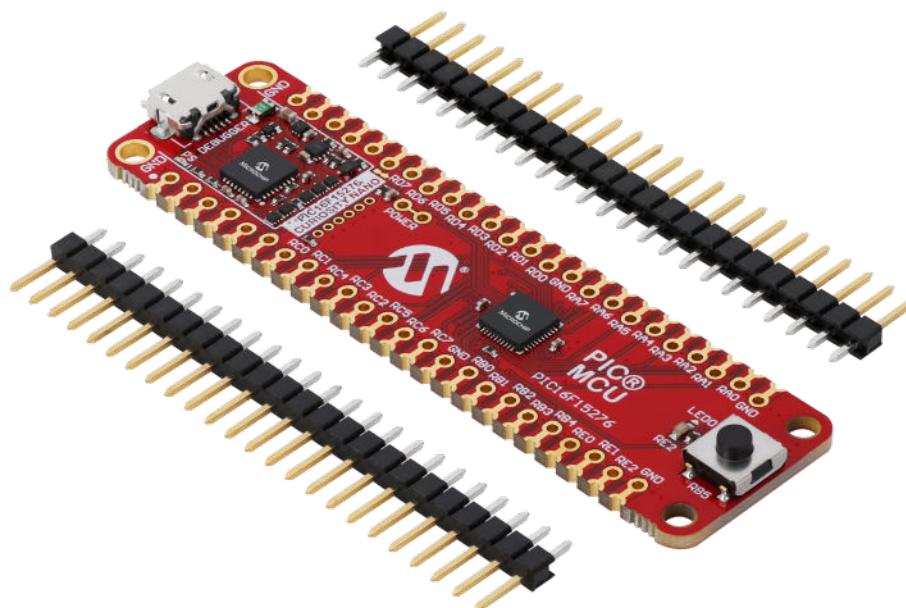

PIC16F15276 Curiosity Nano Hardware User Guide

Preface

The PIC16F15276 Curiosity Nano evaluation kit is a hardware platform to evaluate microcontrollers in the PIC16F152XX family. This board has the PIC16F15276 microcontroller (MCU) mounted.

Supported by MPLAB® X IDE, the board provides easy access to the features of the PIC16F15276 to explore how to integrate the device into a custom design.

The Curiosity Nano series of evaluation boards include an on-board debugger. No external tools are necessary to program and debug the PIC16F15276.



- [MPLAB® X IDE](#) - Software to discover, configure, develop, program, and debug Microchip microcontrollers.
- [Code examples on GitHub](#) - Get started with code examples.
- [PIC16F15276 website](#) - Find documentation, data sheets, sample, and purchase microcontrollers.
- [PIC16F15276 Curiosity Nano website](#) - Kit information, latest user guide, and design documentation.

Table of Contents

Preface.....	1
1. Introduction.....	3
1.1. Features.....	3
1.2. Board Overview.....	3
2. Getting Started.....	4
2.1. Quick Start.....	4
2.2. Design Documentation and Relevant Links.....	4
3. Curiosity Nano.....	6
3.1. On-Board Debugger Overview.....	6
3.2. Curiosity Nano Standard Pinout.....	12
3.3. Power Supply.....	13
3.4. Low-Power Measurement.....	17
3.5. Programming External Microcontrollers.....	18
3.6. Connecting External Debuggers.....	21
4. Hardware User Guide.....	23
4.1. Connectors.....	23
4.2. Peripherals.....	24
5. Hardware Revision History and Known Issues.....	26
5.1. Identifying Product ID and Revision.....	26
5.2. Revision 1.....	26
6. Document Revision History.....	27
7. Appendix.....	28
7.1. Schematic.....	28
7.2. Assembly Drawing.....	30
7.3. Curiosity Nano Base for Click boards™.....	31
7.4. Disconnecting the On-Board Debugger.....	32
The Microchip Website.....	34
Product Change Notification Service.....	34
Customer Support.....	34
Microchip Devices Code Protection Feature.....	34
Legal Notice.....	35
Trademarks.....	35
Quality Management System.....	36
Worldwide Sales and Service.....	37

1. Introduction

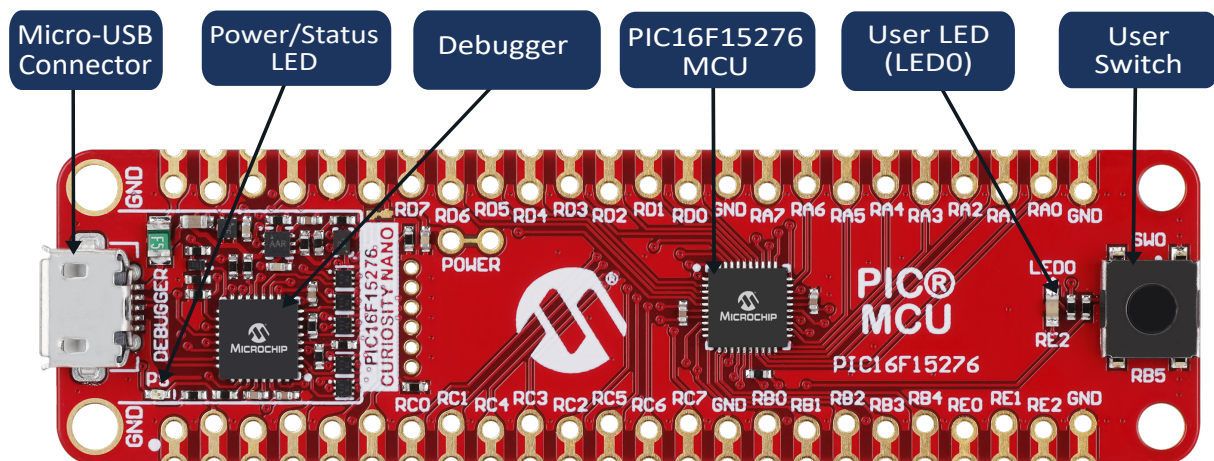
1.1 Features

- PIC16F15276 Microcontroller
- One Yellow User LED
- One Mechanical User Switch
- On-Board Debugger:
 - Board identification in Microchip MPLAB® X IDE
 - One green power and status LED
 - Programming and debugging
 - Virtual serial port (CDC)
 - One debug GPIO channel (DGI GPIO)
- USB Powered
- Adjustable Target Voltage:
 - MIC5353 LDO regulator controlled by the on-board debugger
 - 1.8–5.1V output voltage (limited by USB input voltage)
 - 500 mA maximum output current (limited by ambient temperature and output voltage)

1.2 Board Overview

The Microchip PIC16F15276 Curiosity Nano evaluation kit is a hardware platform to evaluate the PIC16F15276 microcontroller.

Figure 1-1. PIC16F15276 Curiosity Nano Board Overview



2. Getting Started

2.1 Quick Start

Steps to start exploring the PIC16F15276 Curiosity Nano board:

1. Download Microchip [MPLAB® X IDE](#).
2. Download [MPLAB® XC C Compiler](#).
3. Launch MPLAB® X IDE.
4. Optional: Use [MPLAB® Code Configurator](#) to generate drivers and examples.
5. Write the application code.
6. Connect a USB cable (Standard-A to Micro-B or Micro-AB) between the PC and the debug USB port on the board.

2.1.1 Driver Installation

When the board connects to the computer for the first time, the operating system will perform a driver software installation. The driver file supports both 32- and 64-bit versions of Microsoft® Windows® XP, Windows Vista®, Windows® 7, Windows® 8, and Windows® 10. The drivers for the board are included with MPLAB® X IDE.

2.1.2 Kit Window

When the board is connected to a computer and powered on, the green status LED will be lit, the MPLAB® X IDE will then auto-detect which boards are connected. The Kit Window in MPLAB® X IDE will present relevant information like data sheets and board documentation. The PIC16F15276 device on the PIC16F15276 Curiosity Nano board is programmed and debugged by the on-board debugger and, therefore, no external programmer or debugger tool is required.



Tip: If closed, the Kit Window in MPLAB® X IDE can be reopened through the menu bar **Window > Kit Window**.

2.1.3 MPLAB® X IDE Device Family Packs

Microchip MPLAB® X IDE requires specific information to support devices and tools. This information is contained in versioned packs. For the PIC16F15276 Curiosity Nano board, MPLAB® X version 5.40 with device family pack “PIC16F1xxx_DFP” version 1.7.146 and tool pack “nEDBG_TP” version 1.8.318 or newer are required. For more information on packs and how to upgrade them, refer to the [MPLAB® X IDE User's guide - Work with Device Packs](#).



Tip: The latest device family packs are available through **Tools > Packs** in MPLAB® X IDE or online at [Microchip MPLAB® X Packs Repository](#).

2.2 Design Documentation and Relevant Links

The following list contains links to the most relevant documents and software for the PIC16F15276 Curiosity Nano board:

- [MPLAB® X IDE](#) - MPLAB X IDE is a software program that runs on a PC (Windows®, Mac OS®, Linux®) to develop applications for Microchip microcontrollers and digital signal controllers. It is named an Integrated

Development Environment (IDE) because it provides a single integrated “environment” to develop code for embedded microcontrollers.

- **[MPLAB® XC Compilers](#)** - MPLAB® XC8 C Compiler is available as a free, unrestricted-use download. Microchips MPLAB® XC8 C Compiler is a comprehensive solution for your project's software development on Windows®, macOS® or Linux®. MPLAB® XC8 supports all 8-bit PIC® and AVR® microcontrollers (MCUs).
- **[MPLAB® Code Configurator](#)** - MPLAB Code Configurator (MCC) is a free software plug-in that provides a graphical interface to configure peripherals and functions specific to your application.
- **[Microchip Sample Store](#)** - Microchip sample store where you can order samples of devices.
- **[MPLAB Data Visualizer](#)** - MPLAB Data Visualizer is a program used for processing and visualizing data. The Data Visualizer can receive data from various sources such as serial ports and on-board debugger's Data Gateway Interface, as found on Curiosity Nano and Xplained Pro boards.
- **[Microchip PIC® and AVR® Examples](#)** - Microchip PIC and AVR Device Examples is a collection of examples and labs that use Microchip development boards to showcase the use of PIC and AVR device peripherals.
- **[Microchip PIC® and AVR® Solutions](#)** - Microchip PIC and AVR Device Solutions contains complete applications for use with Microchip development boards, ready to be adapted and extended.
- **[PIC16F15276 Curiosity Nano website](#)** - Kit information, latest user guide, and design documentation.
- **[PIC16F15276 Curiosity Nano on Microchip Direct](#)** - Purchase this kit on Microchip Direct.

3. Curiosity Nano

Curiosity Nano is an evaluation platform of small boards with low pin count microcontroller (MCU) boards with on-board debuggers and access to most of the microcontrollers I/Os. The Curiosity Nano platform offers easy integration with MPLAB® X IDE. All boards are identified in the IDE, and when connected a Kit Window is displayed with links to key documentation, including relevant user guides, application notes, data sheets, and example code. Everything is easy to find. The on-board debugger features a virtual serial port (CDC) for serial communication to a host PC and a Data Gateway Interface (DGI) with debug GPIO pin(s).

3.1 On-Board Debugger Overview

PIC16F15276 Curiosity Nano contains an on-board debugger for programming and debugging. The on-board debugger is a composite USB device consisting of several interfaces:

- A debugger that can program and debug the PIC16F15276 in MPLAB® X IDE
- A mass storage device that allows drag-and-drop programming of the PIC16F15276
- A virtual serial port (CDC) that is connected to a Universal Asynchronous Receiver/Transmitter (UART) on the PIC16F15276, and provides an easy way to communicate with the target application through terminal software
- A Data Gateway Interface (DGI) for code instrumentation with logic analyzer channels (debug GPIO) to visualize program flow

The on-board debugger controls a Power and Status LED (marked PS) on the PIC16F15276 Curiosity Nano board. The table below shows how the LED is controlled in different operation modes.

Table 3-1. On-Board Debugger LED Control

Operation Mode	Power and Status LED
Boot Loader mode	The LED blinks slowly during power-up
Power-up	The LED is ON
Normal operation	The LED is ON
Programming	Activity indicator: The LED blinks slowly during programming/debugging
Drag-and-drop programming	Success: The LED blinks slowly for 2 sec. Failure: The LED blinks rapidly for 2 sec.
Fault	The LED blinks rapidly if a power fault is detected
Sleep/Off	The LED is OFF. The on-board debugger is either in a sleep mode or powered down. This can occur if the board is externally powered.



Info: Slow blinking is approximately 1 Hz, and rapid blinking is approximately 5 Hz.

3.1.1 Debugger

The on-board debugger on the PIC16F15276 Curiosity Nano board appears as a Human Interface Device (HID) on the host computer's USB subsystem. The debugger supports full-featured programming and debugging of the PIC16F15276 using MPLAB® X IDE.



Remember: Keep the debugger's firmware up-to-date. Firmware upgrades automatically when using MPLAB® X IDE.

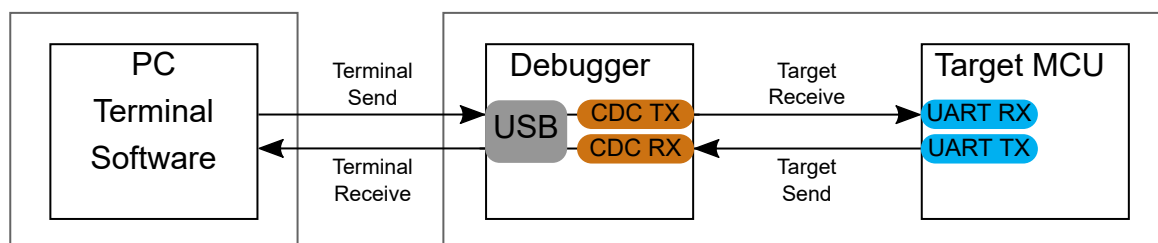
3.1.2 Virtual Serial Port (CDC)

The virtual serial port (CDC) is a general purpose serial bridge between a host PC and a target device.

3.1.2.1 Overview

The on-board debugger implements a composite USB device that includes a standard Communications Device Class (CDC) interface, which appears on the host as a virtual serial port. The CDC can be used to stream arbitrary data in both directions between the host computer and the target: All characters sent through the virtual serial port on the host computer will be transmitted as UART on the debugger's CDC TX pin, and UART characters captured on the debugger's CDC RX pin will be returned to the host computer through the virtual serial port.

Figure 3-1. CDC Connection



Info: As shown in [Figure 3-1](#), the debugger's CDC TX pin is connected to a UART RX pin on the target for receiving characters from the host computer. Similarly, the debugger's CDC RX pin is connected to a UART TX pin on the target for transmitting characters to the host computer.

3.1.2.2 Operating System Support

On Windows® machines, the CDC will enumerate as *Curiosity Virtual COM Port* and appear in the Ports section of the Windows Device Manager. The COM port number can also be found there.



Info: On older Windows systems, the CDC requires a USB driver. This driver is included in installations of MPLAB® X IDE.

On Linux® machines, the CDC will enumerate and appear as `/dev/ttyACM#`.



Info: `tty*` devices belong to the “dialout” group in Linux, so it may be necessary to become a member of that group to have permissions to access the CDC.

On Mac® machines, the CDC will enumerate and appear as `/dev/tty.usbmodem#`. Depending on which terminal program is used, it will appear in the available list of modems as `usbmodem#`.



Info: For all operating systems: Be sure to use a terminal emulator that supports DTR signaling. See [3.1.2.4 Signaling](#).

3.1.2.3 Limitations

Not all UART features are implemented in the on-board debugger CDC. The constraints are outlined here:

- **Baud rate:** Must be in the range of 1200 bps to 500 kbps. Any baud rate outside this range will be set to the closest limit, without warning. Baud rate can be changed on-the-fly.
- **Character format:** Only 8-bit characters are supported.
- **Parity:** Can be odd, even, or none.
- **Hardware flow control:** Not supported.
- **Stop bits:** One or two bits are supported.

3.1.2.4 Signaling

During USB enumeration, the host OS will start both communication and data pipes of the CDC interface. At this point, it is possible to set and read back the baud rate and other UART parameters of the CDC, but data sending and receiving will not be enabled.

When a terminal connects to the host, it must assert the DTR signal. As this is a virtual control signal implemented on the USB interface, it is not physically present on the board. Asserting the DTR signal from the host will indicate to the on-board debugger that a CDC session is active. The debugger will then enable its level shifters (if available) and start the CDC data send and receive mechanisms.

Deasserting DTR in debugger firmware version 1.20 or earlier has the following behavior:

- Debugger UART receiver is disabled, and no further data will be transferred to the host computer
- Debugger UART transmitter will continue to send data that is queued for sending, but no new data is accepted from the host computer
- Level shifters (if available) are not disabled, and the debugger CDC TX line remains driven

Deasserting DTR in debugger firmware version 1.21 or later has the following behavior:

- Debugger UART receiver is disabled, so no further data will be transferred to the host computer
- Debugger UART transmitter will continue to send data that is queued for sending, but no new data is accepted from the host computer
- Once the ongoing transmission is complete, level shifters (if available) are disabled, so the debugger CDC TX line will become high-impedance



Remember: Set up the terminal emulator to assert the DTR signal. Without the signal, the on-board debugger will not send or receive any data through its UART.



Tip: The on-board debugger's CDC TX pin will not be driven until the CDC interface is enabled by the host computer. Also, there are no external pull-up resistors on the CDC lines connecting the debugger and the target, which means that during power-up, these lines are floating. To avoid any glitches resulting in unpredictable behavior like framing errors, the target device may enable the internal pull-up resistor on the pin connected to the debugger's CDC TX pin.

3.1.2.5 Advanced Use

CDC Override Mode

In normal operation, the on-board debugger is a true UART bridge between the host and the device. However, in certain use cases, the on-board debugger can override the basic operating mode and use the CDC TX and RX pins for other purposes.

Dropping a text file into the on-board debugger's mass storage drive can be used to send characters out of the debugger's CDC TX pin. The filename and extension are trivial, but the text file must start with the characters:

```
CMD:SEND_UART=
```

Debugger firmware version 1.20 or earlier has the following limitations:

- The maximum message length is 50 characters – all remaining data in the frame are ignored
- The default baud rate used in this mode is 9600 bps, but if the CDC is already active or has been configured, the previously used baud rate still applies

Debugger firmware version 1.21 and later has the following limitations/features:

- The maximum message length may vary depending on the MSC/SCSI layer timeouts on the host computer and/or operating system. A single SCSI frame of 512 bytes (498 characters of payload) is ensured, and files of up to 4 KB will work on most systems. The transfer will complete on the first NULL character encountered in the file.
- The baud rate used is always 9600 bps for the default command:

```
CMD:SEND_UART=
```

- Do not use the CDC override mode simultaneously with data transfer over the CDC/terminal. If a CDC terminal session is active at the time a file is received via CDC Override mode, it will be suspended for the duration of the operation and resumed once complete.
- Additional commands are supported with explicit baud rates:

```
CMD:SEND_9600=
```

```
CMD:SEND_115200=
```

```
CMD:SEND_460800=
```

USB-Level Framing Considerations

Sending data from the host to the CDC can be done byte-wise or in blocks, which will be chunked into 64-byte USB frames. Each such frame will be queued up for sending to the debugger's CDC TX pin. Transferring a small amount of data per frame can be inefficient, particularly at low baud rates, as the on-board debugger buffers frames and not bytes. A maximum of four 64-byte frames can be active at any time. The on-board debugger will throttle the incoming frames accordingly. Sending full 64-byte frames containing data is the most efficient method.

When receiving data on the debugger's CDC RX pin, the on-board debugger will queue up the incoming bytes into 64-byte frames, which are sent to the USB queue for transmission to the host when they are full. Incomplete frames are also pushed to the USB queue at approximately 100 ms intervals, triggered by USB start-of-frame tokens. Up to eight 64-byte frames can be active at any time.

If the host (or the software running on it) fails to receive data fast enough, an overrun will occur. When this happens, the last-filled buffer frame will be recycled instead of being sent to the USB queue, and a full data frame will be lost. To prevent this occurrence, the user must ensure that the CDC data pipe is being read continuously, or the incoming data rate must be reduced.

3.1.3 Mass Storage Device

The on-board debugger includes a simple Mass Storage Device implementation, which is accessible for read/write operations via the host operating system to which it is connected.

It provides:

- Read access to basic text and HTML files for detailed kit information and support
- Write access for programming Intel® HEX formatted files into the target device's memory
- Write access for simple text files for utility purposes

3.1.3.1 Mass Storage Device Implementation

The on-board debugger implements a highly optimized variant of the FAT12 file system that has several limitations, partly due to the nature of FAT12 itself and optimizations made to fulfill its purpose for its embedded application.

The Curiosity Nano USB device is USB Chapter 9-compliant as a mass storage device but does not, in any way, fulfill the expectations of a general purpose mass storage device. This behavior is intentional.

When using the Windows operating system, the on-board debugger enumerates as a Curiosity Nano USB Device that can be found in the disk drives section of the device manager. The CURIOSITY drive appears in the file manager and claims the next available drive letter in the system.

The CURIOSITY drive contains approximately one MB of free space, and this does not reflect the size of the target device's Flash in any way. When programming an Intel® HEX file, the binary data are encoded in ASCII with metadata providing a large overhead, so 1 MB is a trivially chosen value for disk size.

It is not possible to format the CURIOSITY drive. When programming a file to the target, the filename may appear in the disk directory listing. This is merely the operating system's view of the directory, which has not been updated in reality. It is not possible to read out the file contents. Removing and replugging the board will return the file system to its original state, but the target will still contain the application that has previously been programmed.

To erase the target device, copy a text file starting with "CMD:ERASE" onto the disk.

By default, the CURIOSITY drive contains several read-only files for generating icons as well as reporting status and linking to further information:

- AUTORUN.ICO – icon file for the Microchip logo
- AUTORUN.INF – system file required for Windows Explorer to show the icon file
- KIT-INFO.HTM – redirect to the development board website
- KIT-INFO.TXT – a text file containing details about the board's debugger firmware version, board name, USB serial number, device, and drag-and-drop support
- STATUS.TXT – a text file containing the programming status of the board



Info: STATUS.TXT is dynamically updated by the on-board debugger. The contents may be cached by the OS and, therefore, do not reflect the correct status.

3.1.3.2 Configuration Words

Configuration Words (PIC® MCU Targets)

Configuration Word settings included in the project being programmed after program Flash is programmed. The debugger will not mask out any bits in the Configuration Words when writing them, but since it uses Low-Voltage Programming mode, it is unable to clear the LVP Configuration bit. If the incorrect clock source is selected, for example, and the board does not boot, it is always possible to perform a bulk erase (always done before programming) and restore the device to its default settings.

3.1.3.3 Special Commands

Several utility commands are supported by copying text files to the mass storage disk. The filename or extension is irrelevant – the command handler reacts to content only.

Table 3-2. Special File Commands

Command Content	Description
CMD:ERASE	Executes a chip erase of the target

.....continued

Command Content	Description
CMD:SEND_UART=	Sends a string of characters to the CDC UART. See “ CDC Override Mode .”
CMD:SEND_9600= CMD:SEND_115200= CMD:SEND_460800=	Sends a string of characters to the CDC UART at the baud rate specified. Note that only the baud rates explicitly specified here are supported! See “ CDC Override Mode ”. (Debugger firmware v1.21 or newer.)
CMD:RESET	Resets the target device by entering Programming mode and then exiting Programming mode immediately afterward. Exact timing can vary according to the programming interface of the target device. (Debugger firmware v1.16 or newer.)
CMD:POWERTOGGLE	Powers down the target and restores power after a 100 ms delay. If external power is provided, this has no effect. (Debugger firmware v1.16 or newer.)
CMD:0V	Powers down the target device by disabling the target supply regulator. If external power is provided, this has no effect. (Debugger firmware v1.16 or newer.)
CMD:1V8	Sets the target voltage to 1.8V. If external power is provided, this has no effect. (Debugger firmware v1.21 or newer.)
CMD:3V3	Sets the target voltage to 3.3V. If external power is provided, this has no effect. (Debugger firmware v1.16 or newer.)
CMD:5V0	Sets the target voltage to 5.0V. If external power is provided, this has no effect. (Debugger firmware v1.16 or newer.)



Info: The content sent to the mass storage emulated disk triggers the commands listed here and provides no feedback in the case of either success or failure.

3.1.4 Data Gateway Interface (DGI)

Data Gateway Interface (DGI) is a USB interface for transporting raw and timestamped data between on-board debuggers and host computer-based visualization tools. [MPLAB Data Visualizer](#) is used on the host computer to display debug GPIO data. It is available as a plug-in for MPLAB® X IDE or a stand-alone application that can be used in parallel with MPLAB® X IDE.

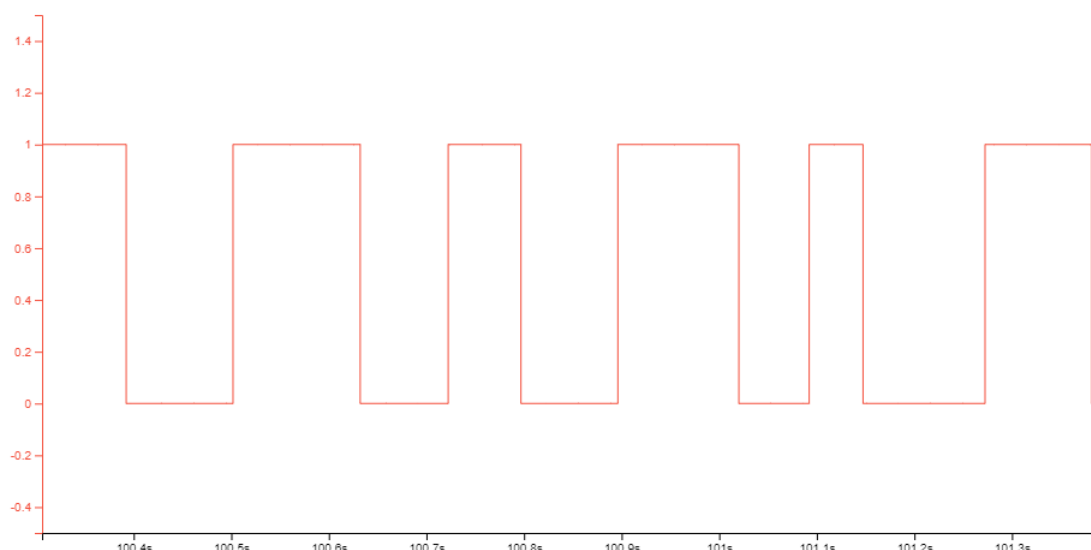
Although DGI encompasses several physical data interfaces, the PIC16F15276 Curiosity Nano implementation includes logic analyzer channels:

- One debug GPIO channel (also known as DGI GPIO)

3.1.4.1 Debug GPIO

Debug GPIO channels are timestamped digital signal lines connecting the target application to a host computer visualization application. They are typically used to plot the occurrence of low-frequency events on a time-axis – for example, when certain application state transitions occur.

The figure below shows the monitoring of the digital state of a mechanical switch connected to a debug GPIO in MPLAB Data Visualizer.

Figure 3-2. Monitoring Debug GPIO with MPLAB® Data Visualizer

Debug GPIO channels are timestamped, so the resolution of DGI GPIO events is determined by the resolution of the DGI timestamp module.



Important: Although bursts of higher frequency signals can be captured, the useful frequency range of signals for which debug GPIO can be used is up to about 2 kHz. Attempting to capture signals above this frequency will result in data saturation and overflow, which may cause the DGI session to be aborted.

3.1.4.2 Timestamping

DGI sources are timestamped as they are captured by the debugger. The timestamp counter implemented in the Curiosity Nano debugger increments at 2 MHz frequency, providing a timestamp resolution of a half microsecond.

3.2 Curiosity Nano Standard Pinout

The 12 edge connections closest to the USB connector on Curiosity Nano boards have a standardized pinout. The program/debug pins have different functions depending on the target programming interface, as shown in the table and figure below.

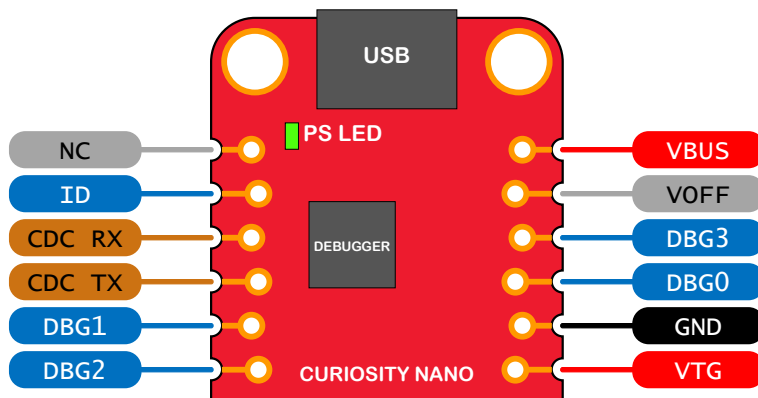
Table 3-3. Curiosity Nano Standard Pinout

Debugger Signal	Target MCU	Description
ID	—	ID line for extensions
CDC TX	UART RX	USB CDC TX line
CDC RX	UART TX	USB CDC RX line
DBG0	ICSPDAT	Debug data line
DBG1	ICSPCLK	Debug clock line
DBG2	GPIO0	debug GPIO0
DBG3	MCLR	Reset line
NC	—	No connect

.....continued

Debugger Signal	Target MCU	Description
V _{BUS}	—	V _{BUS} voltage for external use
VOFF	—	Voltage Off input. Disables the target regulator and target voltage when pulled low.
VTG	—	Target voltage
GND	—	Common ground

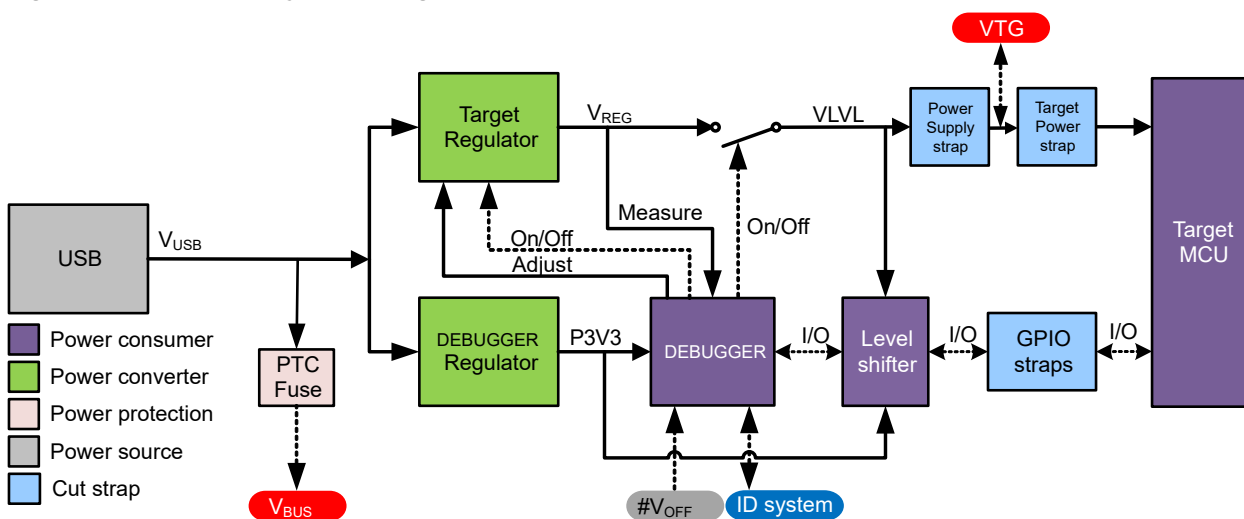
Figure 3-3. Curiosity Nano Standard Pinout



3.3 Power Supply

The board is powered through the USB port and contains two LDO regulators, one to generate 3.3V for the on-board debugger and an adjustable LDO regulator for the target PIC16F15276 microcontroller and its peripherals. The voltage from a USB connector can vary between 4.4V to 5.25V (according to the USB specification) and will limit the maximum voltage to the target. The figure below shows the entire power supply system on PIC16F15276 Curiosity Nano.

Figure 3-4. Power Supply Block Diagram



3.3.1 Target Regulator

The target voltage regulator is a MIC5353 variable output LDO. The on-board debugger can adjust the voltage output supplied to the board target section by manipulating the MIC5353's feedback voltage. The hardware implementation

is limited to an approximate voltage range from 1.7V to 5.1V. Additional output voltage limits are configured in the debugger firmware to ensure that the output voltage never exceeds the hardware limits of the PIC16F15276 microcontroller. The voltage limits configured in the on-board debugger on PIC16F15276 Curiosity Nano are 1.8–5.5V.



Info: The target voltage is set to 3.3V when the board is manufactured. It can be changed through the MPLAB® X IDE project properties. Any change to the target voltage is persistent, even after a power toggle. The resolution is less than 5 mV but may be limited to 10 mV by the adjustment program.



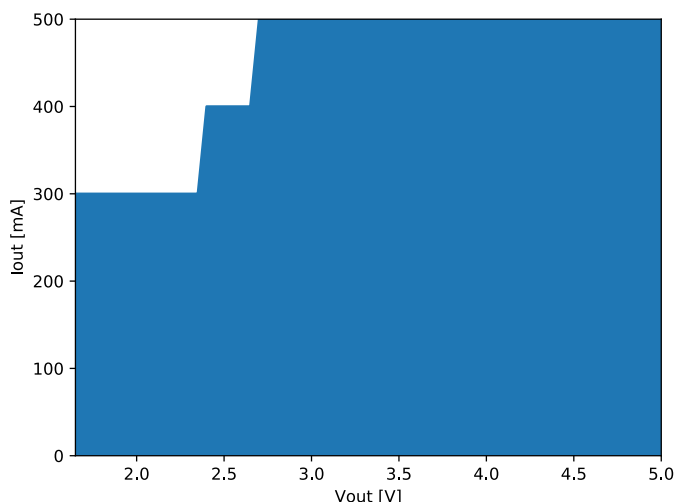
Info: The voltage settings setup in MPLAB® X IDE are not immediately applied to the board. The new voltage setting is applied to the board when the debugger is accessed in any way, like pushing the Refresh Debug Tool Status button in the project dashboard tab or programming/reading program memory.



Info: There is a simple option to adjust the target voltage with a drag-and-drop command text file to the board, which supports a set of common target voltages. See section [3.1.3.3 Special Commands](#) for further details.

The MIC5353 supports a maximum current load of 500 mA. It is an LDO regulator in a small package, placed on a small printed circuit board (PCB), and the thermal shutdown condition can be reached at lower loads than 500 mA. The maximum current load depends on the input voltage, the selected output voltage, and the ambient temperature. The figure below shows the safe operating area for the regulator, with an input voltage of 5.1V and an ambient temperature of 23°C.

Figure 3-5. Target Regulator Safe Operation Area



The voltage output of the target regulator is continuously monitored (measured) by the on-board debugger. If it is more than 100 mV over/under the set device voltage, an error condition will be flagged, and the target voltage regulator will be turned off. This will detect and handle any short-circuit conditions. It will also detect and handle if an external voltage, which causes VCC_TARGET to move outside of the voltage setting monitoring window of ± 100 mV, is suddenly applied to the VTG pin without setting the VOFF pin low.



Info: The on-board debugger has a monitoring window of $VCC_TARGET \pm 100$ mV. If the external voltage is under this limit, the on-board debugger status LED will blink rapidly. If the external voltage is above this limit, the on-board debugger status LED will continue to shine. If the external voltage is removed, the status LED will start to blink rapidly until the on-board debugger detects the new situation and turns the target voltage regulator back on.

3.3.2 External Supply

An external voltage instead of the on-board target regulator can power the PIC16F15276 Curiosity Nano. When the Voltage Off (VOFF) pin is shorted to ground (GND), the on-board debugger firmware disables the target regulator, and it is safe to apply an external voltage to the VTG pin.

It is also safe to apply an external voltage to the VTG pin when no USB cable is plugged into the DEBUG connector on the board.

The VOFF pin can be tied low/let go at any time. This will be detected by a pin-change interrupt to the on-board debugger, which controls the target voltage regulator accordingly.



Applying an external voltage to the VTG pin without shorting VOFF to GND may cause permanent damage to the board.



Do not apply any voltage to the VOFF pin. Let the pin float to enable the power supply.



The absolute maximum external voltage is 5.5V for the on-board level shifters, and the standard operating condition of the PIC16F15276 is 1.8–5.5V. Applying a higher voltage may cause permanent damage to the board.



Info: If an external voltage is applied without pulling the VOFF pin low and an external supply pulls the voltage lower than the monitoring window's lower limit (target voltage setting – 100 mV), the on-board debugger status LED will blink rapidly and shut the on-board regulator off. If an external voltage is suddenly removed when the VOFF pin is not pulled low, the status LED will start to blink rapidly until the on-board debugger detects the new situation and switches the target voltage regulator back on.

Programming, debugging, and data streaming are still possible with an external power supply – the debugger and signal level shifters will be powered from the USB cable. Both regulators, the debugger, and the level shifters are powered down when the USB cable is removed.

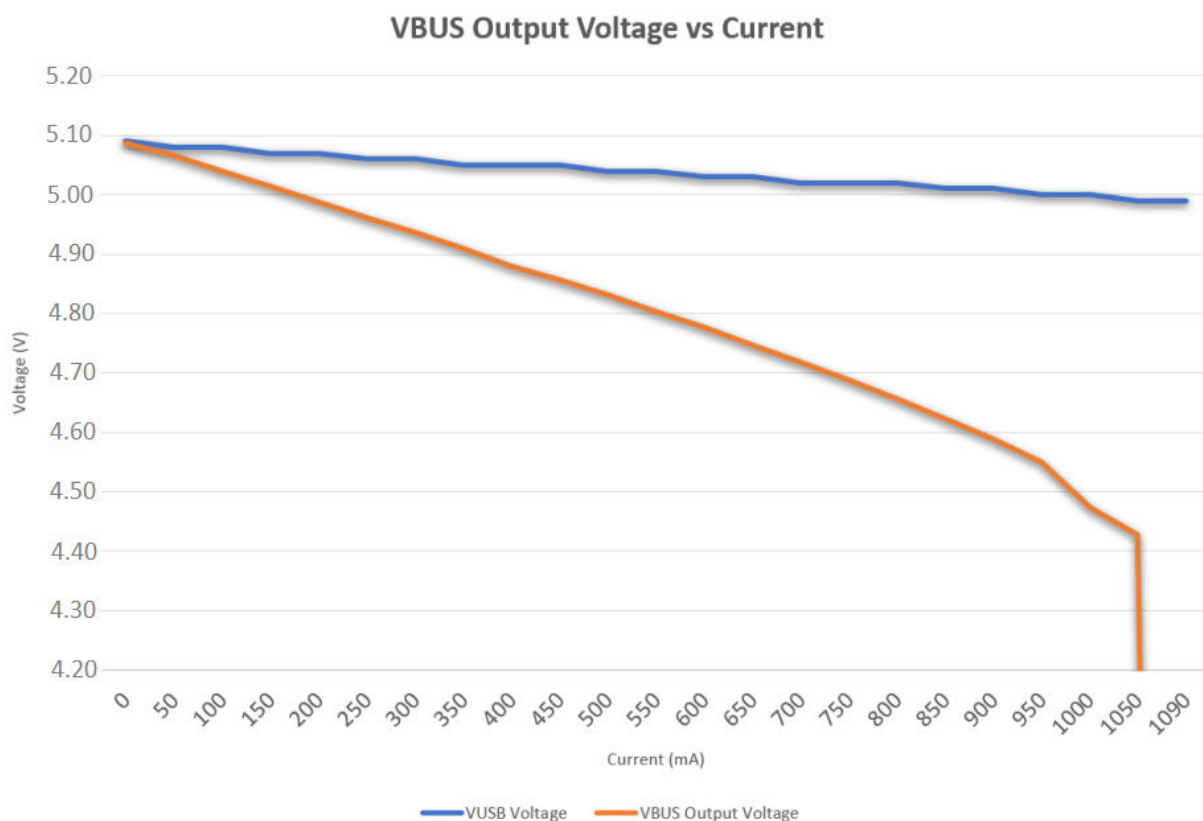


Info: In addition to the power consumed by the PIC16F15276 and its peripherals, approximately 100 μ A will be drawn from any external power source to power the on-board level shifters and voltage monitor circuitry when a USB cable is plugged in the DEBUG connector on the board. When a USB cable is not plugged in, some current is used to supply the level shifters voltage pins, which have a worst-case current consumption of approximately 5 μ A. Typical values may be as low as 100 nA.

3.3.3 VBUS Output Pin

PIC16F15276 Curiosity Nano has a VBUS output pin that can be used to power external components that need a 5V supply. The VBUS output pin has a PTC fuse to protect the USB against short circuits. A side effect of the PTC fuse is a voltage drop on the VBUS output with higher current loads. The chart below shows the voltage versus the current load of the VBUS output.

Figure 3-6. VBUS Output Voltage vs. Current



3.3.4 Power Supply Exceptions

This is a summary of most exceptions that can occur with the power supply.

Target Voltage Shuts Down

This can happen if the target section draws too much current at a given voltage and cause the thermal shutdown safety feature of the MIC5353 regulator to kick in. To avoid this, reduce the current load of the target section.

Target Voltage Setting is Not Reached

The USB input voltage (specified to be 4.4V-5.25V) limits the maximum output voltage, the voltage drop over the MIC5353 regulator at a given voltage setting and current consumption. If a higher output voltage is needed, use a USB power source that can provide a higher input voltage or use an external voltage supply on the VTG pin.

Target Voltage is Different From Setting

This can be caused by an externally applied voltage to the VTG pin without setting the VOFF pin low. If the target voltage differs more than 100 mV over/under the voltage setting, it will be detected by the on-board debugger, and the internal voltage regulator will shut down. To fix this issue, remove the applied voltage from the VTG pin, and the on-board debugger will enable the on-board voltage regulator when the new condition is detected. Note that the PS LED will be blinking rapidly if the target voltage is below 100 mV of the setting but will normally be lit when it is higher than 100 mV above the setting.

No, Or Very Low Target Voltage, and PS LED is Blinking Rapidly

This can be caused by a full or partial short circuit and is a special case of the issue mentioned above. Remove the short circuit, and the on-board debugger will re-enable the on-board target voltage regulator.

No Target Voltage and PS LED is Lit 1

This occurs if the target voltage is set to 0.0V. To fix this, set the target voltage to a value within the specified voltage range for the target device.

No Target Voltage and PS LED is Lit 2

This can be the issue if power jumper J100 and/or J101 is cut, and the target voltage regulator is set to a value within the specified voltage range for the target device. To fix this, solder a wire/bridge between the pads for J100/J101, or add a jumper on J101 if a pin-header is mounted.

V_{BUS} Output Voltage is Low or Not Present

This is most likely caused by a high-current drain on V_{BUS}, and the protection fuse (PTC) will reduce the current or cut off completely. Reduce the current consumption on the VBUS pin to fix this issue.

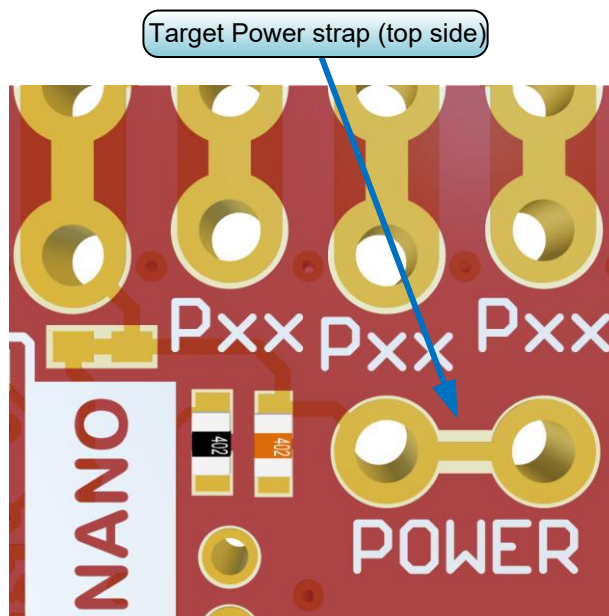
3.4 Low-Power Measurement

Power to the PIC16F15276 is connected from the on-board power supply and VTG pin through a 100 mil pin-header marked with "POWER" in silkscreen (J101). To measure the power consumption of the PIC16F15276 and other peripherals connected to the board, cut the *Target Power strap* and connect an ammeter over the strap.

To measure the lowest possible power consumption, follow these steps:

1. Cut the POWER strap with a sharp tool.
2. Solder a 1x2 100 mil pin-header in the footprint.
3. Connect an ammeter to the pin-header.
4. Write firmware that:
 - 4.1. Tri-states any I/O connected to the on-board debugger.
 - 4.2. Sets the microcontroller in its lowest Power Sleep Mode.
5. Program the firmware into the PIC16F15276.

Figure 3-7. Target Power Strap





Tip: A 100-mil pin-header can be soldered into the *Target Power strap* (J101) footprint for easy connection of an ammeter. Once the ammeter is no longer needed, place a jumper cap on the pin-header.



Info: The on-board level shifters will draw a small amount of current even when they are not in use. A maximum of 2 μ A can be drawn from each I/O pin connected to a level shifter for a total of 10 μ A. Keep any I/O pin connected to a level shifter in tri-state to prevent leakage. All I/Os connected to the on-board debugger are listed in [4.2.3.1 On-Board Debugger Connections](#). To prevent any leakage to the on-board level shifters, they can be disconnected completely, as described in [7.4 Disconnecting the On-Board Debugger](#).

3.5 Programming External Microcontrollers

The on-board debugger on PIC16F15276 Curiosity Nano can be used to program and debug microcontrollers on external hardware.

3.5.1 Supported Devices

All external AVR microcontrollers with the UPDI interface can be programmed and debugged with the on-board debugger with Microchip Studio.

External SAM microcontrollers that have a Curiosity Nano Board can be programmed and debugged with the on-board debugger with Microchip Studio.

PIC16F15276 Curiosity Nano can program and debug external PIC16F15276 microcontrollers with MPLAB X IDE.

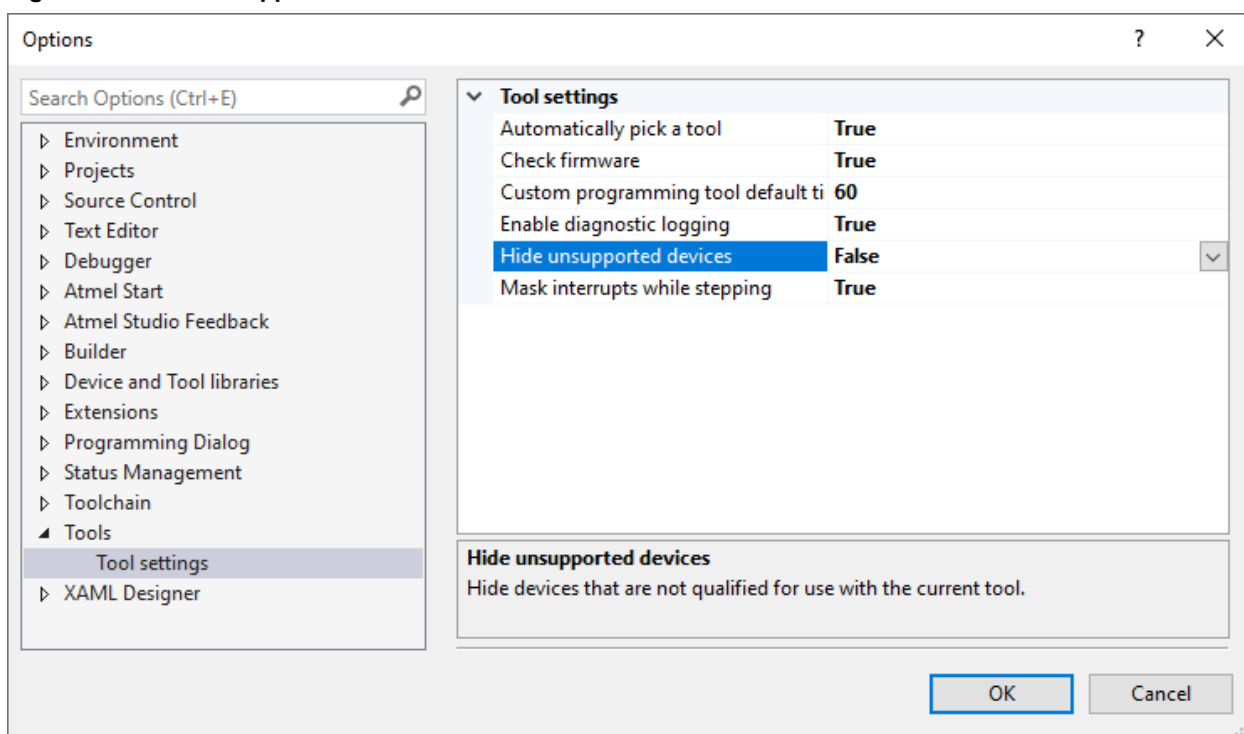
3.5.2 Software Configuration

No software configuration is required to program and debug the same device that is mounted on the board.

To program and debug a different microcontroller than what is mounted on the board, Microchip Studio must be configured to allow free selection of devices and programming interfaces.

1. Navigate to **Tools > Options** through the menu system at the top of the application.
2. Select the **Tools > Tool settings** category in the options window.
3. Set the **Hide unsupported devices** option to **False**.

Figure 3-8. Hide Unsupported Devices

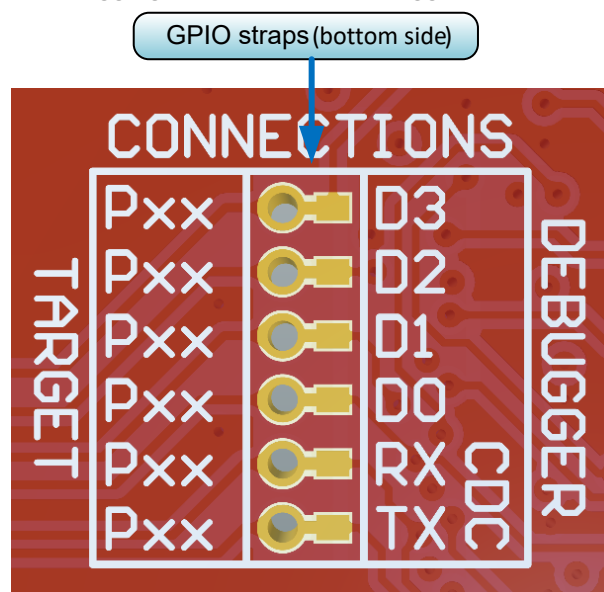


Info: Microchip Studio allows any microcontroller and interface to be selected when the **Hide unsupported devices** setting is set to **False**, also microcontrollers and interfaces which are not supported by the on-board debugger.

3.5.3 Hardware Modifications

The on-board debugger is connected to the PIC16F15276 by default. Remove these connections before any external microcontroller can be programmed or debugged. Cut the *GPIO straps* shown in the figure below with a sharp tool to disconnect the PIC16F15276 from the on-board debugger.

Figure 3-9. Programming and Debugging Connections to Debugger



Info: Cutting the connections to the debugger will disable programming, debugging, and data streaming from the PIC16F15276 mounted on the board.



Tip: Solder 0Ω resistors across the footprints or short circuit them with solder to reconnect the signals between the on-board debugger and the PIC16F15276.

3.5.4 Connecting to External Microcontrollers

The figure and table below show where the programming and debugging signals must be connected to program and debug external microcontrollers. The on-board debugger can supply power to the external hardware or use an external voltage as a reference for its level shifters. Read more about the power supply in [3.3 Power Supply](#).

The on-board debugger and level shifters actively drive data and clock signals used for programming and debugging (DBG0, DBG1, and DBG2). Usually the external resistor on these signals can be ignored. Pull-down resistors are required on the ICSP™ data and clock signals to debug PIC® microcontrollers.

DBG3 is an open-drain connection and requires a pull-up resistor to function.

PIC16F15276 Curiosity Nano has pull-down resistors R204 and R205 connected to the ICSP data and clock signal (DBG0 and DBG1). There is also a pull-up resistor R200 connected to the MCLR signal (DBG3). The location of pull resistors is shown in the [7.2 Assembly Drawing](#) in the appendix.



Remember:

- Connect GND and VTG to the external microcontroller
- Tie the VOFF pin to GND if the external hardware has a power supply
- Make sure there are pull-down resistors on the ICSP data and clock signals (DBG0 and DBG1) to support the debugging of PIC microcontrollers

Figure 3-10. Curiosity Nano Standard Pinout

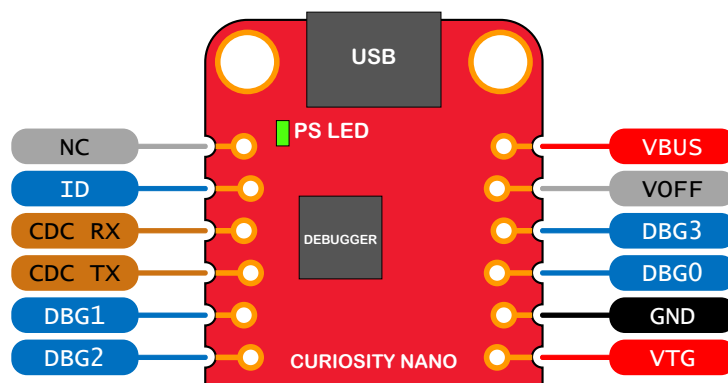


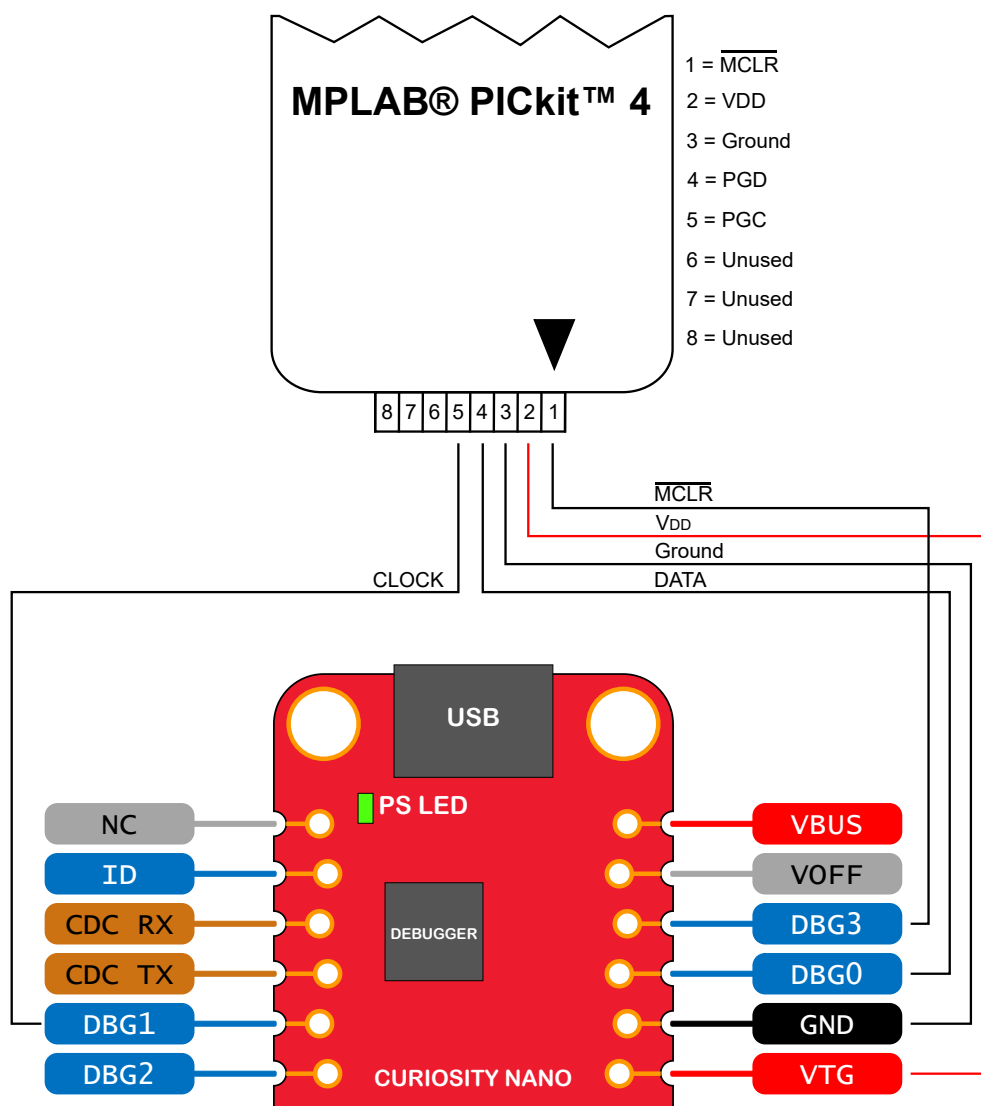
Table 3-4. Programming and Debugging Interfaces

Curiosity Nano Pin	UPDI	ICSP™	SWD
DBG0	UPDI	DATA	SWDIO
DBG1	—	CLK	SWCLK
DBG2	—	—	—
DBG3	—	MCLR	RESET

3.6 Connecting External Debuggers

Even though there is an on-board debugger, external debuggers can be connected directly to the PIC16F15276 Curiosity Nano to program/debug the PIC16F15276. The on-board debugger keeps all the pins connected to the PIC16F15276 and board edge in tri-state when not actively used. Therefore, the on-board debugger will not interfere with any external debug tools.

Figure 3-11. Connecting the MPLAB® PICKit™ 4 In-Circuit Debugger/Programmer to PIC16F15276 Curiosity Nano



CAUTION

The MPLAB® PICKit™ 4 In-circuit Debugger/Programmer is capable of delivering high voltage on the MCLR pin. High voltage can permanently damage R110. If R110 is broken, the on-board debugger cannot enter Programming mode of the PIC16F15276 and will typically fail at reading the device ID.

CAUTION

To avoid contention between the external debugger and the on-board debugger, do not start any programming/debug operation with the on-board debugger through MPLAB® X IDE or mass storage programming while the external tool is active.

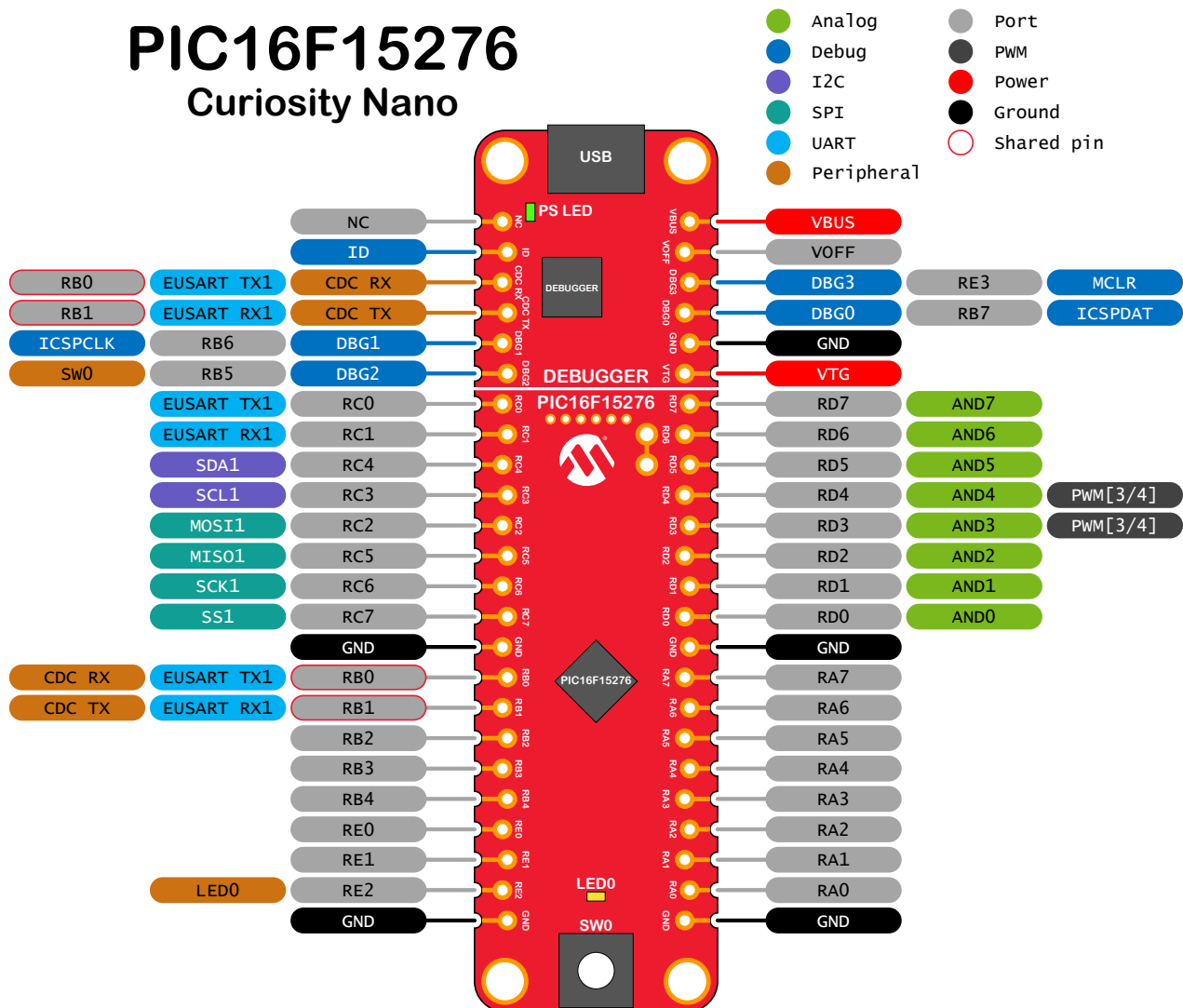
4. Hardware User Guide

4.1 Connectors

4.1.1 PIC16F15276 Curiosity Nano Pinout

All the PIC16F15276 I/O pins are accessible at the edge connectors on the board. The image below shows the board pinout.

Figure 4-1. PIC16F15276 Curiosity Nano Pinout



Info: Peripheral signals shown in the image above, such as UART, I²C, SPI, ADC, PWM, and others, are shown at specific pins to comply with the Curiosity Nano Board standard. These signals can usually be routed to alternate pins using the Peripheral Pin Select (PPS) feature in the PIC16F15276.

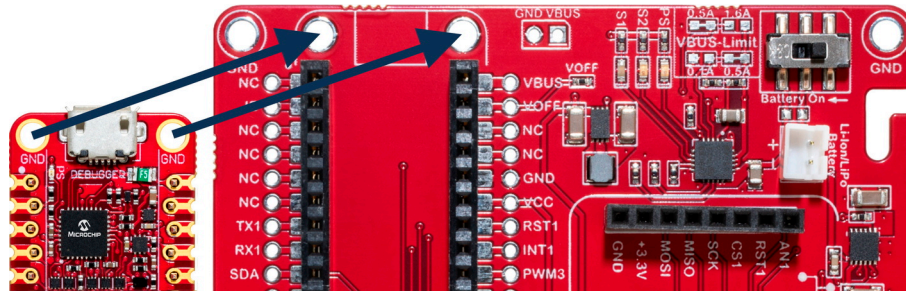
4.1.2 Using Pin-Headers

The edge connector footprint on PIC16F15276 Curiosity Nano has a staggered design where each hole is shifted 8 mil (~0.2 mm) off-center. The hole shift allows the use of regular 100 mil pin-headers on the board without soldering. Once the pin-headers are firmly in place, they can be used in normal applications like pin sockets and prototyping boards without any issues.

Figure 4-2. Attaching Pin-Headers to the Curiosity Nano Board



Figure 4-3. Connecting to Curiosity Nano Base for Click boards™



Tip: Start at one end of the pin-header and gradually insert the header along the length of the board. Once all the pins are in place, use a flat surface to push them in.



Tip: For applications where the pin-headers will be used permanently, it is still recommended to solder them in place.



Important: Once the pin-headers are in place, they are hard to remove by hand. Use a set of pliers and carefully remove the pin-headers to avoid damage to the pin-headers and PCB.

4.2 Peripherals

4.2.1 LED

There is one yellow user LED available on the PIC16F15276 Curiosity Nano board that can be controlled by either GPIO or PWM. Driving the connected I/O line to GND can also activate the LED.

Table 4-1. LED Connection

PIC16F15276 Pin	Function	Shared Functionality
RE2	Yellow LED0	Edge connector

4.2.2 Mechanical Switch

The PIC16F15276 Curiosity Nano board has one mechanical switch, which is a generic user-configurable switch. Pressing the switch will drive the I/O line to ground (GND).



Tip: There is no externally connected pull-up resistor on the switch. Make sure that an internal pull-up resistor is enabled on pin RB5 to use the switch.

Table 4-2. Mechanical Switch

PIC16F15276 Pin	Description	Shared Functionality
RB5	User switch (SW0)	Edge connector , On-board debugger

4.2.3 On-Board Debugger Implementation

PIC16F15276 Curiosity Nano features an on-board debugger that can be used to program and debug the PIC16F15276 using ICSP. The on-board debugger also includes a virtual serial port (CDC) interface over UART and debug GPIO. MPLAB® X IDE can be used as a front-end for the on-board debugger for programming and debugging. [MPLAB Data Visualizer](#) can be used as a front-end for the CDC and debug GPIO.

4.2.3.1 On-Board Debugger Connections

The table below shows the connections between the target and the debugger section. All connections between the target and the debugger are tri-stated as long as the debugger is not actively using the interface. Hence, since there are little contaminations of the signals, the pins can be configured to anything the user wants.

For further information on how to use the capabilities of the on-board debugger, see [3.1 On-Board Debugger Overview](#).

Table 4-3. On-Board Debugger Connections

PIC16F15276 Pin	Debugger Pin	Function	Shared Functionality
RB1	CDC TX	EUSART1 RX (PIC16F15276 RX line)	Edge connector
RB0	CDC RX	EUSART1 TX (PIC16F15276 TX line)	Edge connector
RB7	DBG0	ICSPDAT	Edge connector
RB6	DBG1	ICSPCLK	Edge connector
RB5	DBG2	GPIO0/SW0	Edge connector , SW0
RE3	DBG3	MCLR	Edge connector

5. Hardware Revision History and Known Issues

This user guide provides information about the latest available revision of the board. The following sections contain information about known issues, a revision history of older revisions, and how older revisions differ from the latest revision.

5.1 Identifying Product ID and Revision

There are two ways to find the revision and product identifier of the PIC16F15276 Curiosity Nano: either by utilizing the MPLAB® X IDE Kit Window or by looking at the sticker on the bottom side of the PCB.

By connecting PIC16F15276 Curiosity Nano to a computer with MPLAB® X IDE running, the Kit Window will pop up. The first six digits of the serial number, listed under kit information, contain the product identifier and revision.



Tip: If closed, the Kit Window can be opened in MPLAB® X IDE through the menu bar **Window > Kit Window**.

The same information is found on the sticker on the bottom side of the PCB. Most boards will have the identifier and revision printed in plain text as A09-nnnn\rr, where “nnnn” is the identifier, and “rr” is the revision. Boards with limited space have a sticker with only a data matrix code containing the product identifier, revision, and serial number.

The serial number string has the following format:

```
"nnnnrrssssssssss"
```

n = product identifier

r = revision

s = serial number

The product identifier for PIC16F15276 Curiosity Nano is A09-3431.

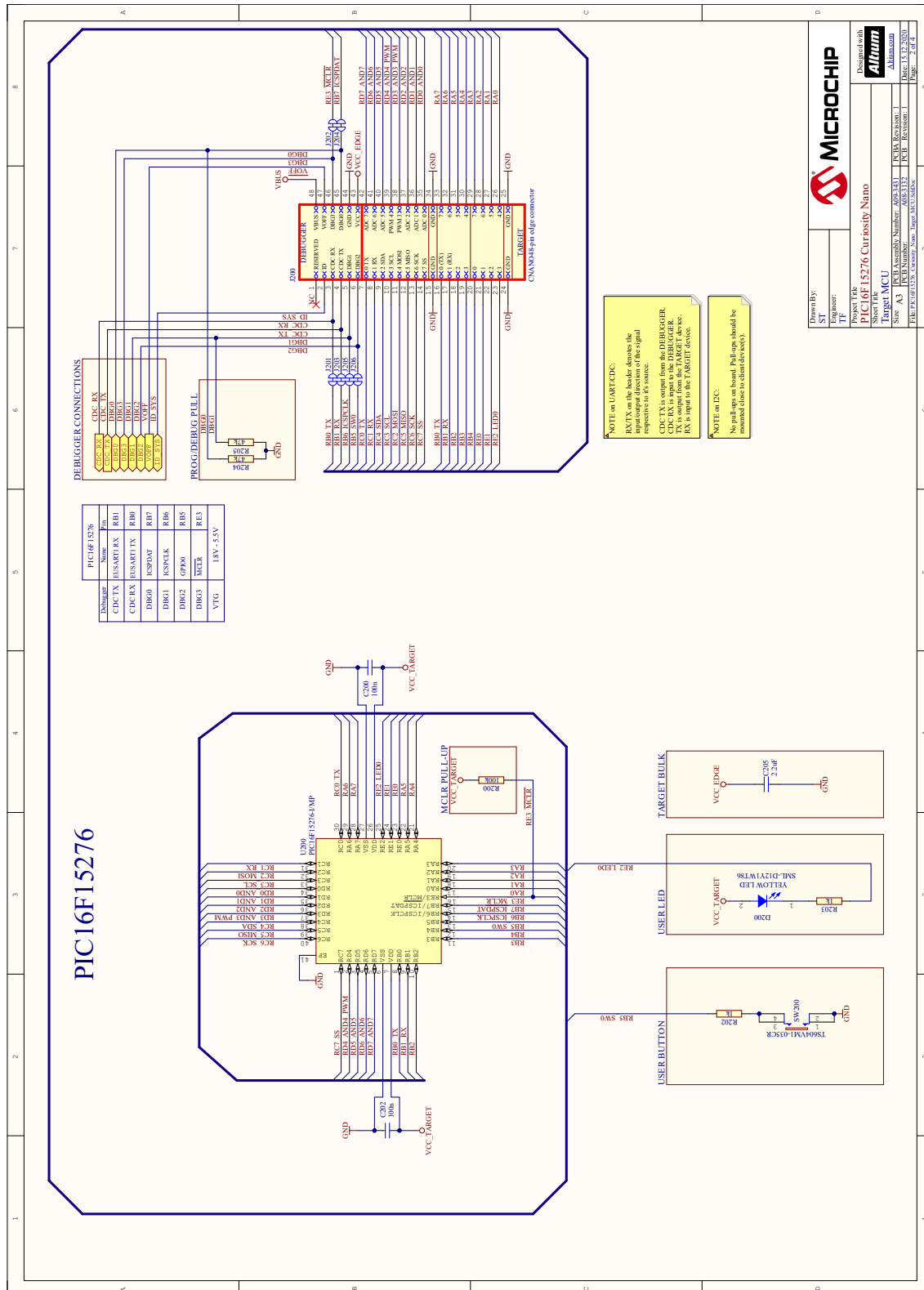
5.2 Revision 1

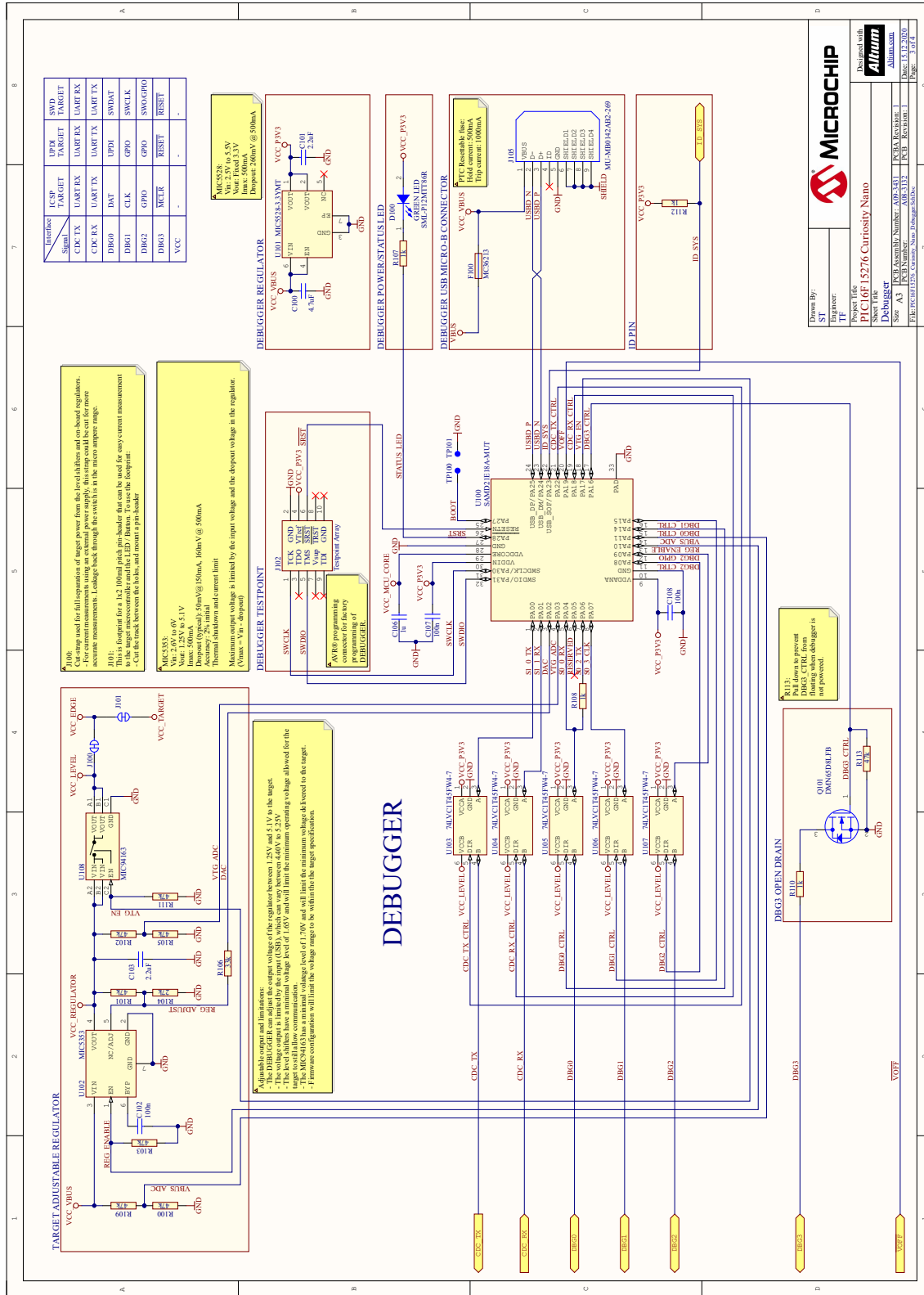
Revision 1 is the initially released version.

6. Document Revision History

Revision	Date	Description
A	04/2021	Initial document release.

Figure 7-1. PIC16F15276 Curiosity Nano Schematic





7.2 Assembly Drawing

Figure 7-2. PIC16F15276 Curiosity Nano Assembly Drawing Top

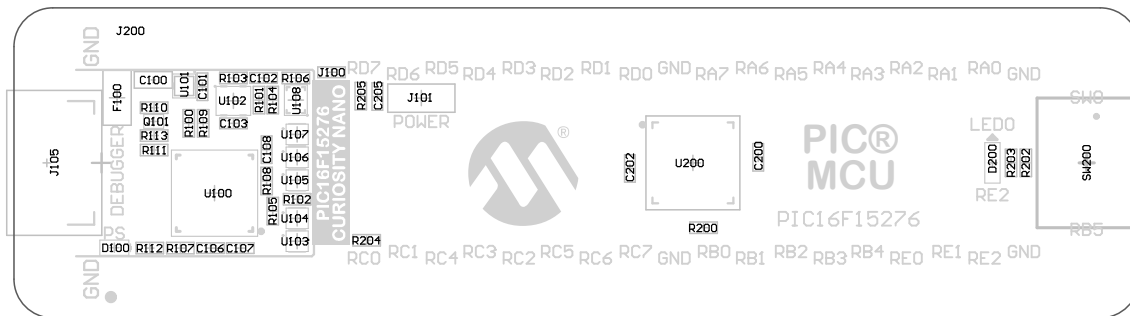
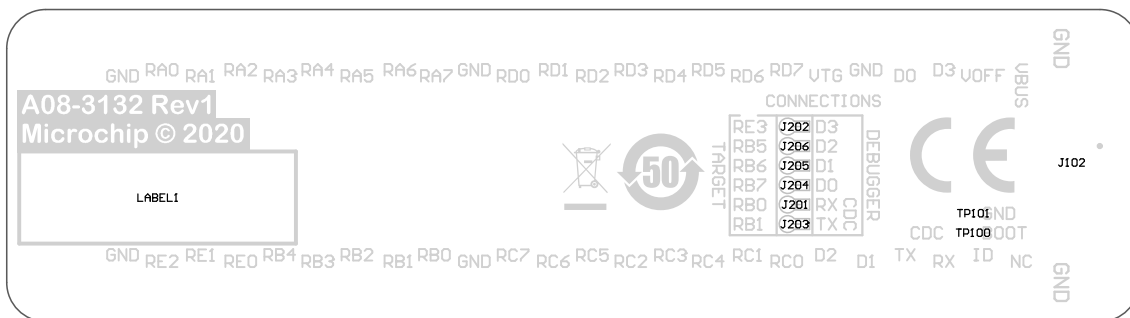
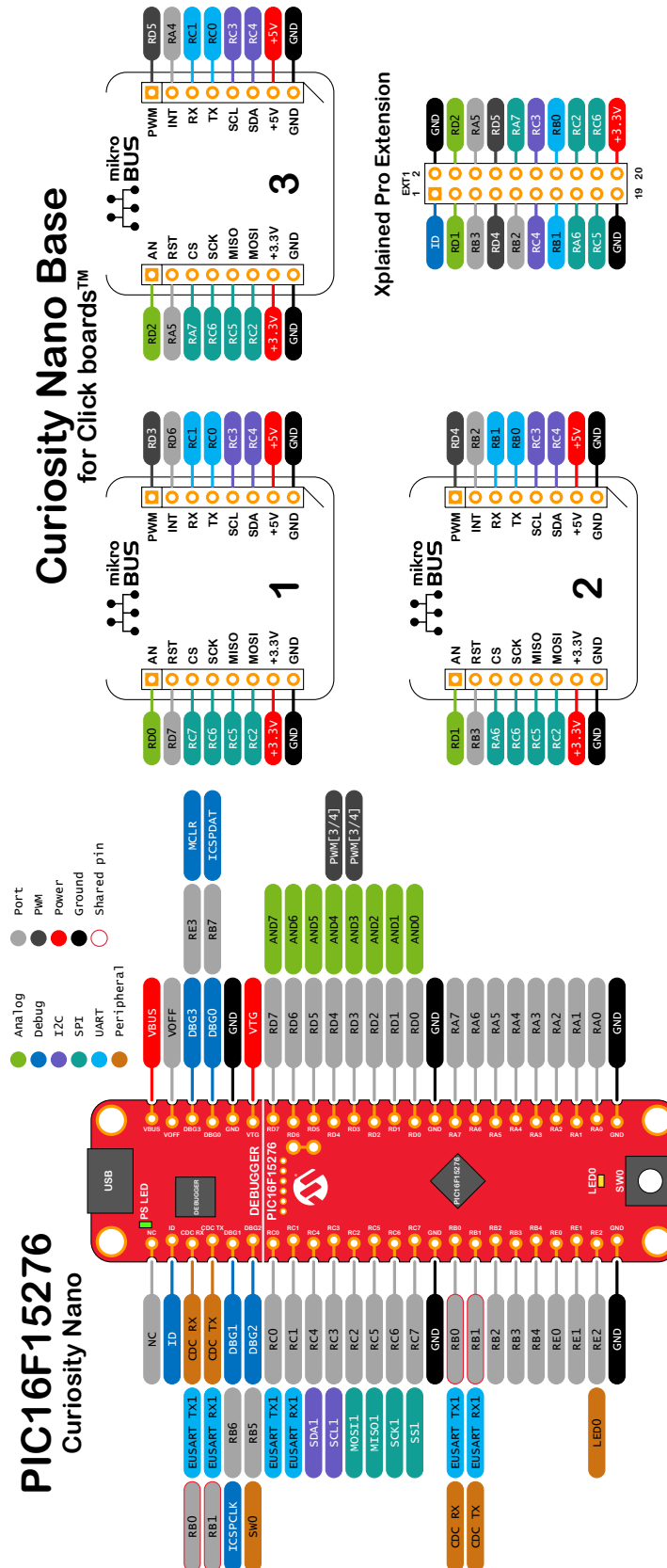


Figure 7-3. PIC16F15276 Curiosity Nano Assembly Drawing Bottom



7.3 Curiosity Nano Base for Click boards™

Figure 7-4. PIC16F15276 Curiosity Nano Pinout Mapping



7.4 Disconnecting the On-Board Debugger

The on-board debugger and level shifters can be completely disconnected from the PIC16F15276.

The block diagram below shows all connections between the debugger and the PIC16F15276. The rounded boxes represent connections to the board edge. The signal names shown are also printed in silkscreen on the bottom side of the board.

To disconnect the debugger, cut the straps shown in [Figure 7-6](#).



Attention: Cutting the GPIO straps to the on-board debugger will disable the virtual serial port, programming, debugging, and data streaming. Cutting the power supply strap will disconnect the on-board power supply.



Tip: Reconnect any cut connection by using solder. Alternatively, mount a 0Ω 0402 resistor.



Tip: When the debugger is disconnected, an external debugger can be connected to holes, as shown in [Figure 7-6](#). Details about connecting an external debugger are described in [3.6 Connecting External Debuggers](#).

Figure 7-5. On-Board Debugger Connections Block Diagram

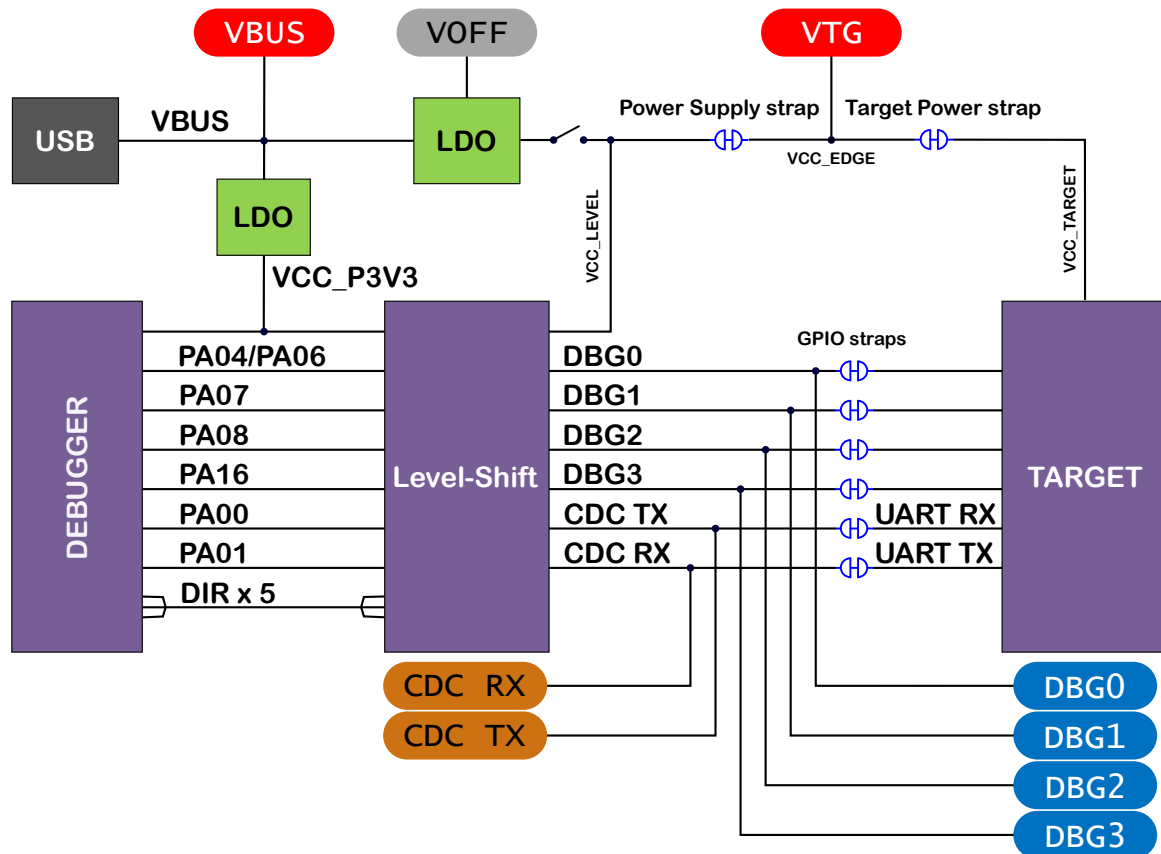
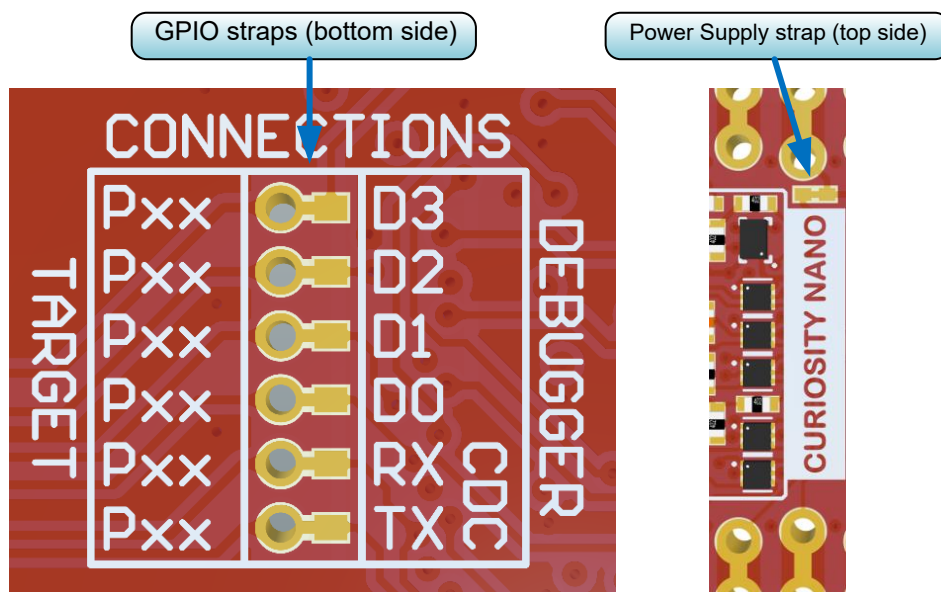


Figure 7-6. On-Board Debugger Connection Cut Straps



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

Microchip Devices Code Protection Feature

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

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "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 is provided for the sole purpose of designing with and using Microchip products. Information 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.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. 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, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, 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, 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, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, maxCrypto, maxView, 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, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SMART-I.S., storClad, SQL, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, 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.

© 2021, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-8155-3

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	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

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Microchip:](#)

[EV35F40A](#)