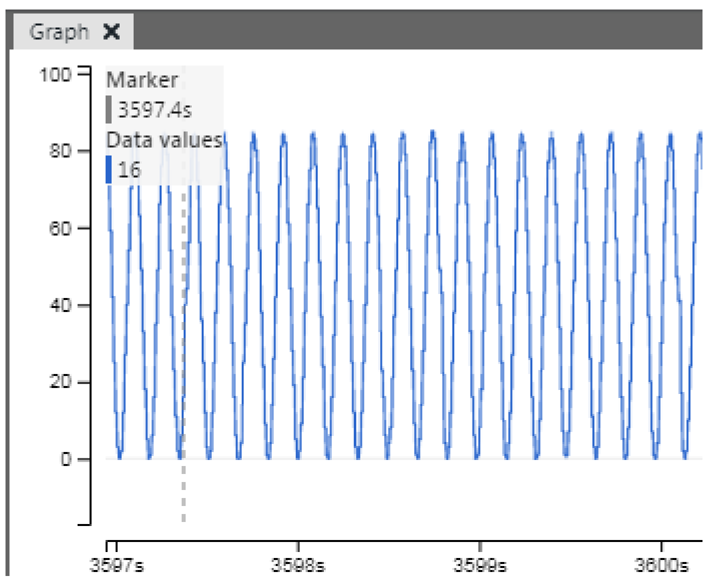


Figure 3-16. After Auto Adjust



3.3 Graph Visualization Controls (Right) Pane

The Graph Visualization Controls (right) pane is for controlling visualization (graphing) of streaming data.

Figure 3-17. Graph Visualization Controls

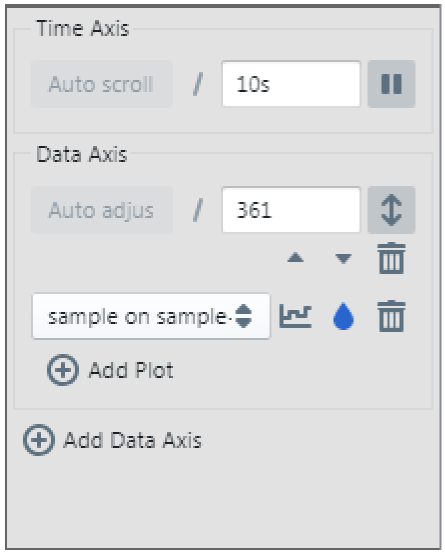


Table 3-3. Time Axis

Control	Description
Offset	When scrolling, “Auto scroll” displayed. When paused, shows the time at the right side of the graph.
Scale	Specify the resolution of the time axis in seconds.
	Pause scrolling or start/continue scrolling

Table 3-4. Data Axis

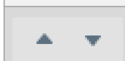
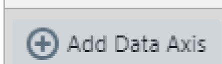
Control	Description
Offset	When auto adjust enabled, “Auto adjust” displayed. When auto adjust disabled, current offset of plot shown.
Scale	Specify the resolution of the data axis in seconds.
	Auto adjust enable/disable. When enabled, automatically adjust range of axis. When disabled, manually adjust range of axis.
	If more than one data axes, move this axis up or down relative to others.
	Click to delete this axis from the graph.
	Add another data axis to the graph under current axes.

Table 3-5. Data Axis - Plot Source and Format

Control	Description
Data Source	Select the data source to plot from the drop down list. See the Data Sources pane for selection and setup.

...........continued

Control	Description
	Click to select how data points are shown on the graph. • Connect points with stepped lines (default) • Connect points with straight lines • Only draw points, not lines
	Click to select a graph color.
	Click to delete this plot.
	Add another data source to plot on the current axis.

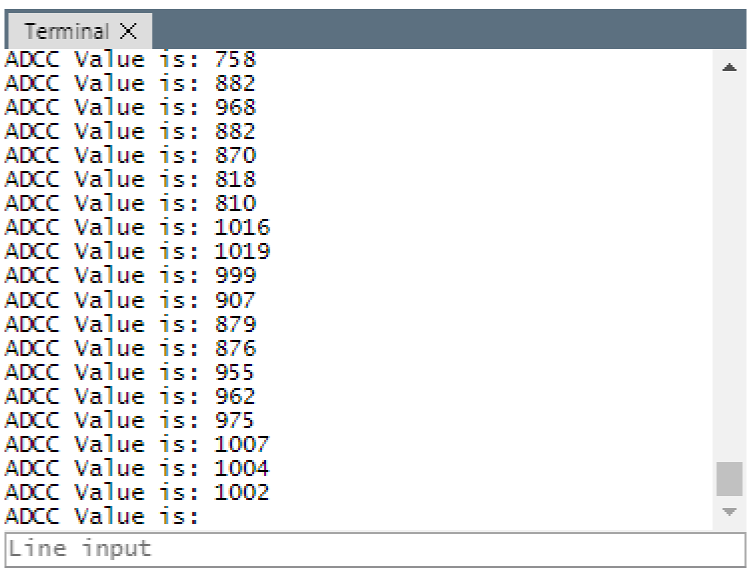
3.4 Terminal Window

The Terminal window of the MPLAB Data Visualizer shows streaming data in different formats. Use the Terminal Visualization Controls (right) pane to select one or more formats from the drop-down list.

When enabled, data will scroll continuously.

The terminal can also stream characters and lines of text to a target when connected through a COM port by typing either in the terminal area or the text box below.

Figure 3-18. Terminal Window with Data



You can toggle the data streaming view by hovering over the top of the Terminal window and clicking “Pause Scrolling/Scroll to End”. Although the window view is paused, data continues to stream in the background.

3.5 Terminal Visualization Controls (Right) Pane

In the Terminal area, the Terminal Visualization Controls (right) pane is for selecting the source and controlling the format of streaming data.

Figure 3-19. Terminal Visualization Controls

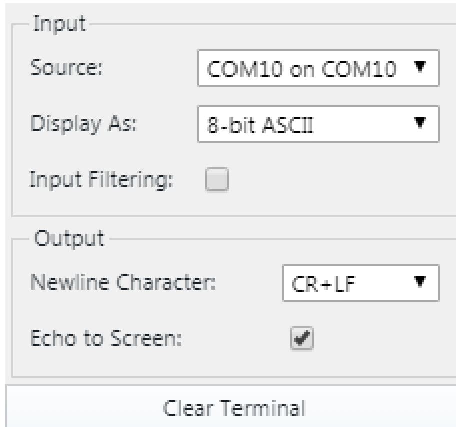


Table 3-6. Input Data

Control	Description
Source	All data sources, apart from Debug GPIO, are supported.
Display As	Select how the data stream is translated to terminal characters. Current selections are: • ASCII • UTF-8 • Hex values
Input Filtering	Click the checkbox to filter out ANSI/VT100 terminal control characters 1B, 90, 98, 9B, 9D, 9E and 9F from the input stream, as these have special meaning to the embedded terminal component.

The controls in the output section are only enabled for a COM port connection.

Table 3-7. Output Data

Control	Description
Newline Character	Select which character(s) will represent a newline in the output stream. • None • CR+LF: Carriage return + line feed • LF: Line feed
Echo to Screen	Check to echo typed characters to the screen.

To delete the content of the terminal window, click **Clear Terminal**.

3.6 Toolbar Controls

Visualizer toolbars, on the top of the interface, have the controls listed in the table below.

Table 3-8. Toolbar Controls

Control Image	Control	Description
	Clear Workspace	Clear data and settings in the workspace. All streaming of data will be stopped.
	Load Workspace	Load data and settings from a previous session into the visualizer.
	Save Workspace	Save data and settings from your current session into a file.
	Show Graph	If the Graph has been closed by clicking the “x” next to it, you can use this control to open it again.
	Show Terminal	If the Terminal has been closed by clicking the “x” next to it, you can use this control to open it again.
	Documentation	Show web help for the visualizer.
	Use Dark Theme	Enabled: Workspace background is black Disabled: Workspace background is white

3.7 Standalone Menus

If MPLAB Data Visualizer is used as a standalone application, a menu bar with be available with menus and items of similar function to MPLAB X IDE menu items. Basic text editor functions are included to allow editing of, for example, protocol definition (`.ds`) files.

Menu>Item	Items and Descriptions
File>*Items*	Basic File menu items. Choices are: New File, Open (Recent) File, Exit File.
Edit>*Items*	Basic Edit menu items. Choices are: Undo/Redo, Cut/Copy/Paste, Delete, Find/ Replace.
View>*Items*	Basic View menu items. Choices are: Editors, Split, IDE Log, Toolbars, Show Only Editor, Full Screen.
Tools>Embedded	Select installed embedded tools.
Tools>Plugins	Open the Plugins dialog to add, delete or manage plugins.
Tools>Options	Select data visualizer options: • General - web and proxy options • Keymap - keymapping options • Appearance - interface appearance options • Miscellaneous - File and output font/color options
Window> *Items*	Basic Window menu items. Choices are: Favorites, Output, Editor, IDE Tools (Notifications or Processes), Configure (size, float, dock, split, etc.), Reset, Close, Close all Documents, Close Other Documents, Document Groups, Documents.
Help>*Items*	Basic Help menu items. Choices are: Help Contents, Online Docs and Support, Keyboard Shortcuts Card, Check for Updates, About.

4. External Connections

External connections refer to the hardware connections used between the target hardware and the PC. These connections define the type of communication between the target and the MPLAB Data Visualizer. The type of connection depends on the device support.

To connect the target to the PC: Follow the instructions for the device or demonstration board.

To select the connection the visualizer: On the Data Sources (left) pane:

- Serial/CDC Connections - selection specifies communication with any serial port on the system that can be set up using baud rate, parity, data bits and stop bits.
- DGI Tools - selection specifies communication with any tool that has the Data Gateway Interface. It is capable of input streaming communication over SPI, I2C, USART and Debug GPIO. The feature set varies by tool.

4.1 Serial Port

The Data Visualizer can be connected to a target board via a standard PC serial port. Set up serial controls in the Data Sources (left) pane.

Baud rate, Stop bits, and parity must be set to match the required settings for the communication partner. Serial port data is treated as unsigned 8-bit data in the Graph and Terminal.

Table 4-1. Configuration

Field name	Values	Usage
Baud rate	600-2000000	Baud rate of serial interface
Char Length	5, 6, 7, or 8 bits	Number of bits in each transfer
Parity	None, Even, Odd, Mark, or Space	Parity type used for communication
Stop bits	1, 1.5, or 2 bits	Number of Stop bits

Notes: Asynchronous serial protocols (e.g., UART protocols used by DGI USART and CDC Virtual COM port interfaces) use the following **baud rates**:

- 9600
- 19200
- 38400
- 57600
- 115200
- 230400
- 500000
- 1000000
- 2000000

Using any other baud rates will not work for protocols over asynchronous interfaces (DGI UART and Serial port/CDC Virtual COM port).

Difference between tty and cu ports

See the list article by Godfrey van der Linden posted on the Apple® Listserv at lists.apple.com/archives/darwin-dev/2009/Nov/msg00099.html.

4.2 Data Gateway Interface (DGI)

The Data Gateway Interface is available on most kits with an Embedded Debugger (EBDG). The visualizer DGI controls can communicate with a DGI device. Set up DGI controls in the Data Sources (left) pane.

All detected DGI devices are listed on the left pane. The available interfaces will be listed under **Connections**. To enable an interface, click on the name. The visualizer accepts streaming input from a DGI-capable board.

4.2.1 GPIO Interface

The GPIO interface contains the bit values of the enabled Debug GPIO pins. A packet of unsigned 8-bit data is transmitted every time a pin toggles. For further details on the physical part of the GPIO interface, see the user guide of the debugging tool to be used to sample the GPIO data.

On the Data Sources (left) pane, when the GPIO interface is selected, the GPIO settings are displayed on the lower section.

Table 4-2. Configuration

Field Name	Values	Usage
GPIO 0 Change Triggers Bus Read	ON, OFF	Monitor change on GPIO pin 0 to trigger a bus read
GPIO 1 Change Triggers Bus Read	ON, OFF	Monitor change on GPIO pin 1 to trigger a bus read
GPIO 2 Change Triggers Bus Read	ON, OFF	Monitor change on GPIO pin 2 to trigger a bus read
GPIO 3 Change Triggers Bus Read	ON, OFF	Monitor change on GPIO pin 3 to trigger a bus read

Important: When plotting the Debug GPIO data source, all GPIOs are sampled but only those GPIOs that have change triggers enabled will trigger a sample on change. For example, if GPIO n (n = 0,1,2) all have “GPIO n Change Triggers Bus Read” disabled, but GPIO 3 has this function enabled, then GPIO values will only be sampled when GPIO 3 changes; that is, all four GPIO values will be read only when GPIO 3 changes.

4.2.2 USART Interface

The USART **source** streams the raw values received on the USART interface. For further details on the physical part of the USART interface, see the user guide of the debug tool to be used to sample the USART data.

On the Data Sources (left) pane, when the USART source is selected, the USART settings are displayed on the lower section.

Note: Asynchronous serial protocols (e.g., UART protocols used by DGI USART and CDC Virtual COM port interfaces) use the **baud rates** listed in 4.1 Serial Port.

Table 4-3. USART Settings

Field Name	Values	Usage
Baud Rate	0-2000000	Baud rate for UART interface in Asynchronous mode
Char Length	5, 6, 7, or 8 bits	Number of bits in each transfer
Parity	None, Even, Odd, Mark, or Space	Parity type used for communication
Stop bits	1, 1.5, or 2 bits	Number of Stop bits

4.2.3 I2C Interface

The I2C **source** streams the raw values received on the I2C interface. For further details on the physical part of the I2C interface, see the user guide of the debug tool to be used to sample the I2C data.

The **I2C Configuration** options are displayed under the **I2C** interface in the **DGI** section of the left pane.

The I2C interface is under the **DGI** section of the Data Sources (left) pane. When an I2C connection is selected, the I2C settings are displayed in the lower section of this pane.

Table 4-4. I2C Settings

Field Name	Values	Usage
Speed	0	The expected operation speed of the interface in Hertz helps the slave device adjust the timings. Up to 400 kHz is supported.
Address	1	Address of the slave device.
Kit-side Timestamping	Check to enable.	Target timestamping

4.2.4 SPI Interface

The SPI **source** streams the raw values received on the SPI interface. For further details on the physical part of the SPI interface, see the user guide of the debug tool to be used to sample the SPI data.

Important: The SPI hardware module uses an active-low Chip Select (CS) signal. Any data sent when the CS pin is high will be ignored.

The SPI interface is under the **DGI** section of the Data Sources (left) pane. When an SPI connection is selected, the SPI settings are displayed in the lower section of this pane.

Table 4-5. SPI Settings

Field Name	Values	Usage
Char Length	5, 6, 7, or 8 bits	Number of bits in each transfer
Mode	• Clock idle normally low, Sample data on rising edge • Clock idle normally low, Sample data on falling edge • Clock idle normally high, Sample data on falling edge • Clock idle normally high, Sample data on rising edge	SPI mode, controlling clock phase and sampling.
Force CS Sync	Check to enable.	The SPI interface is only enabled after the Chip Select line has toggled twice.
Kit-side Timestamping	Check to enable.	Target timestamping

5. Variable Streamers

Most communication interfaces use streams of bytes to transfer data. This is enough for single data values of 8-bit precision, but when multiple values are required to be transmitted over the same interface, data must be packed in a protocol. The MPLAB Data Visualizer supports the **Data Stream** protocol.

The Data Stream protocol uses a light-weight framing format to pack several numerical values over one interface. It is only capable of handling incoming data and it only supports synchronous streams (i.e., every data packet must contain one sample from each data stream). **Data Stream Decoder** information resides in the visualizer workspace.

The visualizer data stream module takes an incoming raw data stream and splits it into multiple data streams. The data stream format is specified by the **Variable Streamer** you provide.

5.1 Variable Data Types

Variable Streamers are set up and plotted using a wizard. This set up is saved in the workspace.

Variables defined in a Variable Streamer must be of a type listed in the table below.

Table 5-1. Allowed Data Types

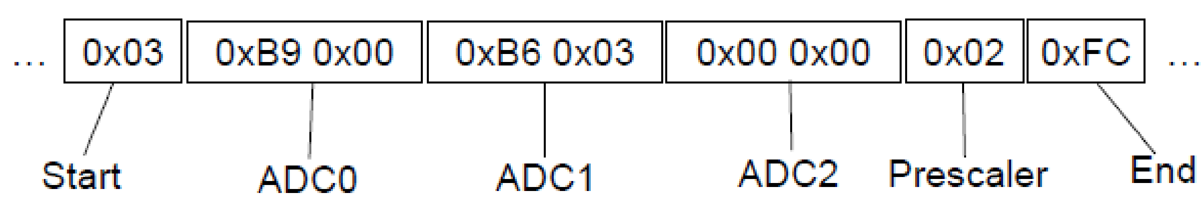
Type	Size (Bytes)
int8	1
int16	2
int32	4
uint8	1
uint16	2
uint32	4
float32	4
float64	8

5.2 Stream Format

The data stream Stream Format is processed in the same order as the variables specified in the Variable Streamer. All data must be given as little endian values, meaning that the lowest byte must be sent first. Additionally, a wrapper consisting of one byte before and one byte after the data stream variables must be added. This wrapper is used by the interpreter to synchronize to the data stream. The start byte can be of an arbitrary but the end byte must be the inverse of the Start byte. The start and end bytes are not defined in the configuration.

The figure below gives an example raw data transmission where ADC0 is 185, ADC1 is 950, ADC2 is 0, and Prescaler is 2.

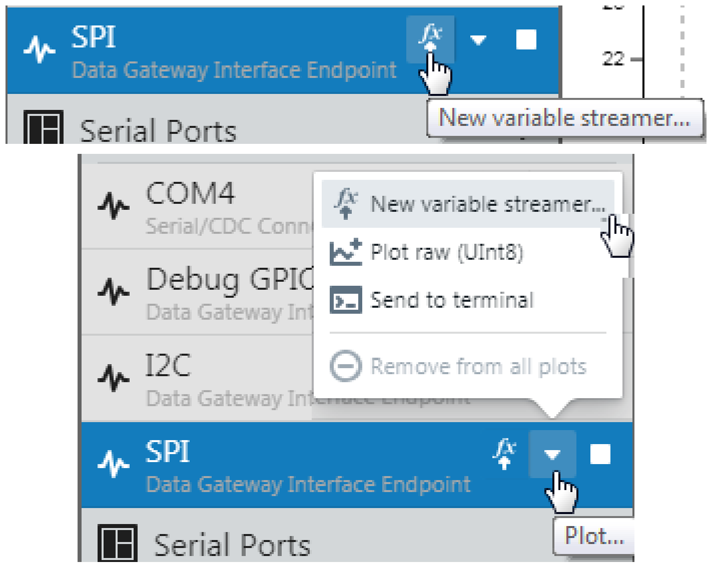
Figure 5-1. Data Streamer



5.3 Variable Streamer Setup and Plot

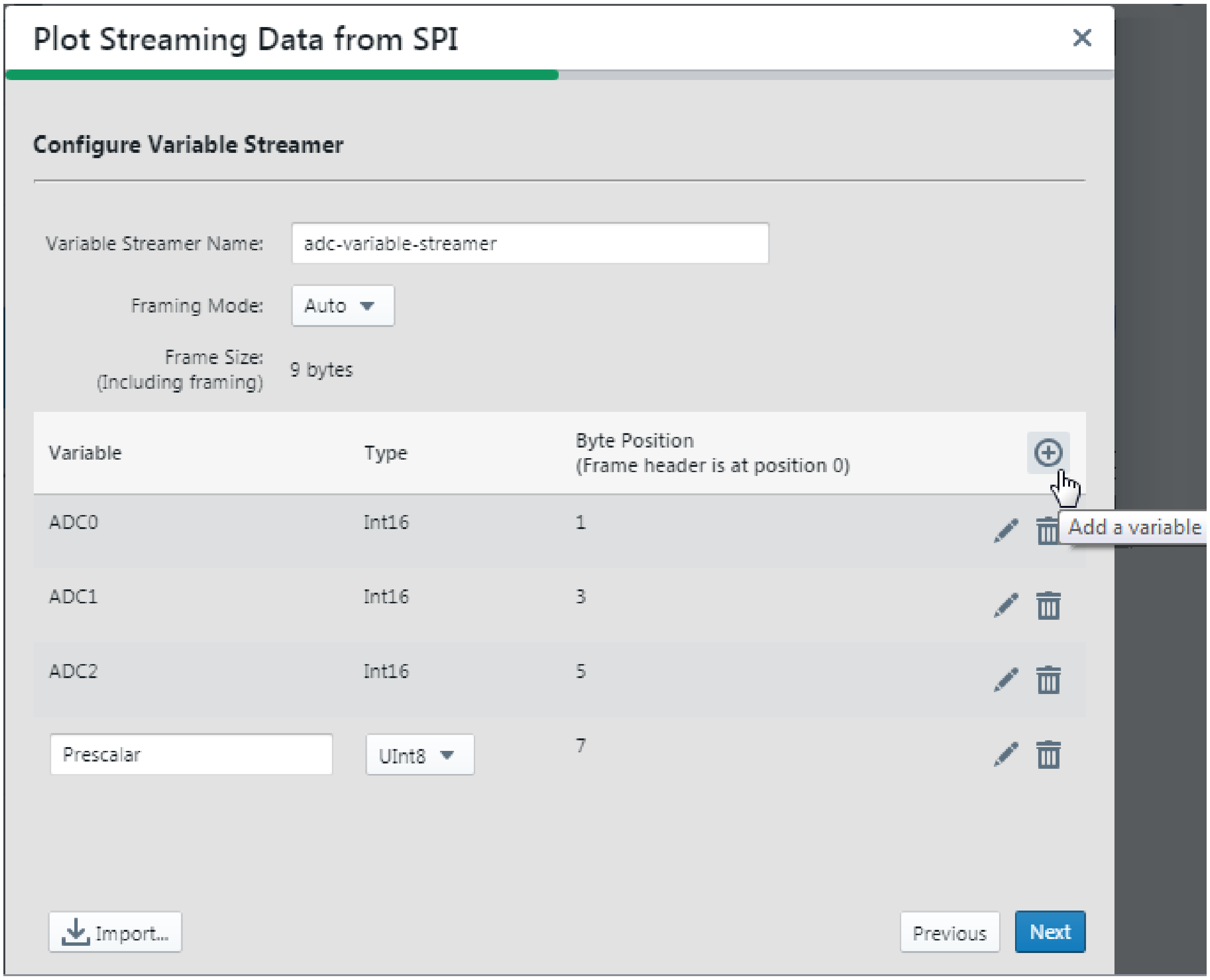
A Variable Streamer defines the variables that are embedded in a data stream as output from your application. The Variable Streamer is used in the configuration of a Data Stream Decoder instance. The output from a decoder instance are new data streams that can be visualized using plots on the graph.

To create a new Variable Streamer, go to the Data Sources pane, **Connections** tab. From the data sources available, select either “New Variable Steamer” or “Plot>New Variable Streamer”.

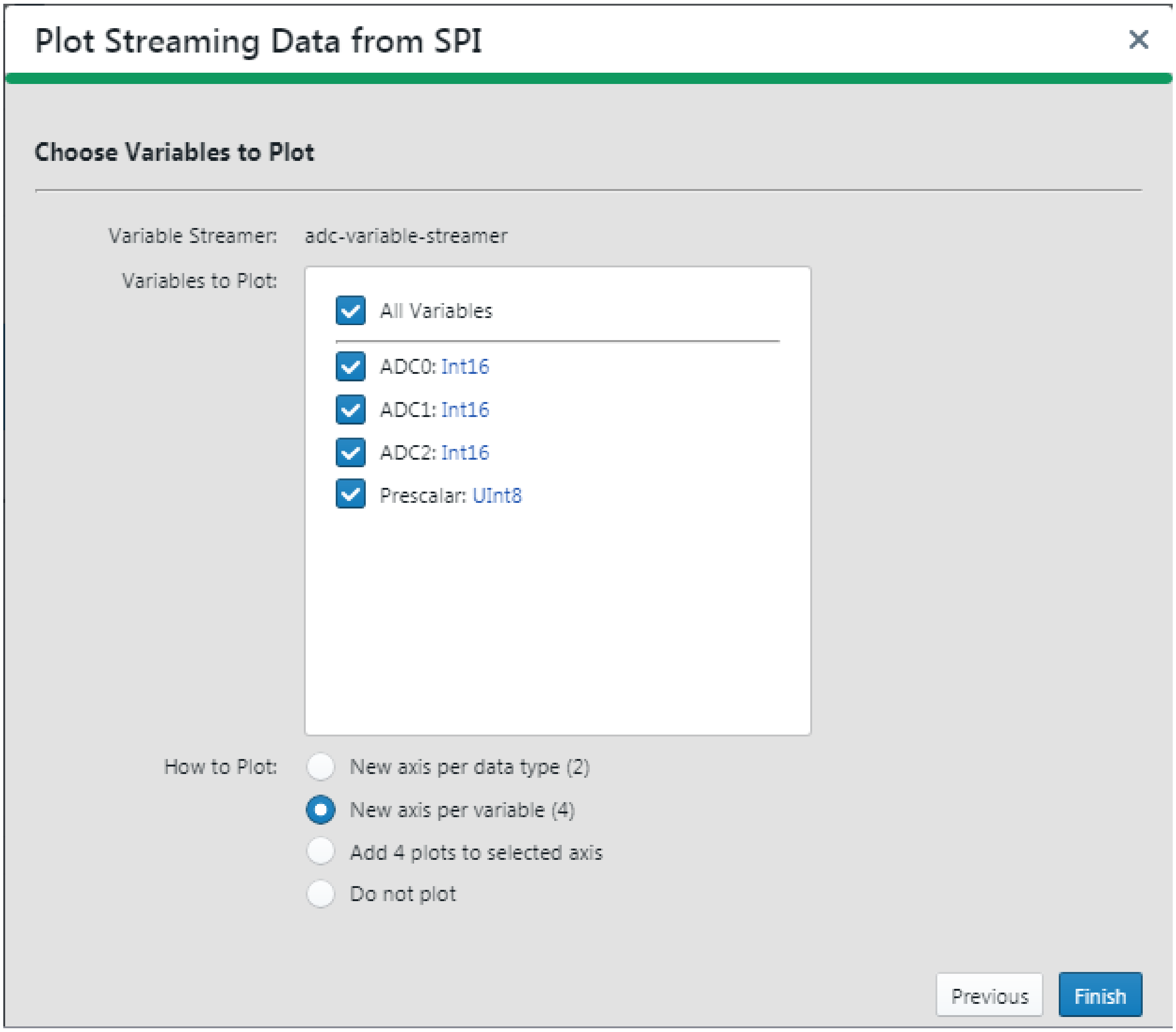


The **Plot Streaming Data** wizard will open to the **Configure Variable Streamer** window.

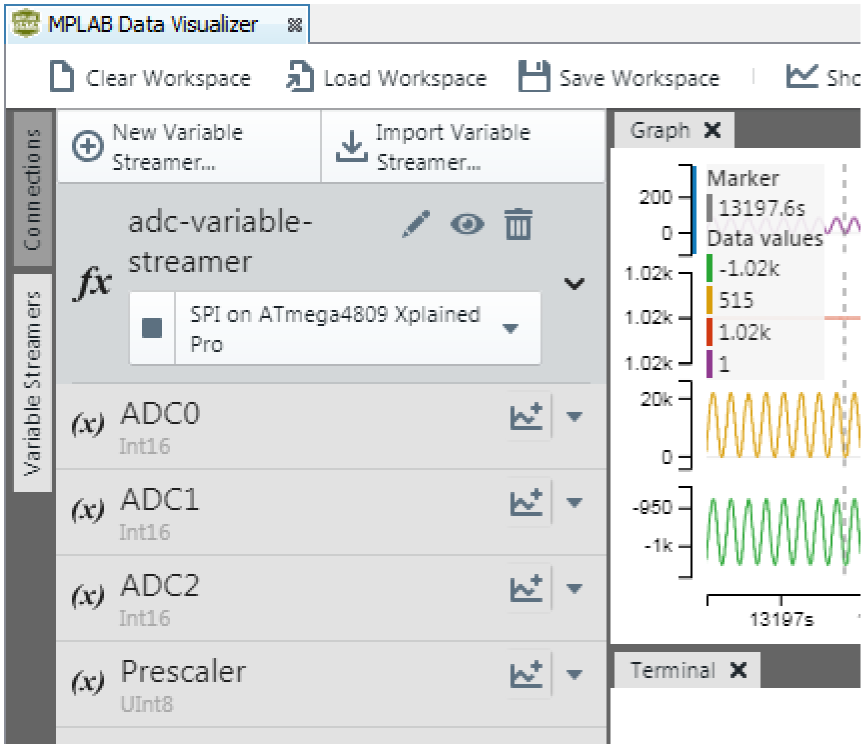
Option	Description
Variable Streamer Name	Choose a descriptive name for the variable streamer
Framing Mode	If the start byte of data streamer protocol is 0x5F and end byte is 0xA0, “Auto” can be used. For any other start byte and end byte pattern “one’s complement” can be used.
Framing Size	See 5.2 Stream Format.
Variable	Enter the name of a variable from application code.
Type	See 5.1 Variable Data Types.
Byte Position	See 5.2 Stream Format.



Click **Next** to **Choose Variables to Plot**. Select a plotting method and click **Finish**.

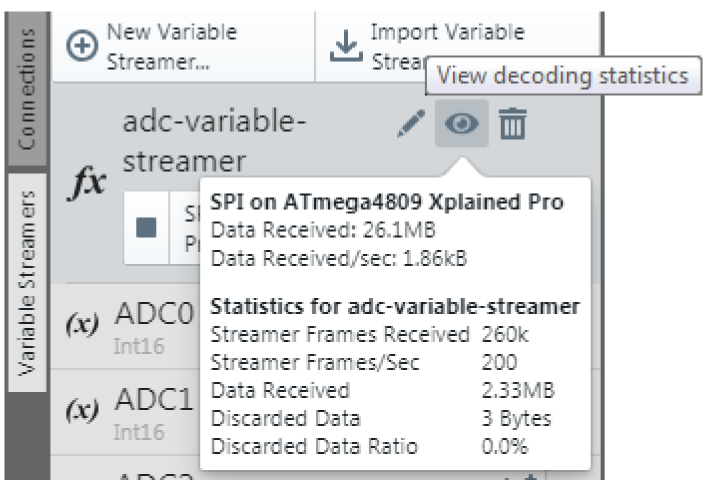


The new Variable Streamer will be shown on the **Variable Streamers** tab. To save your setup, save the workspace at the end of your session and then load the workspace for your next session.



5.4 View Statistics

To view statistics on streaming data, click on the eye icon to see the Variable Streamer data.



6. Troubleshooting

See an MPLAB Data Visualizer component below to find tips on troubleshooting related issues.

- Streaming
- Data Stream Protocol

6.1 Data Streaming

SPI over DGI - If the data waveform doesn't match expectations or can't be decoded by the Data Stream Decoder, try enabling "Force synchronization on CS". Sync issues are especially common when starting a debug session on an Xplained Pro development kit and then streaming over SPI from the same kit.

6.2 Data Stream Decoder

No Data Input

Make sure that your data source is connected and transmitting data. An easy way to verify this is to plot the raw data stream or display the data in the Terminal. If there are no values in the plot or in the terminal, then data is likely not being received.

Decoder Mismatch

If you're getting a Variable Streamer mismatch warning, make sure that the variables in the Variable Streamer match the incoming data exactly. The decoder expects a start of frame (SoF) byte, followed by a sequence of bytes that match the fields defined in the Variable Streamer, and an end of frame (EoF) byte which is a one's complement of SoF. The following situations would prevent a data packet from being decoded:

- Either the SoF or EoF bytes are not present
- Any of the fields are missing from a data packet transmission
- The size of the variable doesn't match that in the Variable Streamer

Each variable must have the number of bytes that match the data type. For instance, `Int8` would be one byte and `Int16` would be 2 bytes.

[Start of Frame character][Data Field 1 * data type size][Data Field 2 * data type size in bytes]...[Data Field n * data type size in bytes][End of Frame character]

Example:

For two `Int16` fields, x and y, the protocol byte sequence would look like this:

[SoF][x1][x0][y1][y0][EoF]

A data stream that matches this protocol might look like this:

[`0x3`][`0x34`][`0x12`][`0x78`][`0x56`][`0xFC`]

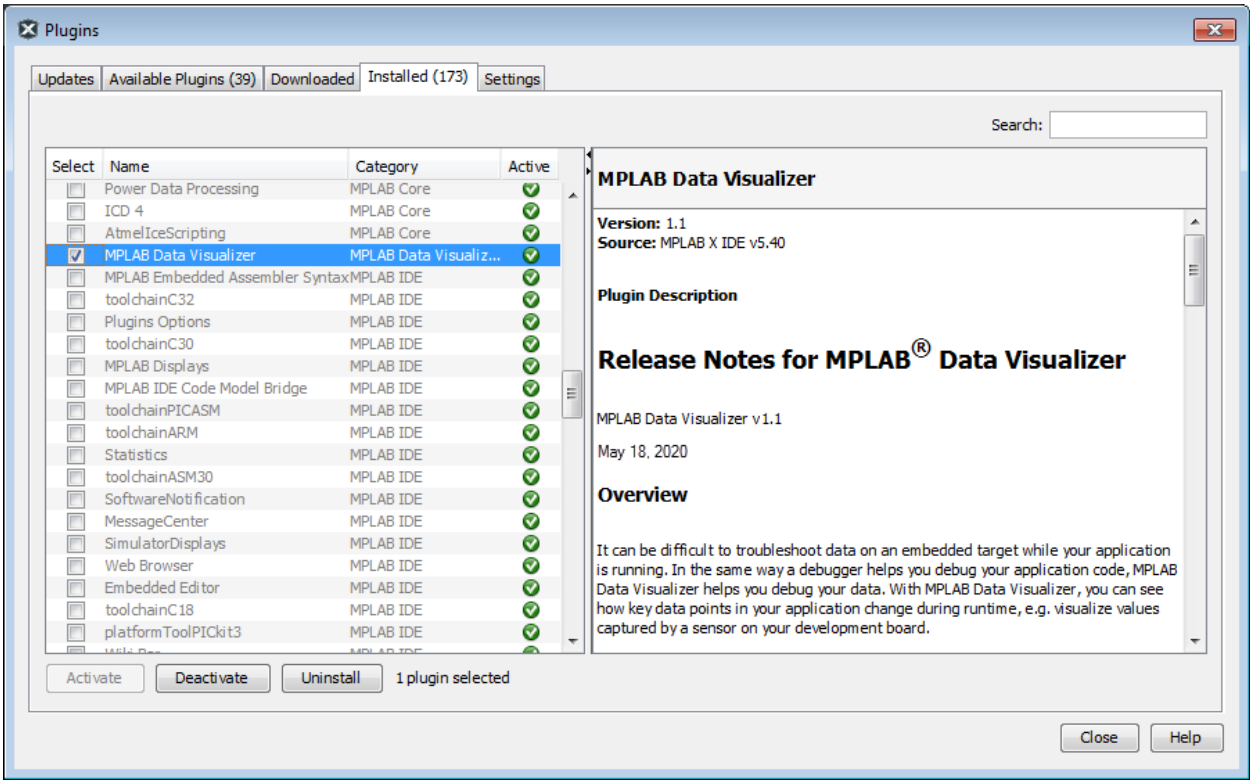
The decoded x and y values would be `0x1234`, `0x5678`.

7. Finding Release Notes

Release notes for the MPLAB Data Visualizer can be found:

- for the plugin, under *Tools>Plugins>Installed>MPLAB Data Visualizer*.
- for the standalone program, under the Help menu.

Figure 7-1. Plugin Release Notes



8. Example of Plotting Data - Code Listing

Example Header and C code for the ATmega4809 Xplained Pro project may be found in the following sections.

Note: Care should be taken when copying across pages, as the page footer may appear in the code listing.

8.1 C Header Code

configure.h

```
/*
 * File:   configure.h
 * Author: Microchip Technology Inc.
 *
 * Created on September 20, 2018, 11:00 AM
 */

#ifndef CONFIGURE_H
#define CONFIGURE_H

#ifdef __cplusplus
extern "C" {
#endif

void initializePeripherals();

#ifdef __cplusplus
}
#endif

#endif  /* CONFIGURE_H */
```

memutil.h

```
/*
 * File:   memutil.h
 * Author: Microchip Technology Inc.
 *
 * Created on September 19, 2018, 1:03 PM
 */

#ifndef MEMUTIL_H
#define MEMUTIL_H

#ifdef __cplusplus
extern "C" {
#endif

#define LEN(a) (sizeof(a) / sizeof(*a))

#ifdef __cplusplus
}
#endif

#endif /* MEMUTIL_H */
```

pins.h

```
/*
 * File:   pins.h
 * Author: Microchip Technology Inc.
 *
 * Created on September 19, 2018, 11:22 AM
 */

#ifndef PINS_H
#define PINS_H

#ifdef __cplusplus
extern "C" {
```

```
#endif

#define MISO_PIN  5
#define MOSI_PIN  4
#define CS_PIN  3
#define SCK_PIN  6

#ifdef __cplusplus
}
#endif

#endif  /* PINS_H */
```

spi.h

```
/*
 * File:   spi.h
 * Author: Microchip Technology Inc.
 *
 * Created on September 19, 2018, 11:21 AM
 */

#ifndef SPI_H
#define SPI_H

#ifdef __cplusplus
extern "C" {
#endif

void init_spi0(void);

void select_dgi_spi(void);
void deselect_dgi_spi(void);
void tx_spi0(uint8_t tx_usart1);
void tx_string_spi0(char* tx_string);
void tx_data_spi0(uint8_t tx_byte[], int length);

#ifdef __cplusplus
}
#endif

#endif  /* SPI_H */
```

timer_callback.h

```
/*
 * File:   timer_callback.h
 * Author: Microchip Technology Inc.
 *
 * Created on September 19, 2018, 11:15 AM
 */

#ifndef TIMER_CALLBACK_H
#define TIMER_CALLBACK_H

#ifdef __cplusplus
extern "C" {
#endif

void timer_callback();

#ifdef __cplusplus
}
#endif

#endif  /* TIMER_CALLBACK_H */
```

8.2 C Source Code

configure.c

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/cpufunc.h>
#include "timer_callback.h"
#include "spi.h"
#include "pins.h"

void init_sysclock(void);
void init_tcb0(void);

#define PORTB_PIN2CTRL  _SFR_MEM8(0x0432)

void initializePeripherals() {
    init_sysclock();
    init_tcb0();
    init_spi0();

    PORTB_PIN2CTRL |= 0x8;
}

void init_sysclock(void)
{
    CPU_CCP = 0xD8;
    CLKCTRL_MCLKCTRLB = 0x00;
}

void init_tcb0(void)
{
    TCB0.CTRLA = TCB_CLKSEL_CLKDIV2_gc;  // base clock 16Mhz / 2 = 8 MHz
    TCB0.CTRLB = 0x00;  // all the defaults
    TCB0.CCMP = 7999;  // 8Mhz / 8000 = 1 kHz
    TCB0.INTCTRL = 0x01;  // enable interrupt
    TCB0.CTRLA |= 0x01;  // enable timer
}
```

main.c

```
/*
 * File:   main.c
 */

#include <stdio.h>
#include <stdlib.h>
#include "configure.h"
#include <avr/interrupt.h>

int main(int argc, char** argv)
{
    initializePeripherals();
    sei(); //set global interrupt flag

    while(1) ;
}
```

sine_app.c

```
/*
* There are two waveforms in this application:
* 1. sine wave
* 2. triangle wave
*
* There are two global variables for control in this application:
*  - amp_factor - this defines the amplitude of the waveform
*  - wave_select - this defines the waveform selection.
*    It can either be 0 for sine or 1 for triangle.
*/

#include <stdio.h>
```

```
#include "timer_callback.h"
#include "spi.h"
#include "memutil.h"

int amp_factor = 1;
int wave_select = 0;
int counter = 0;

int sine[] = {
    0x2b,0x2d,0x30,0x32,0x35,0x38,0x3a,0x3d,
    0x3f,0x41,0x43,0x46,0x48,0x49,0x4b,0x4d,
    0x4e,0x50,0x51,0x52,0x53,0x54,0x54,0x55,
    0x55,0x55,0x55,0x55,0x54,0x54,0x53,0x52,
    0x51,0x50,0x4e,0x4d,0x4b,0x49,0x48,0x46,
    0x43,0x41,0x3f,0x3d,0x3a,0x38,0x35,0x32,
    0x30,0x2d,0x2b,0x28,0x25,0x23,0x20,0x1d,
    0x1b,0x18,0x16,0x14,0x12,0xf,0xd,0xc,
    0xa,0x8,0x7,0x5,0x4,0x3,0x2,0x1,
    0x1,0x0,0x0,0x0,0x0,0x0,0x1,0x1,
    0x2,0x3,0x4,0x5,0x7,0x8,0xa,0xc,
    0xd,0xf,0x12,0x14,0x16,0x18,0x1b,0x1d,
    0x20,0x23,0x25,0x28,0x2b
};

int tri_1k[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1};

struct
{
    int cnt;
    int *amp;
} waveform[] =
{
    {LEN(sine), sine},
    {LEN(tri_1k), tri_1k}
};

void timer_callback()
{
    uint8_t sample = (amp_factor * waveform[wave_select].amp[counter]) & 0x7F;
    if (++counter >= waveform[wave_select].cnt) {
        counter = 0;
    }

    select_dgi_spi();
    tx_spi0(0x03);
    tx_spi0(sample);
    tx_spi0(0xFC);
    deselect_dgi_spi();
}
```

spi.c

```
#include <avr/io.h>
#include "spi.h"
#include "pins.h"

void init_spi0(void)
{
    VPORTA.DIR |= (1 << MOSI_PIN) | (1 << SCK_PIN);
    VPORTF.DIR |= (1 << CS_PIN);

    SPI0.CTRLA = 0 << SPI_CLK2X_bp | 0 << SPI_DORD_bp | 1 << SPI_MASTER_bp |
SPI_PRESC_DIV64_gc;
    SPI0.CTRLB = (1 << SPI_SSD_bp);  // disable SS#
    SPI0.CTRLA |=  1 << SPI_ENABLE_bp;
}

void select_dgi_spi(void)
{
    VPORTF.OUT &= ~(1 << CS_PIN);
}

void deselect_dgi_spi(void)
{
```

```
    VPORTF.OUT |= (1 << CS_PIN);
}

void tx_spi0(uint8_t tx_byte)
{
    uint8_t tx_rdy = 0;

    SPI0.DATA = tx_byte;
    while(!tx_rdy)
        tx_rdy = (SPI0.INTFLAGS & SPI_IF_bm );
}

void tx_string_spi0(char* tx_string)
{
    while (*tx_string)
        tx_spi0(*(tx_string++));
}

void tx_data_spi0(uint8_t tx_byte[], int length)
{
    while (length--)
        tx_spi0(*(tx_byte++));
}
```

timer_loop.c

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/cpufunc.h>
#include "timer_callback.h"

unsigned char period = 1;
unsigned char tick = 0;

ISR(TCB0_INT_vect)
{
    TCB0.INTFLAGS = 0x01;

    if (++tick > period) {
        tick = 0;
        timer_callback();
    }
}
```

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X][(1)]	-	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	PIC16F18313, PIC16LF18313, PIC16F18323, PIC16LF18323	
Tape and Reel Option:	Blank	= Standard packaging (tube or tray)
	T	= Tape and Reel[(1)]
Temperature Range:	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package:[(2)]	JQ	= UQFN
	P	= PDIP
	ST	= TSSOP
	SL	= SOIC-14
	SN	= SOIC-8
	RF	= UDFN
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

- PIC16LF18313- I/P Industrial temperature, PDIP package
- PIC16F18313- E/SS Extended temperature, SSOP package

Notes:

1. Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.
2. Small form-factor packaging options may be available. Please check www.microchip.com/packaging for small-form factor package availability, or contact your local Sales Office.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-6204-0

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820