# dsPIC33AK512MPS506 Curiosity Nano User Guide

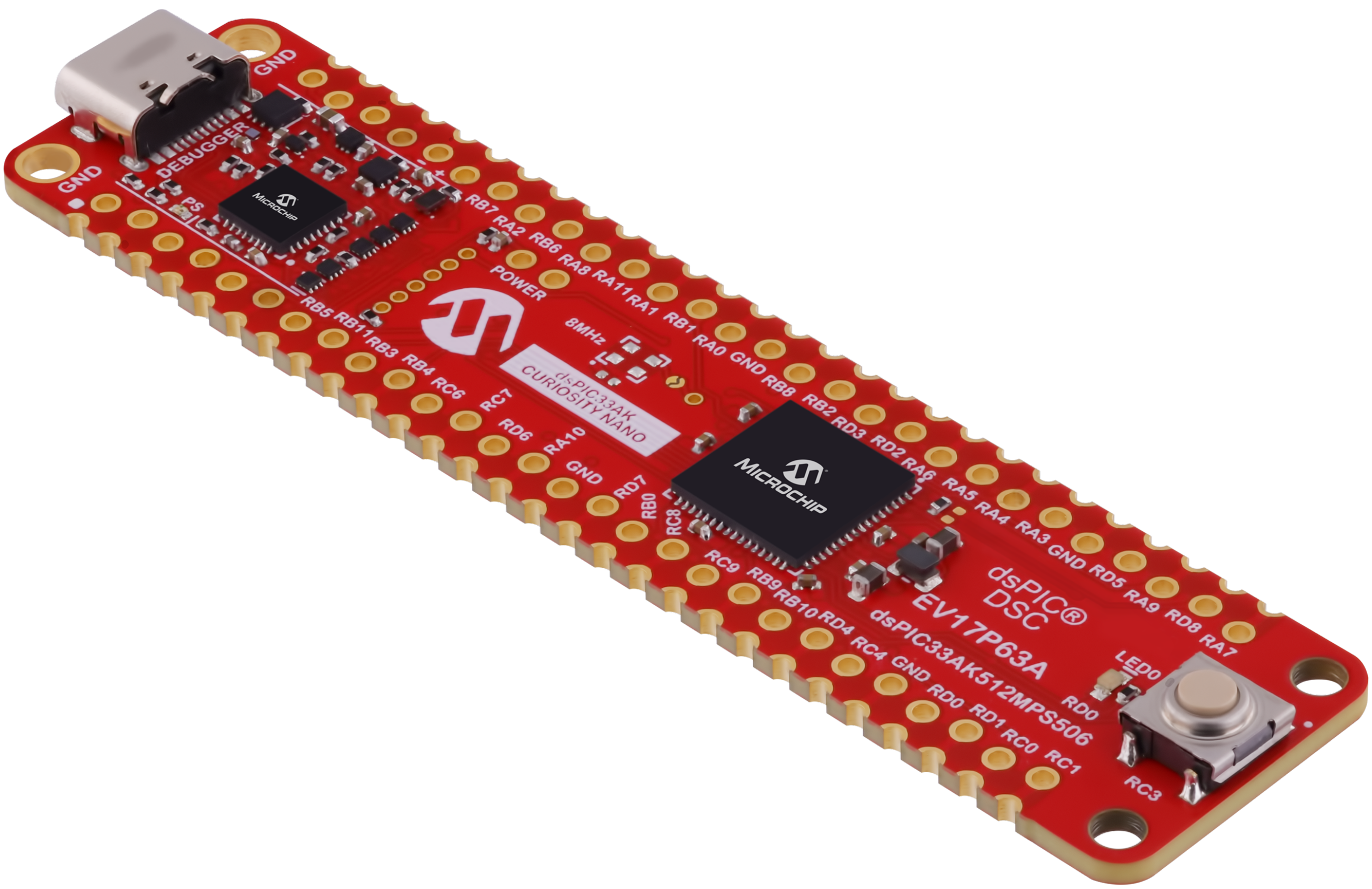
## Preface

The dsPIC33AK512MPS506 Curiosity Nano evaluation kit is a hardware platform to evaluate microcontrollers in the dsPIC33AK Family. This board has the dsPIC33AK512MPS506 Digital Signal Controller (DSC) mounted.

Supported by MPLAB® X IDE, the board provides easy access to the features of the dsPIC33AK512MPS506 to explore integrating the device into a custom design.

The Curiosity Nano series of evaluation boards includes an on-board debugger. No external tools are necessary to program and debug the dsPIC33AK512MPS506.

**Figure 1.** dsPIC33AK Curiosity Nano




**dsPIC33AK512MPS506 Website** - Find documentation, data sheets, and sample and for-purchase microcontrollers.

**dsPIC33AK512MPS506 Curiosity Nano Website** - Find kit information, the latest user guide and design documentation.

# Table of Contents

# 1. Introduction

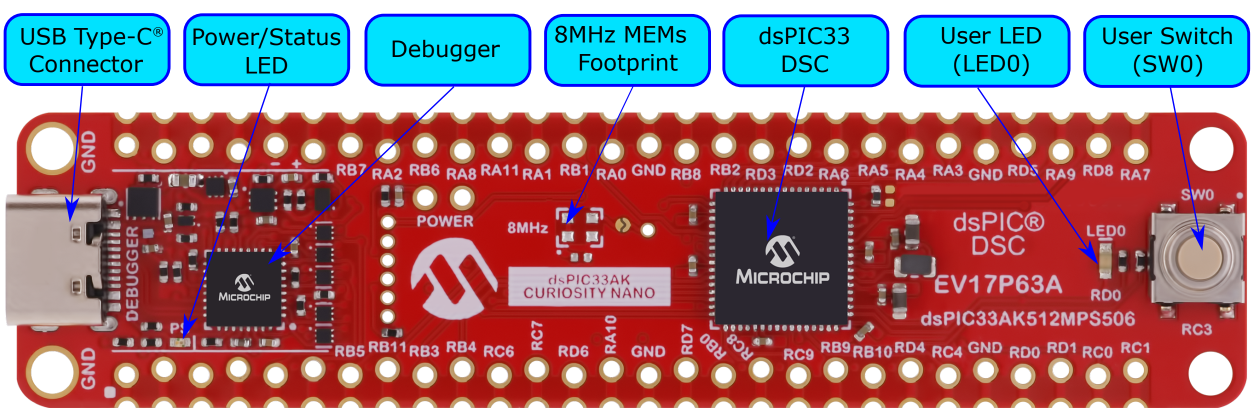
## 1.1. Features

- dsPIC33AK512MPS506 Digital Signal Controller
- One Yellow User LED
- One Mechanical User Switch
- Footprint for 8 MHz MEMs Oscillator
- 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
  - 3.0V to 3.6V output voltage (limited by USB input voltage)
  - 500 mA maximum output current (limited by ambient temperature and output voltage)

## 1.2. Board Overview

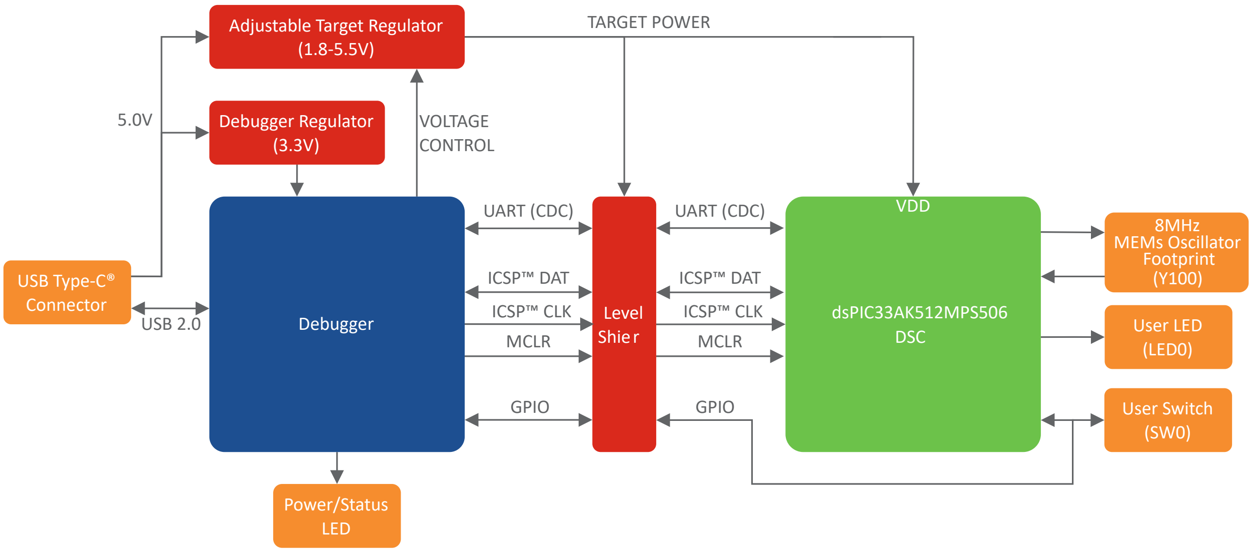
The Microchip dsPIC33AK512MPS506 Curiosity Nano evaluation kit is a hardware platform to evaluate the dsPIC33AK512MPS506 Digital Signal Controller.

**Figure 1-1.** dsPIC33AK512MPS506 Curiosity Nano Board Overview



## 1.3. Block Diagram

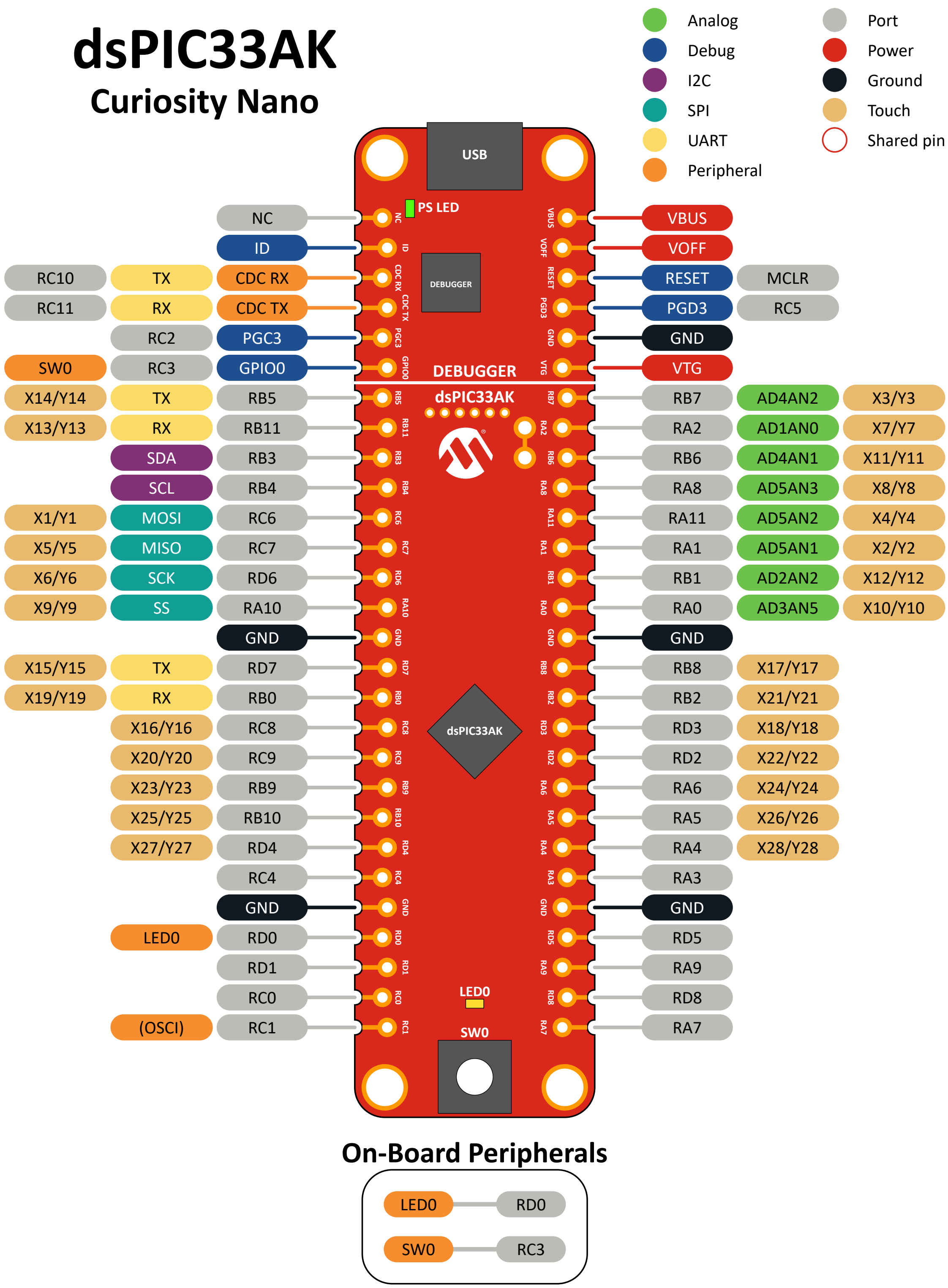
**Figure 1-2.** Block Diagram




## 1.4. Pinout

All the dsPIC33AK512MPS506 I/O pins are accessible at the edge connectors on the board. Figure 1-3 shows the board pinout. Refer to the I/O Multiplexing and Configuration section in the dsPIC33AK512MPS506 data sheet for all available functions on each pin.

**Figure 1-3.** dsPIC33AK512MPS506 Curiosity Nano Pinout



**Info:** Peripheral signals shown in the image, such as UART, $I^2C$, 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 dsPIC33AK512MPS506.

## 1.5. Preprogrammed Application

The dsPIC33AK512MPS506 mounted on the Curiosity Nano Evaluation Kit is preprogrammed with an example application that showcases the usage of the UART protocol and I/O toggling via user interface push-button activity. A link to the demo code can be found on the board's **web page**.

# 2. Getting Started

## 2.1. Quick Start

Steps to start exploring the dsPIC33AK512MPS506 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/Develop the application code.
6. Connect a USB cable (USB-C) between the PC and the debug USB port on the board.
7. Program your application onto the device.

   The dsPIC33AK512MPS506 device on the dsPIC33AK512MPS506 Curiosity Nano board is programmed and debugged by the on-board debugger. Therefore, no external programmer or debugger tool is required.

   **Notes:**
   - MPLAB® X IDE supports XC C compilers and the AVR® and Arm® Toolchains (GCC Compilers)
   - Microchip's XC8 C Compiler supports all 8-bit PIC® and AVR® microcontrollers
   - Included in the Microchip Studio download are the AVR® and Arm® Toolchains (GCC Compilers)

### 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®. The drivers for the board are included with MPLAB X IDE. When creating a new project, the part number on the Curiosity Nano will be detected, as will the debug tool.

### 2.1.2. Kit Window

When the board is connected to a computer and powered on, the green status LED will be lit, and the MPLAB X IDE will auto-detect which boards are connected. The Kit Window in MPLAB X IDE will present relevant information like data sheets and board documentation.

**Tip:** If closed, reopen the Kit Window in MPLAB X IDE 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 dsPIC33AK512MPS506 Curiosity Nano board, MPLAB X version 6.30 with device family pack "dsPIC33AK-MP_DFP" version 1.1.106 and tool pack "nEDBG_TP" version 1.17.882 or newer is 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's 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 dsPIC33AK512MPS506 Curiosity Nano board:

- **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** are available as a free, unrestricted-use download. Microchip’s MPLAB XC16 C Compiler is a comprehensive solution for your project’s software development on Windows, Mac OS or Linux. MPLAB XC16 supports all 16-bit PIC® MCUs and dsPIC® Digital Signal Controllers (DSCs).
- **MPLAB Code Configurator** is a free software plug-in that provides a graphical interface to configure peripherals and functions specific to your application.
- **Microchip Sample Store** is a place where one can order device samples.
- **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 the on-board debugger’s Data Gateway Interface, as found on Curiosity Nano and Xplained Pro boards.
- **Code Examples on GitHub** - Get started with code examples.
- **dsPIC33AK512MPS506 Curiosity Nano Website** - Kit information, the latest user guide and design documentation.
- **dsPIC33AK512MPS506 Curiosity Nano on Microchip Direct** - Purchase this kit from Microchip Direct.

# 3. On-Board Debugger

Curiosity Nano is an evaluation platform of small boards with low-pin count Digital Signal Controller (DSC) boards with on-board debuggers and access to most of the Digital Signal Controller's I/Os. The Curiosity Nano platform offers easy integration with MPLAB X IDE. All boards are identified in the IDE. When connected, a kit window appears with links to key documentation, including relevant user's 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

dsPIC33AK512MPS506 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 dsPIC33AK512MPS506 in MPLAB X IDE
- A virtual serial port (CDC) that is connected to a Universal Asynchronous Receiver/Transmitter (UART) on the dsPIC33AK512MPS506 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 dsPIC33AK512MPS506 Curiosity Nano board. The table below shows how the different operation modes control the LED.

**Table 3-1.** On-Board Debugger LED Control

| Operation Mode | Power and Status LED |
|---|---|
| Boot Loader | The LED blinks slowly during power-up. |
| Power-up | The LED is ON. |
| Normal Operation | The LED is ON. |
| Programming | The LED blinks slowly during programming/debugging. |
| Fault | The LED blinks rapidly if a power Fault is detected. |
| Sleep/Off | The LED is OFF. The on-board debugger is either in 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 about 5 Hz.

### 3.1.1. 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.

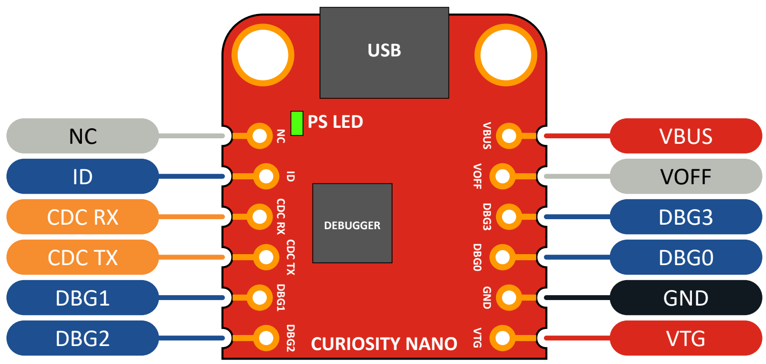
**Table 3-2.** Curiosity Nano Standard Pinout

| 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 |

**Table 3-2.** Curiosity Nano Standard Pinout (continued)

| Debugger Signal | Target MCU | Description |
|---|---|---|
| DBG3 | $\overline{\text{MCLR}}$ | Reset line |
| NC | — | No connect |
| $V_{BUS}$ | — | $V_{BUS}$ voltage for external use |
| $\overline{\text{VOFF}}$ | — | $\overline{\text{Voltage Off}}$ input. Disables the target regulator and target voltage when pulled low. |
| VTG | — | Target voltage |
| GND | — | Common ground |

**Figure 3-1.** Curiosity Nano Standard Pinout




## 3.1.2. Debugger USB Enumeration

The on-board debugger on the dsPIC33AK512MPS506 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 dsPIC33AK512MPS506 using MPLAB X IDE.

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

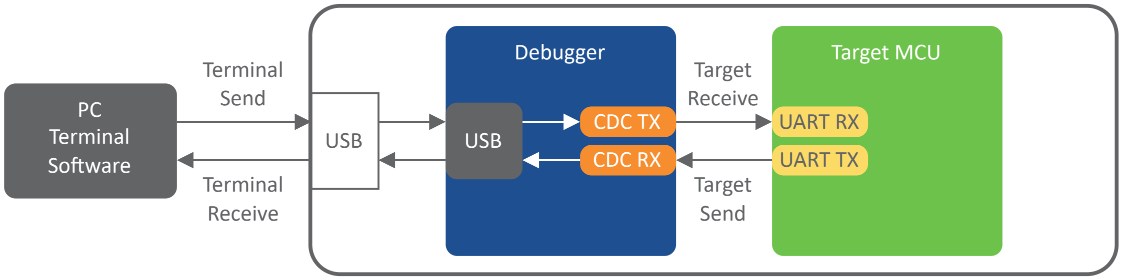
## 3.1.3. Virtual Serial Port (CDC)

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

### 3.1.3.1. Overview

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

**Figure 3-2.** CDC Connection




**Info:** The debugger's CDC TX pin is connected to a UART RX pin on the target for receiving characters from the host computer, as shown in the figure above. Similarly, the debugger's CDC RX pin is connected to a UART TX pin on the target for transmitting characters to the host computer.

#### 3.1.3.2. 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. The MPLAB X IDE installation includes this driver.

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 permission to access the CDC.

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

**Info:** For all operating systems, use a terminal emulator that supports DTR signaling. See Signaling.

#### 3.1.3.3. 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. The 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.

#### 3.1.3.4. Signaling

During USB enumeration, the host OS will start the 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 sending and receiving data will not be enabled.

The terminal must assert the DTR signal when it connects to the host. 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 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 more data will be transferred to the host computer
- Debugger UART transmitter will continue to send queued data ready for transfer, 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, and no further data will be transferred to the host computer
- Debugger UART transmitter will continue to send queued data ready for transfer, but no new data is accepted from the host computer
- Once the ongoing transmission is complete, level shifters (if available) are disabled, and 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 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, meaning the lines are floating during power-up. The target device may enable the internal pull-up resistor on the pin connected to the debugger's CDC TX pin to avoid glitches resulting in unpredictable behavior like framing errors.

#### 3.1.3.5. Advanced Use

**CDC Override Mode**

In ordinary operation, the on-board debugger is a 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 send characters out of the debugger's CDC TX pin. The filename and extension are trivial, but the text file will 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 configured, the previously used baud rate still applies.

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

- The maximum message length will 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 up to 4 KB will work on most systems. The transfer will be completed upon encountering the first NULL character 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 when receiving a file via the 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, chunked into 64-byte USB frames. Each frame will be queued for transfer to the debugger's CDC TX pin. Sending a small amount of data per frame can be inefficient, particularly at low baud rates, as the on-board debugger buffers frames but 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.

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

**Sending Break Characters**

The host can send a UART break character to the device using the CDC, which can be useable for resetting a receiver state machine or signaling an exception condition from the host to the application running on the device.

A break character is a sequence of at least 11 zero bits transmitted from the host to the device.

Not all UART receivers support detecting a break, but a correctly formed break character usually triggers a framing error on the receiver.

Sending a break character using the debugger's CDC has the following limitations:

- Sending a break will cause any data being sent to be lost. Be sure to wait a sufficient amount of time to allow all characters in the transmission buffer to be sent (see the above section) before sending the break, which is also in line with expected break character usage. For example, reset a receiver state machine after a timeout occurs while waiting for returning data to the host.
- The CDC specification allows for debugger-timed breaks of up to 65,534 ms in duration to be requested. For simplicity, the debugger will limit the break duration to a maximum of 11 bit durations at its minimum supported baud rate.

- The CDC specification allows for indefinite host-timed breaks. It is the terminal application/user's responsibility to release the break state in this case.

**Note:** Sending break characters is available in debugger firmware version 1.24 and later.

### 3.1.4. 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 and UF2 formatted files into the target device's memory
- Write access for simple text files for utility purposes

**Note:** Support for UF2 format is available in debugger firmware version 1.31 or later.

#### 3.1.4.1. Mass Storage Device Implementation

The on-board debugger implements a highly optimized variant of the FAT12 file system with 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 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 does not reflect the target device's Flash size. When programming an Intel HEX or UF2 file, the binary data are encoded in ASCII with metadata providing a vast overhead, so 1 MB is a trivially chosen value for disk size.

It is not possible to format the CURIOSITY drive. The filename may appear in the disk directory listing when programming a file to the target, which is merely the operating system's view of the directory that, in reality, has not been updated. 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 previously programmed application.

Copy a text file starting with "`CMD:ERASE`" onto the disk to erase the target device.

By default, the CURIOSITY drive contains several read-only files for generating icons, 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` – redirects 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 and device.
- `STATUS.TXT` – a text file containing the programming status of the board

**Info:** The on-board debugger dynamically updates `STATUS.TXT`. The contents may not reflect the correct status as the OS may cache it.

#### 3.1.4.2. 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 3-3.** Special File Commands

| Command Content | Description |
|---|---|
| `CMD:ERASE` | Executes a target chip erase |
| `CMD:SEND_UART=` | Sends a string of characters to the CDC UART. See "CDC Override Mode." |
| `CMD:SEND_9600=`<br>`CMD:SEND_115200=`<br>`CMD:SEND_460800=` | Sends a string of characters to the CDC UART at the specified baud rate. Note that only the baud rates explicitly specified here are supported. See "CDC Override Mode." (Debugger firmware v1.25.6 or newer.) |
| `CMD:RESET` | Resets the target device by entering Programming mode and exiting Programming mode immediately afterward. The exact timing can vary according to the programming interface of the target device. (Debugger firmware v1.25.6 or newer.) |
| `CMD:POWERTOGGLE` | Powers down the target and restores it after a 100 ms delay. If external power is provided, this has no effect. (Debugger firmware v1.25.6 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 v1.25.6 or newer.) |
| `CMD:1V8` | Sets the target voltage to 1.8V. If using external power, this has no effect. (Debugger firmware v1.25.6 or newer.) |
| `CMD:3V3` | Sets the target voltage to 3.3V. If using external power, this has no effect. (Debugger firmware v1.25.6 or newer.) |

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

### 3.1.5. Data Gateway Interface (DGI)

The Data Gateway Interface (DGI) is a USB interface that transports 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 any debug GPIO data. It is available as a plug-in for MPLAB X IDE or as a stand-alone application that can be used in parallel with MPLAB® X IDE.

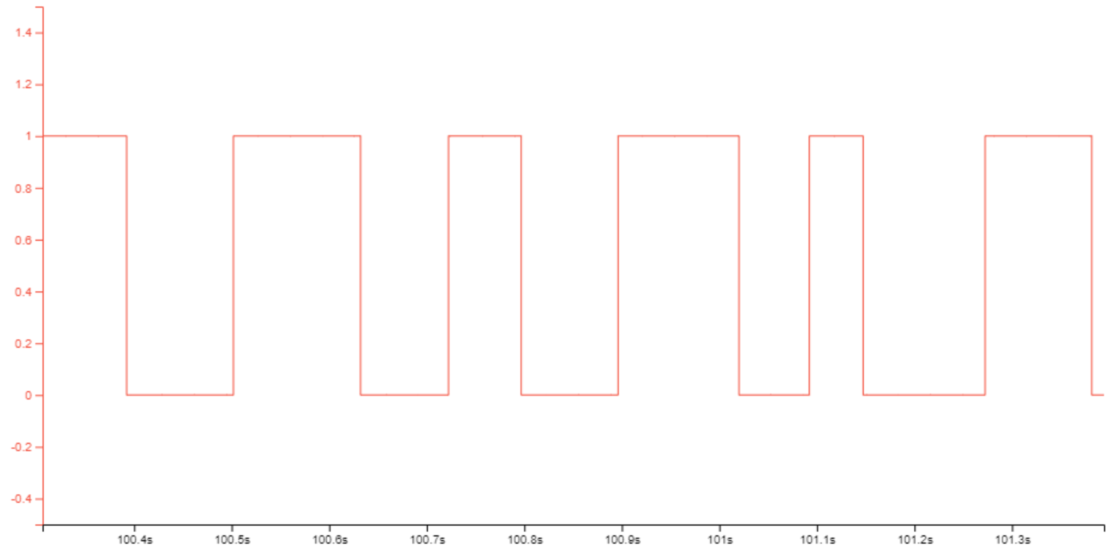
The dsPIC33AK512MPS506 Curiosity Nano has one available debug GPIO channel DGI GPIO0

#### 3.1.5.1. 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 low-frequency events on a time axis, such as when given Application state transitions occur.

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

**Figure 3-3.** Monitoring Debug GPIO with MPLAB Data Visualizer




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

**Important:** Although capturing higher frequency signal bursts is possible, the signals' frequency range where the 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 aborting the DGI session.

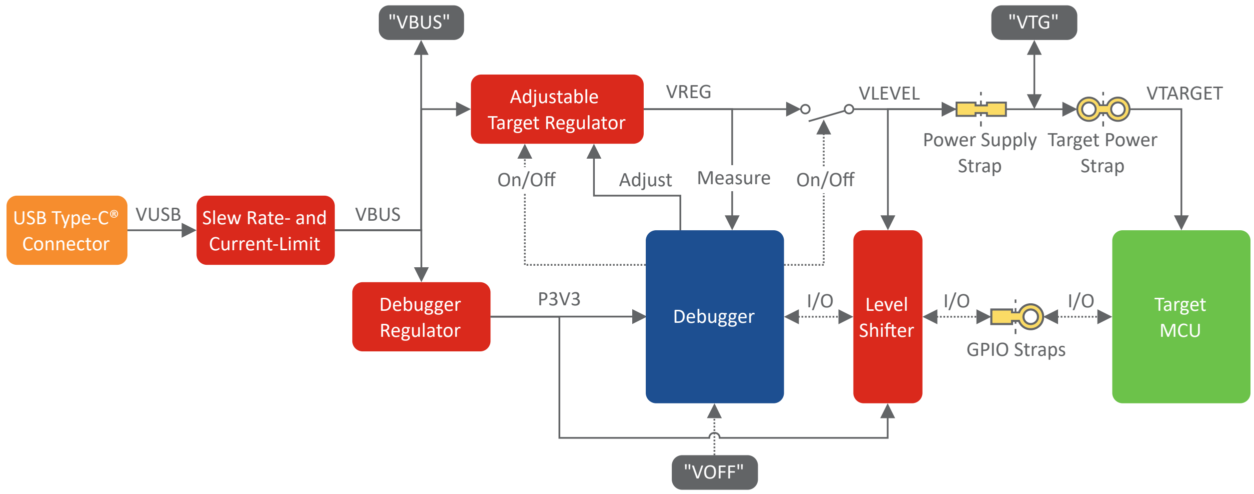
#### 3.1.5.2. Timestamping

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

## 3.2. Power Supply

The USB port powers the board. It contains two LDO regulators, one to generate 3.3V for the on-board debugger and an adjustable LDO regulator for the target dsPIC33AK512MPS506 Digital Signal Controller and its peripherals. The voltage from a USB connector can vary between 4.4V and 5.25V (according to the USB specification) and will limit the maximum voltage supplied to the target. Figure 3-4 shows the entire power supply system on dsPIC33AK512MPS506 Curiosity Nano.

**Figure 3-4.** Power Supply Block Diagram




## 3.2.1. 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 dsPIC33AK512MPS506 Digital Signal Controller. The voltage limits configured in the on-board debugger on the dsPIC33AK512MPS506 Curiosity Nano are 3.0–3.6V.

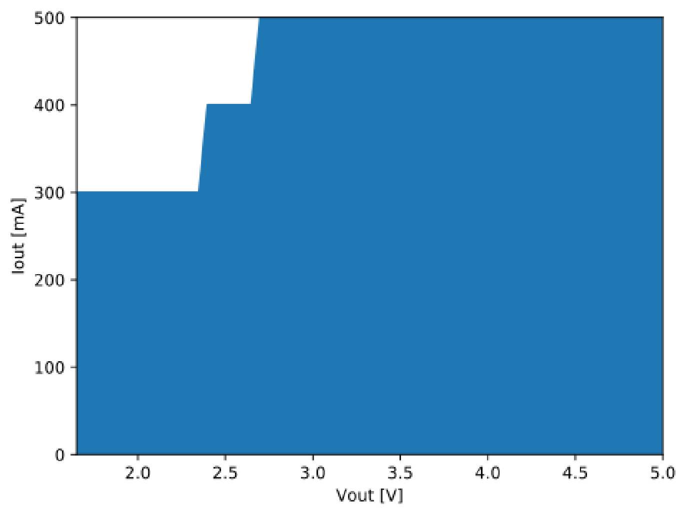
**Info:** The target voltage is set to 3.3V when the board is manufactured. It can be changed through the MPLAB X IDE project properties. Any change to the target voltage is persistent, even after a power toggle. The resolution is less than 5 mV but may be limited to 10 mV by the adjustment program.

**Info:** The voltage settings set up in MPLAB X IDE are not applied immediately to the board. The new voltage setting is applied to the board when accessing the debugger, such as pushing the Refresh Debug Tool Status button in the project dashboard tab or programming/reading program memory.

**Note:** The max input voltage of dsPIC33AK512MPS506 should never exceed 3.6V. Exceeding the max input could cause permanent damage to the device.

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. Figure 3-5 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 or under the set device voltage, an error condition will be flagged, and the target voltage regulator will be switched off, detecting and handling any short-circuit conditions. It will also detect and handle if an external voltage, which causes $V_{CC}$_TARGET to move outside the voltage setting monitoring window of ±100 mV, is suddenly applied to the VTG pin without setting the $V_{OFF}$ pin low.

**Info:** The on-board debugger has a monitoring window of $V_{CC}$_TARGET±100 mV. The on-board debugger status LED will blink rapidly if the external voltage is below this limit. The on-board debugger status LED will continue to shine if the external voltage is above this limit. When removing the external voltage, the status LED will start blinking rapidly until the on-board debugger detects the new situation and turns the target voltage regulator back on.

### 3.2.2. External Supply

Instead of the on-board target regulator, an external voltage can power the dsPIC33AK512MPS506 Curiosity Nano. When shorting the Voltage Off (VOFF) pin to the ground (GND) pin, 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, which will be detected by a pin-change interrupt to the on-board debugger, which controls the target voltage regulator accordingly.

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

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

WARNING The absolute maximum external voltage is 5.5V for the on-board level shifters, and the standard operating condition of the dsPIC33AK512MPS506 is 3.0–3.6V. Applying a higher voltage may cause permanent damage to the board.

**Info:** The on-board debugger monitors the voltage supplied to the board. If VOFF is not pulled low and the external power supplied differs by more than ±100 mV from the target regulator setting, the on-board debugger will shut off the target regulator and begin blinking the status LED rapidly, indicating an error condition. Once the input voltage returns within ±100 mV of the target regulator setting, the on-board debugger will switch on the target regulator and stop blinking the status LED.

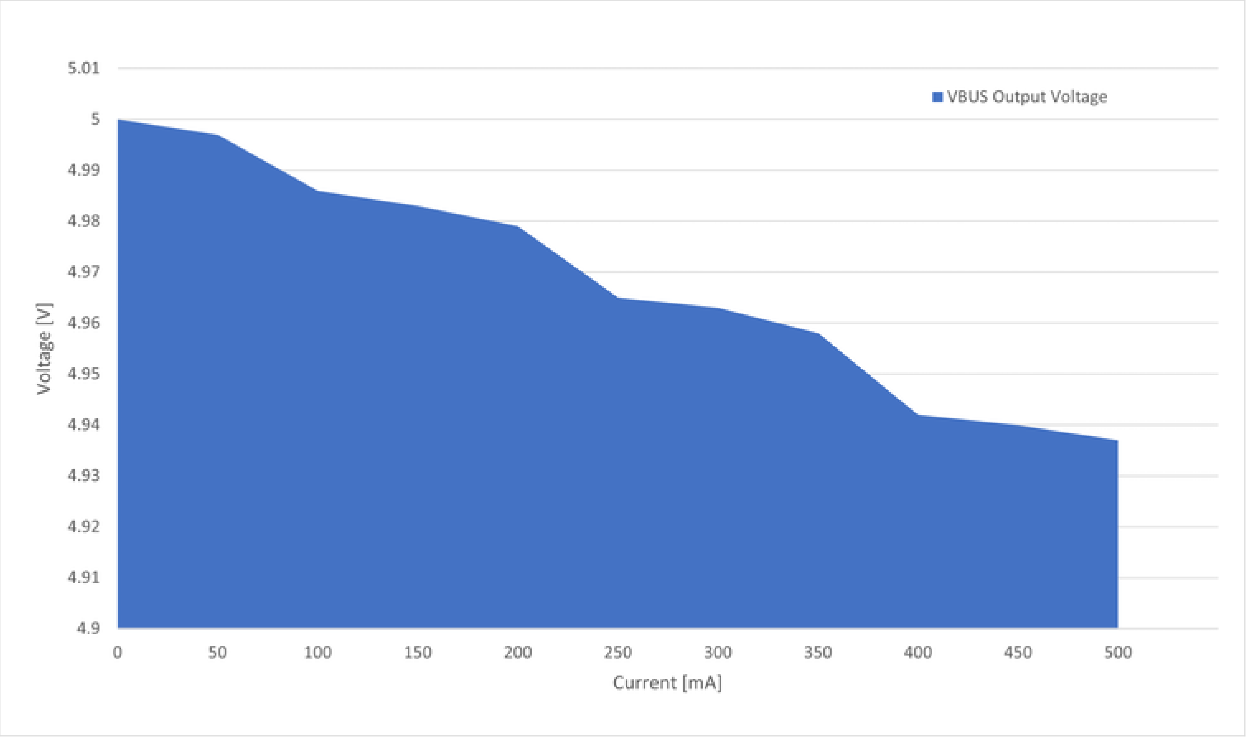
Programming, debugging, and data streaming are still possible with an external power supply. The USB cable will power the debugger and signal level shifters. Both regulators, the debugger, and the level shifters are powered down when the USB cable is removed.

**Info:** In addition to the power consumed by the dsPIC33AK512MPS506 and its peripherals, approximately 100 µA will be drawn from any external power source to power the on-board level shifters and voltage monitor circuitry when plugging a USB cable into the DEBUG connector on the board. When a USB cable is unplugged, some current is used to supply the level shifter's voltage pins, having a worst-case current consumption of approximately 5 µA. Typical values may be as low as 100 nA.

### 3.2.3. VBUS Output Pin

dsPIC33AK512MPS506 Curiosity Nano has a VBUS output pin that can be used to power external components that need a 5V supply. The VBUS output pin is protected by the same start-up delay with a slew rate and current limiter as the rest of the power supply. A side effect is a voltage drop on the VBUS output with higher current loads. The chart below shows the VBUS output voltage versus the current load of the VBUS output.

**Figure 3-6.** VBUS Output Voltage vs. Current




### 3.2.4. Power Supply Exceptions

This section summarizes most issues that can arise with the power supply.

#### Target Voltage Shuts Down

Not reaching the set target voltage can happen if the target section draws too much current at a given voltage, causing 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 4.4-5.25V) limits the maximum output voltage of the MIC5353 regulator at a given voltage setting and current consumption. If a higher output voltage is needed, use a USB power source with a higher input voltage or use an external voltage supply on the VTG pin.

#### Target Voltage is Different From Setting

An externally applied voltage to the VTG pin without setting the VOFF pin low can cause this. If the target voltage fluctuates by over 100 mV over/under the voltage setting, the on-board debugger will detect it, 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 blink rapidly if the target voltage is below 100 mV of the setting but will ordinarily turn on when it is more than 100 mV above it.

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

A full or partial short circuit can cause this and is a particular case of the issue above. Remove it, and the on-board debugger will re-enable the on-board target voltage regulator.

#### No Target Voltage and PS LED is Lit 1

This situation 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 situation can be an issue when cutting power jumper J200 and/or J201 and setting the target voltage regulator to a value within the specified voltage range for the target device. To fix this, solder a wire/bridge between the pads for J200/J201 or add a jumper on J201 if a pin header is mounted.

#### $V_{BUS}$ Output Voltage is Low or Not Present

If the VBUS output voltage is low or missing, the reason is probably a high-current drain on $V_{BUS}$, and the current limit set by U202 (MIC2008) is tripped and has cut off $V_{BUS}$ completely. Reduce the current consumption on the VBUS pin to fix this issue.

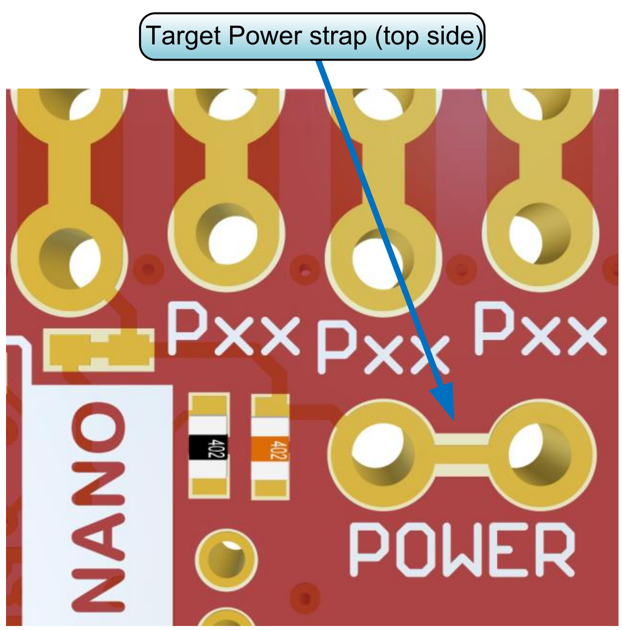
## 3.3. Low-Power Measurement

Power to the dsPIC33AK512MPS506 is connected from the on-board power supply and VTG pin through a 100 mil pin-header marked with POWER in silkscreen (J201). To measure the power consumption of the dsPIC33AK512MPS506 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:
   a. Tri-states any I/O connected to the on-board debugger.
   b. Sets the Digital Signal Controller to its lowest power Sleep mode.
5. Program the firmware into the dsPIC33AK512MPS506.

**Figure 3-7.** Target Power Strap



**Tip:** A 100-mil pin header can be soldered into the Target Power strap (J201) footprint for a simple connection of an ammeter. Place a jumper cap on the pin header once the ammeter is no longer needed.

**Info:** The on-board level shifters will draw a small amount of current even when not used. A maximum of 2 µA can be drawn from each I/O pin connected to a level shifter; the worst-case maximum for the five on-board level shifters is therefore 10 µA. 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 Figure 3-1. The on-board level shifters can be completely disconnected to prevent leakage, as described in Disconnecting the On-Board Debugger.

## 3.4. Programming External Microcontrollers

The on-board debugger on dsPIC33AK512MPS506 Curiosity Nano can be used to program and debug microcontrollers on external hardware.

### 3.4.1. 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 with a Curiosity Nano board can be programmed and debugged with the on-board debugger with Microchip Studio.

dsPIC33AK512MPS506 Curiosity Nano can program and debug external dsPIC33AK512MPS506 microcontrollers with MPLAB X IDE.

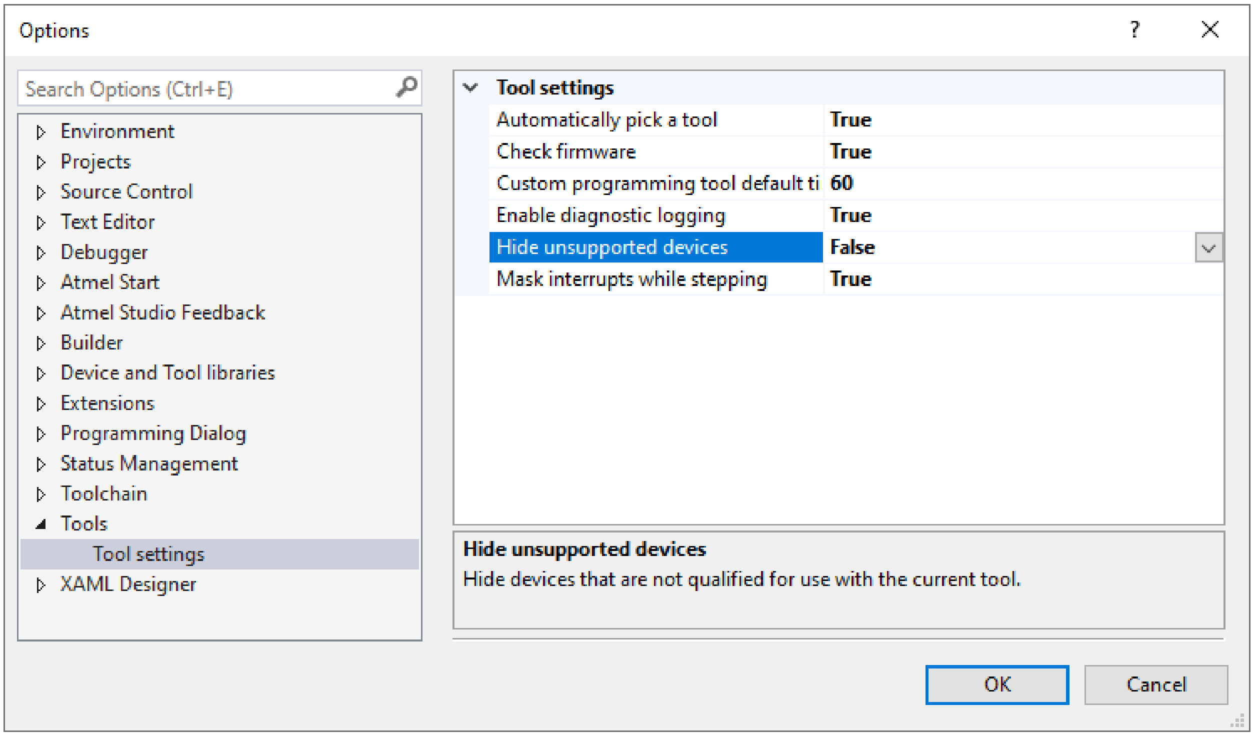
### 3.4.2. Software Configuration

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

To program and debug a different target device than the one mounted on the board, configure Microchip Studio to allow an independent 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** > **Tools Settings** category in the options window.
3. Set the **Hide unsupported devices** option to **False**.

**Figure 3-8.** Hide Unsupported Devices

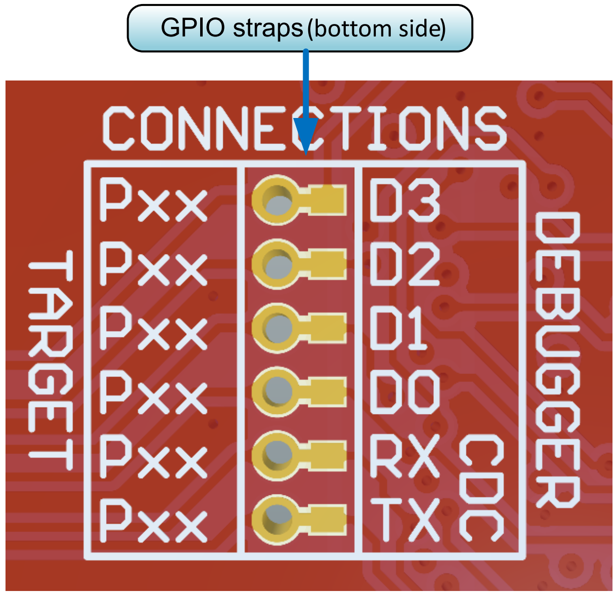



**Info:** Microchip Studio allows any target device and interface to be selected when the **Hide unsupported devices** setting is set to **False**, including digital signal controllers and interfaces not supported by the onboard debugger.

### 3.4.3. Hardware Modifications

The on-board debugger is connected to the dsPIC33AK512MPS506 by default. Remove these connections before any external microcontroller can be programmed or debugged. Cut the GPIO straps shown in Figure 3-9 with a sharp tool to disconnect the dsPIC33AK512MPS506 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 dsPIC33AK512MPS506 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 dsPIC33AK512MPS506.

### 3.4.4. Connecting to External Microcontrollers

Figure 3-10 and Table 3-4 show where to connect the programming and debugging signals 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 Power Supply.

The on-board debugger and level shifters actively drive data and clock signals used for programming and debugging (DBG0, DBG1 and DBG2). Pull-down resistors are required on the ICSP™ data and clock signals to debug PIC microcontrollers. All other interfaces are functional with or without pull-up or pull-down resistors.

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

dsPIC33AK512MPS506 Curiosity Nano has pull-down resistors R100 and R101 connected to the ICSP data and clock signal (DBG0 and DBG1). There is also a pull-up resistor R102 connected to the $\overline{\text{MCLR}}$ signal (DBG3). The location of pull resistors is shown in Assembly Drawing

**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

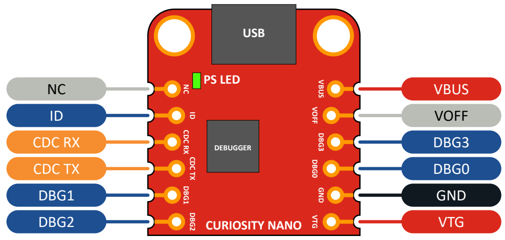


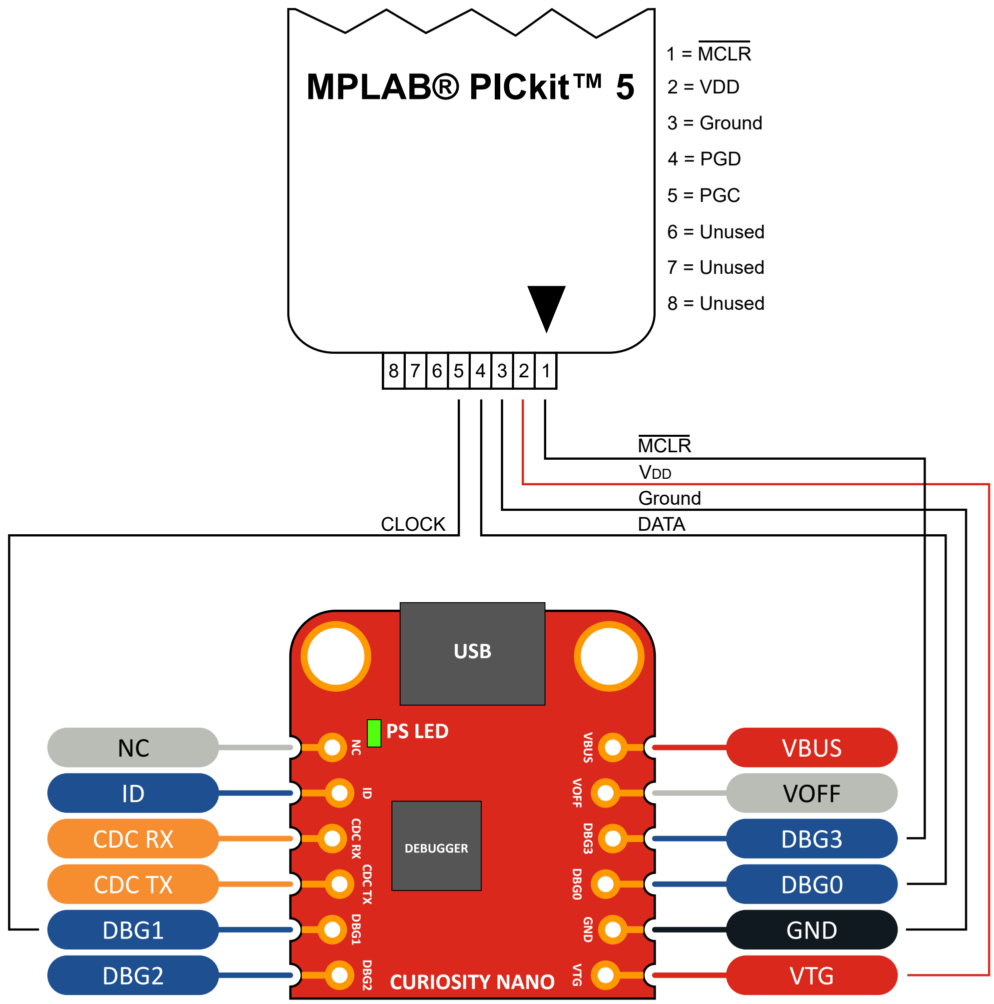

**Table 3-4.** Programming and Debugging Interfaces

| Curiosity Nano Pin | UPDI | ICSP™ | SWD |
|---|---|---|---|
| DBG0 | UPDI | DATA | SWDIO |
| DBG1 | — | CLK | SWCLK |
| DBG2 | — | — | — |
| DBG3 | — | $\overline{\text{MCLR}}$ | $\overline{\text{RESET}}$ |

## 3.5. Connecting External Debuggers

Even though there is an on-board debugger, external debuggers can be connected directly to the dsPIC33AK512MPS506 Curiosity Nano to program/debug the dsPIC33AK512MPS506. When not actively used, the on-board debugger keeps all the pins connected to the dsPIC33AK512MPS506 and board edge in tri-state. Therefore, the on-board debugger will not interfere with any external debug tools.

**Figure 3-11.** Connecting the MPLAB PICkit™ 5 In-Circuit Debugger/Programmer to dsPIC33AK512MPS506 Curiosity Nano




**CAUTION** The MPLAB PICkit 5 In-circuit Debugger/Programmer can deliver high voltage on the MCLR pin. High voltage can permanently damage R209. If R209 is broken, the on-board debugger cannot enter the Programming mode of the dsPIC33AK512MPS506 and will typically fail to read 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.

## 3.6. Disconnecting the On-Board Debugger

The on-board debugger and level shifters can be disconnected from the dsPIC33AK512MPS506.

The power supply block diagram (Figure 3-4) shows all connections between the debugger and the dsPIC33AK512MPS506. The signal names are printed in silkscreen on the top or bottom side of the board.

Cut the GPIO straps in Figure 3-9 to disconnect the debugger.

**Attention:** Cutting the GPIO straps to the on-board debugger prevents the virtual serial port, programming, debugging, and data streaming from functioning. Cutting the power supply strap disconnects 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 the holes. Connecting External Debuggers describes how to connect an external debugger.

# 4. Hardware Description

Introduction, feature list, Connectors/Pinout, Embedded Debugger Connections

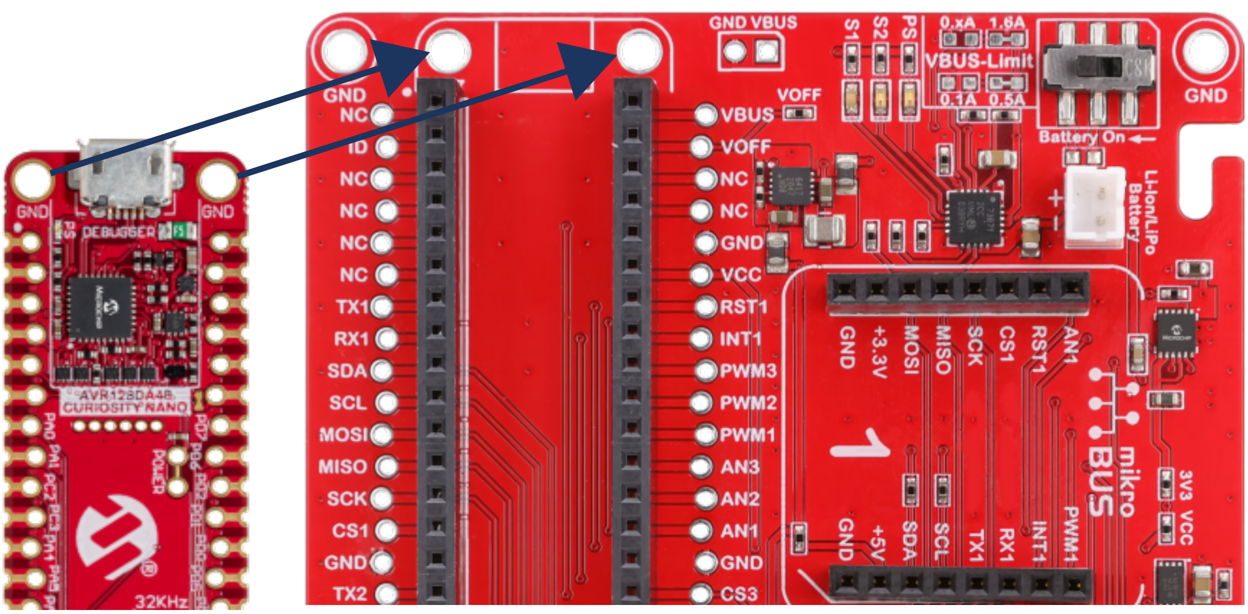
## 4.1. Connectors

### 4.1.1. Using Pin Headers

The edge connector footprint on the dsPIC33AK512MPS506 Curiosity Nano has a staggered design where each hole is shifted 8 mils (~0.2 mm) off-center. The hole shift allows using regular 100 mil pin headers without soldering on the board. The pin headers can be used in applications like pin sockets and prototyping boards without issues once they are firmly in place.

**Figure 4-1.** Attaching Pin Headers to the Curiosity Nano Board

**Figure 4-2.** 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
- For applications permanently using pin headers, it is still recommended to solder them in place
- 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

One yellow user LED is available on the dsPIC33AK512MPS506 Curiosity Nano board. Either GPIO or PWM can control it. Driving the connected I/O line to GND can also activate the LED.

**Figure 4-3.** dsPIC33AK512MPS506 Curiosity Nano LED0 Block Diagram

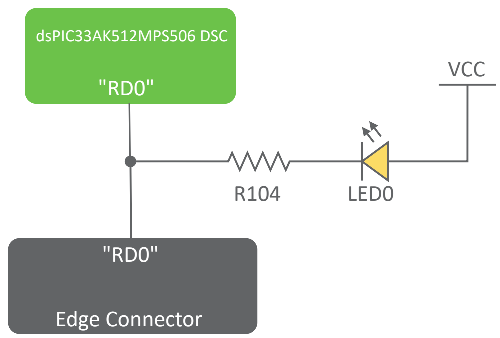


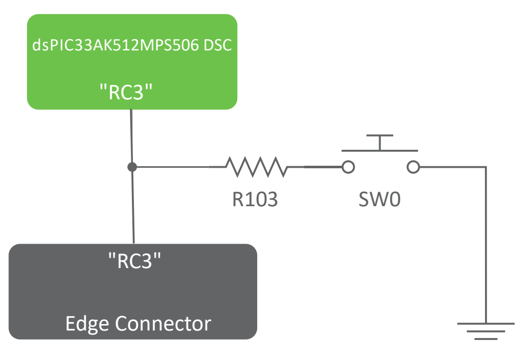

**Table 4-1.** LED Connection

| dsPIC33AK512MPS506 Pin | Function | Shared Functionality |
|---|---|---|
| RD0 | Yellow LED0 | Edge connector |

### 4.2.2. Mechanical Switch

The dsPIC33AK512MPS506 Curiosity Nano board has one mechanical switch—a generic user-configurable switch. Pressing it will connect the I/O pin to ground (GND).

**Figure 4-4.** dsPIC33AK512MPS506 Curiosity Nano SW0 Block Diagram



**Tip:** There is no externally connected pull-up resistor on the switch. Enable the internal pull-up resistor on Pin RC3 to use it.

**Table 4-2.** Mechanical Switch Connection

| dsPIC33AK512MPS506 Pin | Description | Shared Functionality |
|---|---|---|
| RC3 | User switch (SW0) | Edge connector, On-board debugger |

### 4.2.3. MEMs Oscillator

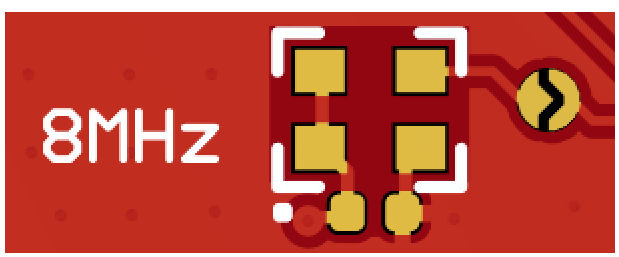
The dsPIC33AK512MPS506 Curiosity Nano is provisioned for an 8 MHz MEMs Oscillator. The MEMs is not connected to the dsPIC33AK512MPS506 by default because the GPIOs are routed out to the edge connector. To use the MEMs, some hardware modifications are required. The MEMs Oscillator Y1 (MCHP CLOCK OSCILLATOR SINGLE 8.000MHz DSC6011JI2B-008.0000), and the capacitor C8 (TDK C1005X7R1E104K050BB) or comparable parts, must be mounted on the board. The I/O lines routed to the edge connector should be disconnected to reduce the chance of contention to the MEMs and to remove excessive capacitance on the lines. This can be done by cutting the strap on the bottom side of the board, marked RC1 as shown in Figure 4-5. Next, add solder to the solder point next to the MEMs on the top side of the board as shown in Figure 4-6.

**Table 4-3.** MEMs I/O

| dsPIC33AK512MPS506 | Function | Shared Functionality |
|---|---|---|
| RC0 | OSCO | Edge connector |
| RC1 | OSCI | Edge connector |

**Figure 4-5.** RC1


**Figure 4-6.** Solder Point Location




### 4.2.4. On-Board Debugger Implementation

dsPIC33AK512MPS506 Curiosity Nano features an on-board debugger that can be used to program and debug the dsPIC33AK512MPS506 using UPDI / 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.

#### 4.2.4.1. On-Board Debugger Connections

All the connections between the target and the debugger are tri-stated when the debugger is not using the interface. Hence, there is little contamination of the signals, e.g., 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 On-Board Debugger.

**Table 4-4.** On-Board Debugger Connections

| dsPIC33AK512MPS506 Pin | Debugger Pin | Function | Shared Functionality |
|---|---|---|---|
| RC11 | CDC TX | UART RX (dsPIC33AK512MPS506 RX line) | Edge connector |
| RC10 | CDC RX | UART TX (dsPIC33AK512MPS506 TX line) | Edge connector |
| RC5 | DBG0 | PGD3 | Edge connector |
| RC2 | DBG1 | PGC3 | Edge connector |
| RC3 | DBG2 | SW0/GPIO | Edge connector |
| $\overline{MCLR}$ | 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 dsPIC33AK512MPS506 Curiosity Nano: The MPLAB® X IDE Kit Window or the sticker on the bottom of the PCB.

The Kit Window appears in MPLAB X IDE when connecting a Curiosity Nano development board to the computer. An additional page for the dsPIC33AK512MPS506 Curiosity Nano shows when the CNANO is plugged into the socket on the board.

The first nine 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. The data matrix code on the sticker contains a string with the product identifier 02-00236, revision, and serial number.

The string in the data matrix code has the following format:

```
"nnnnnnnrrssssssssss"
n = product identifier
r = revision
s = serial number
```

## 5.2. Revision 2

Revision 2 is the initially released version.

# 6. Document Revision History

**Revision A (November 2025)**

This is the initial version of the document.

# 7. Appendix

Schematic, Assembly Drawing, Adapter Pinout, Programming External MCUs, External Debuggers

## 7.1. Schematic

Figure 7-1. dsPIC33AK512MPS506 Curiosity Nano MCU Schematic

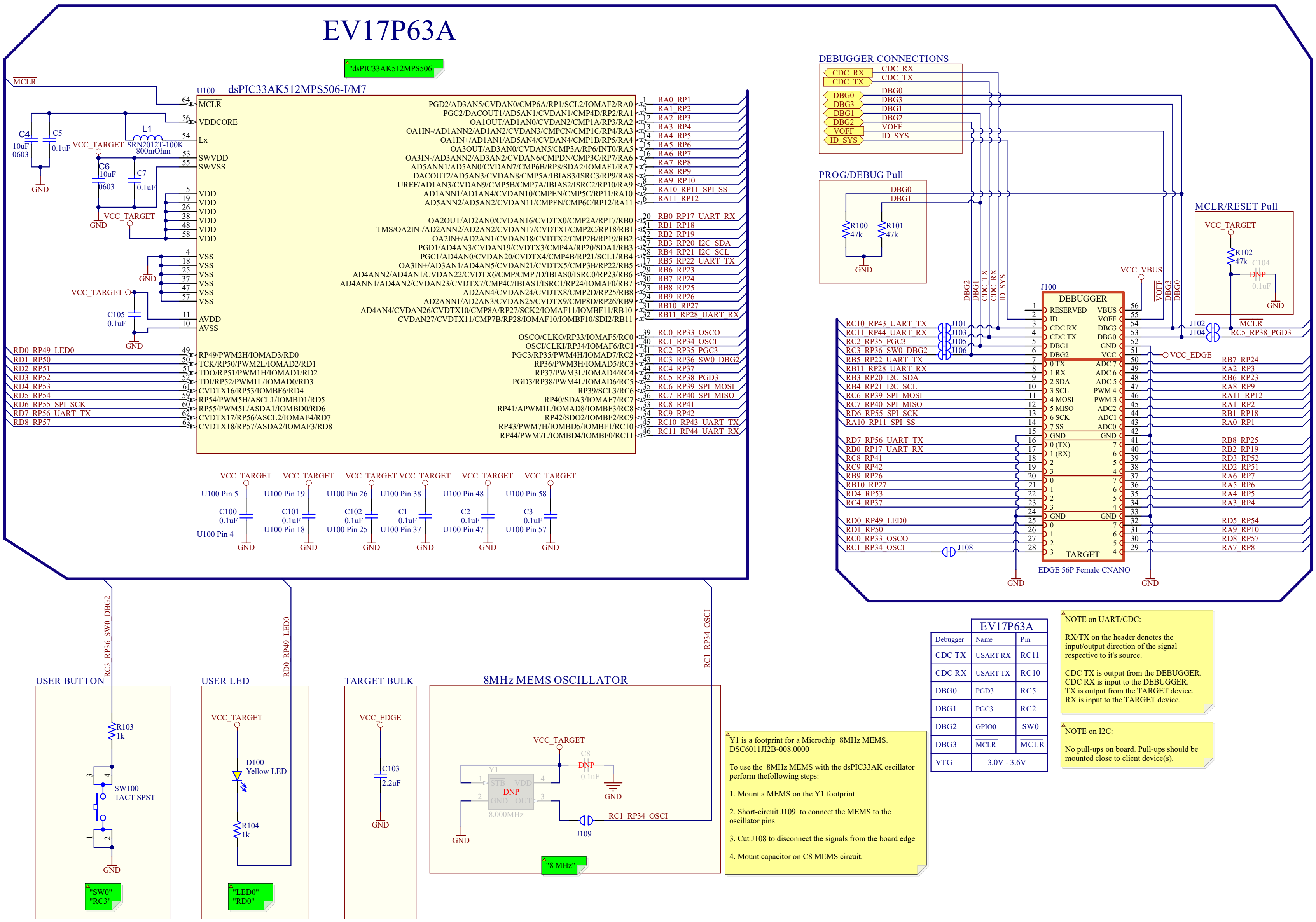
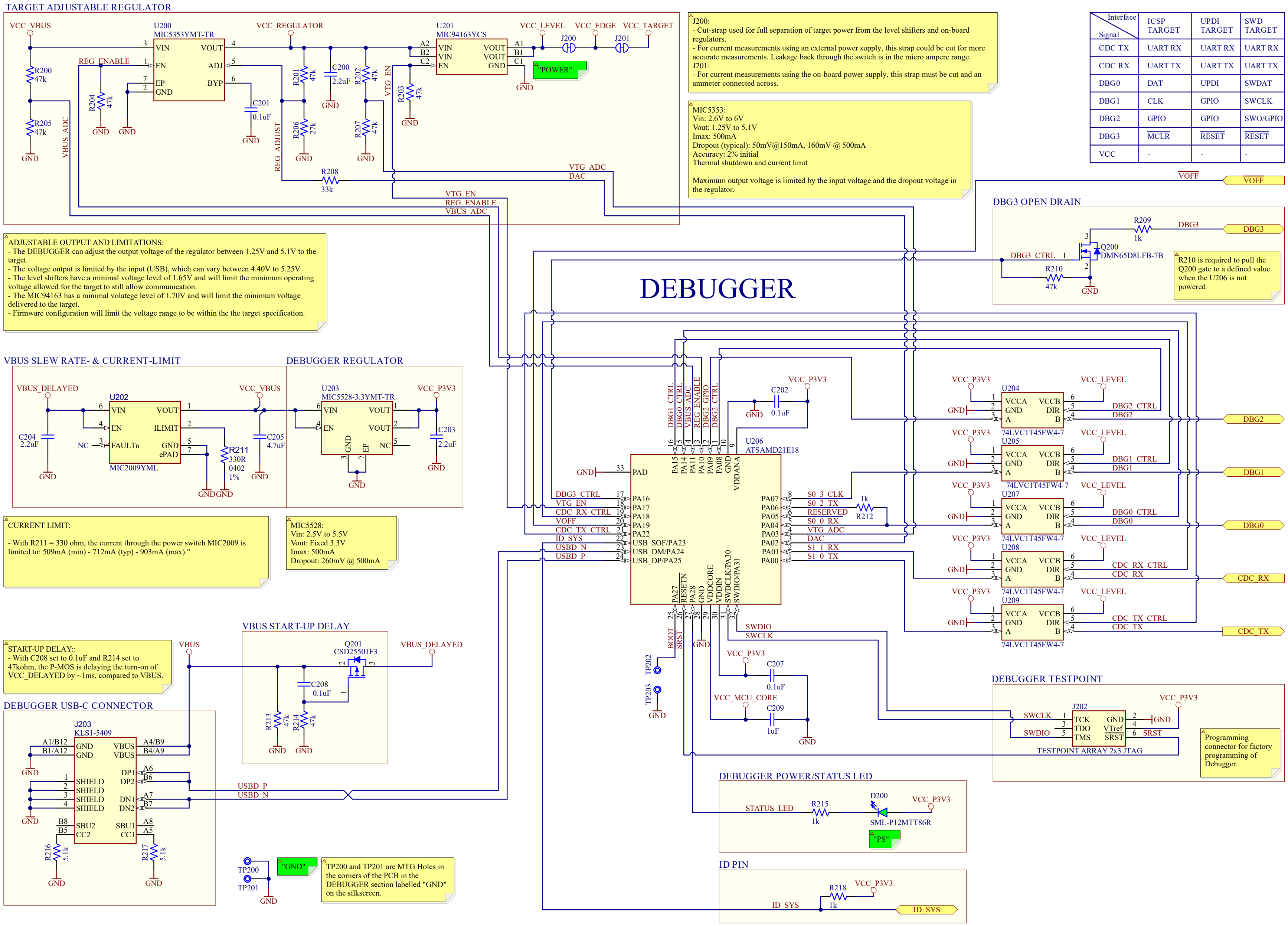


**Figure 7-2.** dsPIC33AK512MPS506 Curiosity Nano Debugger Schematic



## 7.2. Assembly Drawing

**Figure 7-3.** dsPIC33AK512MPS506 Curiosity Nano Assembly Drawing Top

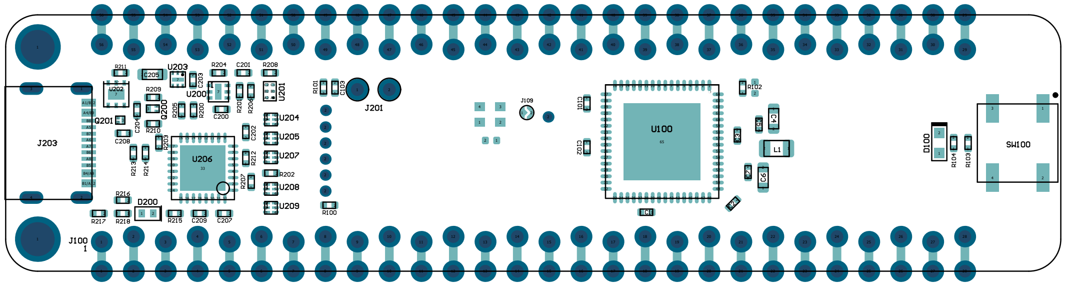
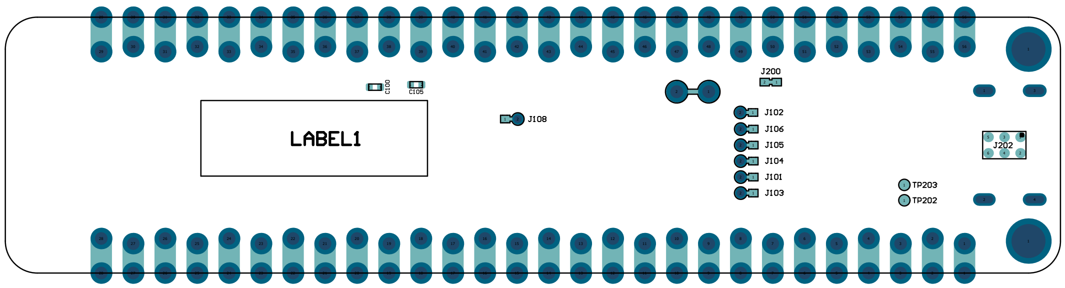
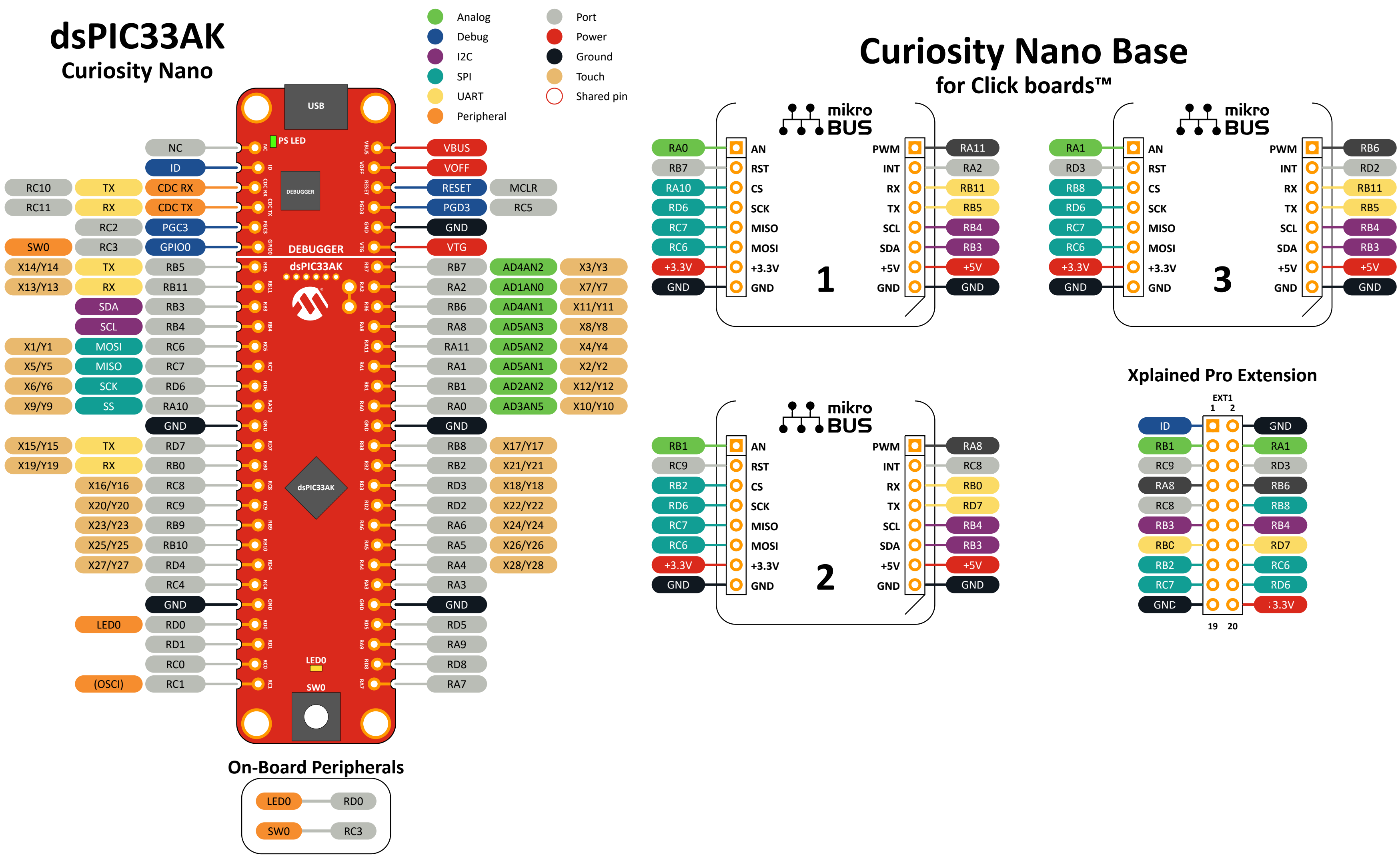



**Figure 7-4.** dsPIC33AK512MPS506 Curiosity Nano Assembly Drawing Bottom



## 7.3. Curiosity Nano Base for Click boards™

**Figure 7-5.** dsPIC33AK512MPS506 Curiosity Nano Pinout Mapping



# Microchip Information

## Trademarks



ISBN: 979-8-3371-2269-4

## Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications 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. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

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.

## Microchip Devices Code Protection Feature

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

- 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, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- 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. Microchip is committed to continuously improving the code protection features of our products.