AVR128DA48 Curiosity Nano

AVR128DA48 Curiosity Nano User Guide

Preface

The AVR128DA48 Curiosity Nano Evaluation Kit is a hardware platform to evaluate microcontrollers in the AVR-DA family. This board has the AVR128DA48 microcontroller (MCU) mounted.

Supported by Atmel Studio and Microchip MPLAB® X Integrated Development Environments (IDEs), the board provides easy access to the features of the AVR128DA48 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 AVR128DA48.
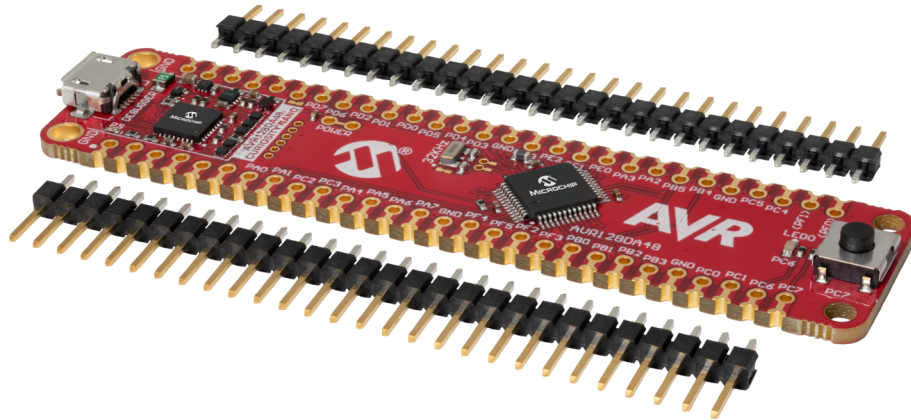
- **MPLAB® X IDE** and **Atmel Studio** - Software to discover, configure, develop, program, and debug Microchip microcontrollers.
- **Code examples in Atmel START** - Get started with code examples or generate drivers for a custom application.
- **Code examples on GitHub** - Get started with code examples.
- **AVR128DA48 website** - Find documentation, datasheets, sample, and purchase microcontrollers.
- **AVR128DA48 Curiosity Nano website** - Kit information, latest user guide and design documentation.

Table of Contents

1. Introduction

1.1 Features

- AVR128DA48-I/PT Microcontroller
- One Yellow User LED
- One Mechanical User Switch
- One 32.768 kHz Crystal
- On-Board Debugger:
    - Board identification in Atmel Studio/Microchip MPLAB® X IDE
    - One green power and status LED
    - Programming and debugging
    - Virtual serial port (CDC)
    - Two debug GPIO channels (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 Kit Overview

The Microchip AVR128DA48 Curiosity Nano Evaluation Kit is a hardware platform to evaluate the AVR128DA48 microcontroller.

Figure 1-1. AVR128DA48 Curiosity Nano Evaluation Kit Overview
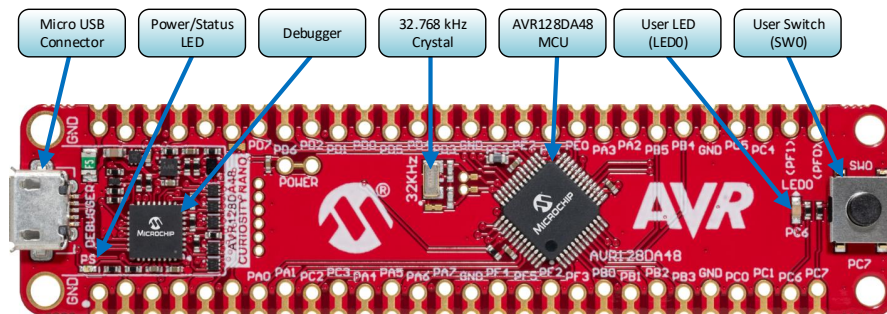
2. Getting Started

2.1 Quick Start

Steps to start exploring the AVR128DA48 Curiosity Nano Board:

1. Download Atmel Studio/Microchip MPLAB® X IDE.
2. Launch Atmel Studio/Microchip MPLAB® X IDE.
3. Optional: Use MPLAB® Code Configurator or Atmel START to generate drivers and examples.
4. Write your application code.
5. Connect a USB cable (Standard-A to Micro-B or Micro-AB) between the PC and the debug USB port on the board.

Driver Installation

When the board is connected to your 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 Atmel Studio/Microchip MPLAB® X IDE.

Kit Window

Once the board is powered, the green status LED will be lit, and Atmel Studio/Microchip MPLAB® X IDE will auto-detect which boards are connected. Atmel Studio/Microchip MPLAB® X IDE will present relevant information like data sheets and board documentation. The AVR128DA48 device on the AVR128DA48 Curiosity Nano Board is programmed and debugged by the on-board debugger and, therefore, no external programmer or debugger tool is required.

> **Tip:** The Kit Window can be opened in MPLAB X IDE through the menu bar *Window > Kit Window*.

2.2 Design Documentation and Relevant Links

The following list contains links to the most relevant documents and software for the AVR128DA48 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 called an Integrated Development Environment (IDE) because it provides a single integrated "environment" to develop code for embedded microcontrollers.
- **Atmel Studio** - Free IDE for the development of C/C++ and assembler code for microcontrollers.
- **IAR Embedded Workbench® for AVR®** - This is a commercial C/C++ compiler that is available for AVR microcontrollers. There is a 30-day evaluation version as well as a 4 KB code-size-limited kick-start version available from their website.
- **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.
- **Atmel START** - Atmel START is an online tool that hosts code examples, helps the user to select and configure software components, and tailor your embedded application in a usable and optimized manner.
- **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.

- **Studio Data Visualizer** - Studio Data Visualizer is a program used for processing and visualizing data. The Data Visualizer can receive data from various sources such as serial ports, on-board debugger's Data Gateway Interface as found on Curiosity Nano and Xplained Pro boards, and power data from the Power Debugger.
- **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.
- **AVR128DA48 Curiosity Nano website** - Kit information, latest user guide and design documentation.
- **AVR128DA48 Curiosity Nano on microchipDIRECT** - Purchase this kit on microchipDIRECT.

3. Curiosity Nano

Curiosity Nano is an evaluation platform of small boards with access to most of the microcontrollers I/Os. The platform consists of a series of low pin count microcontroller (MCU) boards with on-board debuggers, which are integrated with Atmel Studio/Microchip MPLAB® X IDE. Each board is identified in the IDE. When plugged in, 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

AVR128DA48 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 AVR128DA48 in Atmel Studio/Microchip MPLAB® X IDE
- A mass storage device that allows drag-and-drop programming of the AVR128DA48
- A virtual serial port (CDC) that is connected to a Universal Asynchronous Receiver/Transmitter (UART) on the AVR128DA48, 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 AVR128DA48 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 AVR128DA48 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 AVR128DA48 using Atmel Studio/Microchip MPLAB® X IDE, as well as some third-party IDEs.

> **Remember:** Keep the debugger's firmware up-to-date. Firmware upgrades are done automatically when using Atmel Studio/Microchip 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
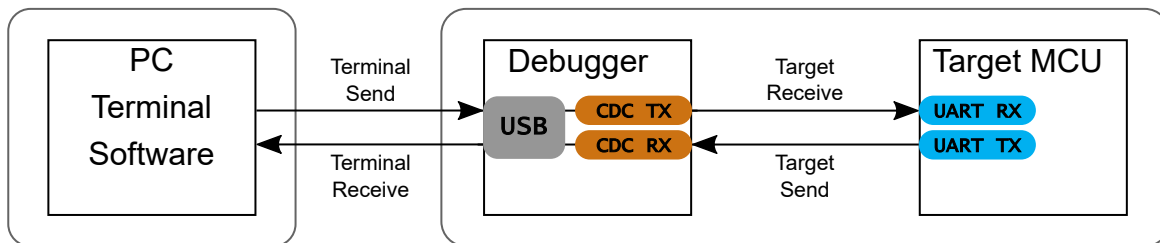
> **Info:** As shown in Figure 3-1, the debugger's CDC TX pin is connected to a UART RX pin on the target for receiving characters from the host computer. Similarly, the debugger's CDC RX pin is connected to a UART TX pin on the target for transmitting characters to the host computer.

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, a USB driver is required for CDC. This driver is included in installations of Atmel Studio/Microchip 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 on 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 the DTR signal will not disable the level shifters but disable the receiver so no further data will be streamed to the host. Data packets that are already queued up for sending to the target will continue to be sent out, but no further data will be accepted.

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 should 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=
```

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.

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, because 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 frame of data 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. 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 one 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, 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 application that has been previously 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 Fuse Bytes

Fuse Bytes (AVR® MCU Targets)
When doing drag-and-drop programming, the debugger masks out fuse bits that attempt to disable Unified Program and Debug Interface (UPDI). This means that the UPDI pin cannot be used in its reset or GPIO modes; selecting one of the alternative functions on the UPDI pin would render the device inaccessible without using an external debugger capable of high-voltage UPDI activation.

3.1.3.3 Limitations of Drag-and-Drop Programming

Lock Bits
Lock bits included in the hex file will be ignored when using drag-and-drop programming. To program lock bits, use Atmel Studio/Microchip MPLAB® X IDE.

Enabling CRC Check in Fuses
It is not advisable to enable the CRC check in the target device's fuses when using drag-and-drop programming. This because a subsequent chip erase (which does not affect fuse bits) will effect a CRC mismatch, and the application will fail to boot. To recover a target from this state, a chip erase must be done using Atmel Studio/Microchip MPLAB® X IDE, which will automatically clear the CRC fuses after erasing.

3.1.3.4 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
`CMD:SEND_UART=`	Sends a string of characters to the CDC UART. See "CDC Override Mode".
`CMD:RESET`	Resets the target device by entering Programming mode and then exiting Programming mode immediately thereafter. 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: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 commands listed here are triggered by the content being sent to the mass storage emulated disk, and no feedback is provided 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 Atmel Studio/Microchip MPLAB® X IDE.

Although DGI encompasses several physical data interfaces, the AVR128DA48 Curiosity Nano implementation includes logic analyzer channels:

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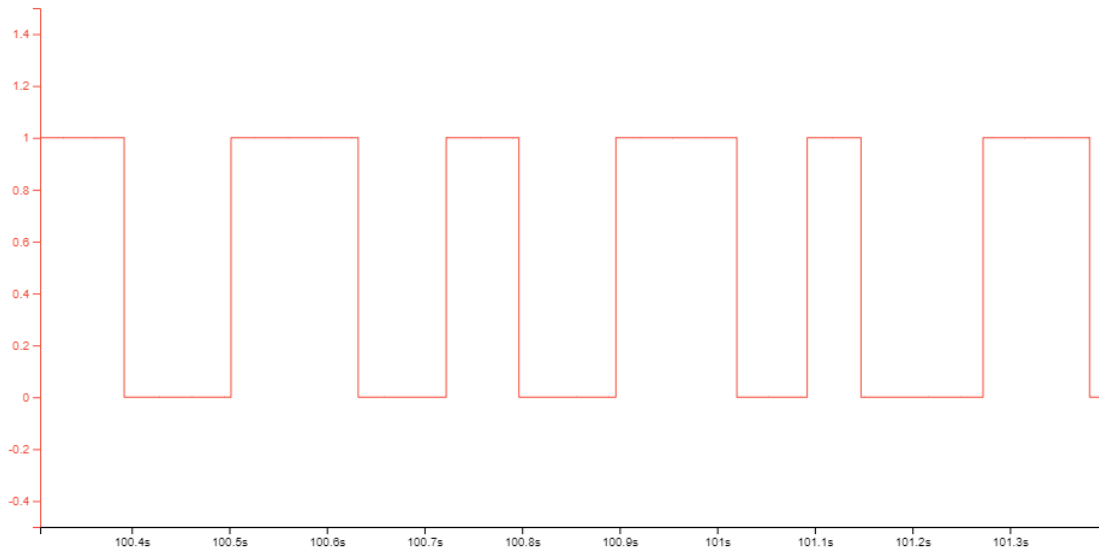
Debug GPIO channels are timestamped, so the resolution of DGI GPIO events is determined by the resolution of the DGI timestamp module.

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	UPDI	Debug data line
DBG1	GPIO1	debug GPIO1
DBG2	GPIO0	debug GPIO0
DBG3	RESET	Reset line
NC	—	No connect
VBUS	—	VBUS voltage for external use
$\overline{VOFF}$	—	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
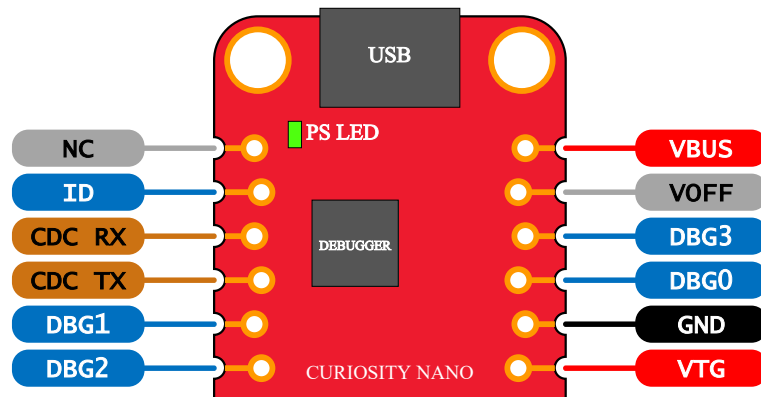
3.3 Power Supply

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 microcontroller AVR128DA48 and its peripherals. The voltage from the 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 AVR128DA48 Curiosity Nano.

Figure 3-4. Power Supply Block Diagram
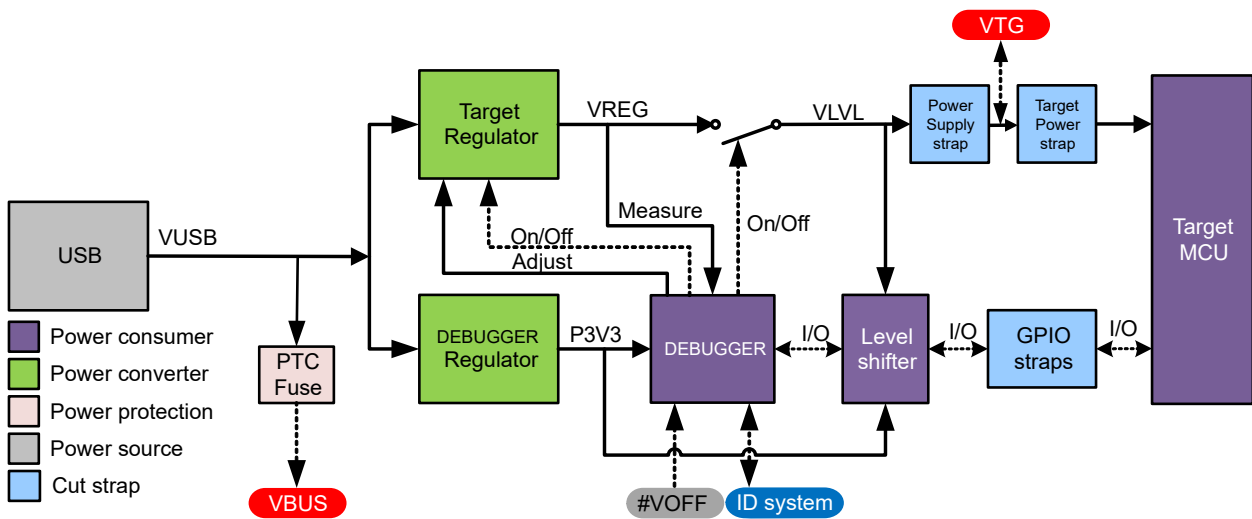
3.3.1 Target Regulator

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 AVR128DA48 microcontroller. The voltage limits configured in the on-board debugger on AVR128DA48 Curiosity Nano are 1.8-5.1V.

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

> **Info:** Voltage settings that are set up in Atmel Studio/Microchip 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. This only supports settings of 0.0V, 3.3V, and 5.0V. See section 3.1.3.4 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
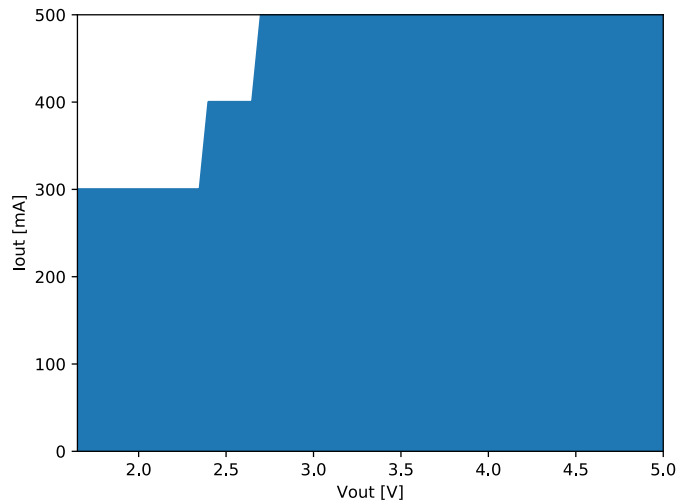
The voltage output of the target regulator is continuously monitored (measured) by the on-board debugger. If it is more than 100 mV over/under the voltage setting value, an error condition will be flagged, and the target voltage regulator will be turned off. This will detect and handle any short-circuit 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:** If the external voltage is lower than the monitoring window lower limit (target voltage setting - 100 mV), the on-board debugger status LED will blink rapidly. If the external voltage is higher than the monitoring window upper limit (target voltage setting + 100 mV), 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

AVR128DA48 Curiosity Nano can be powered by an external voltage instead of the on-board target regulator. 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.

> ⚠ **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** Absolute maximum external voltage is 5.5V for the on-board level shifters, and the standard operating condition of the AVR128DA48 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 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 is 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 AVR128DA48 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 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 µA. Typical values may be as low as 100 nA.

3.3.3 VBUS Output Pin

AVR128DA48 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
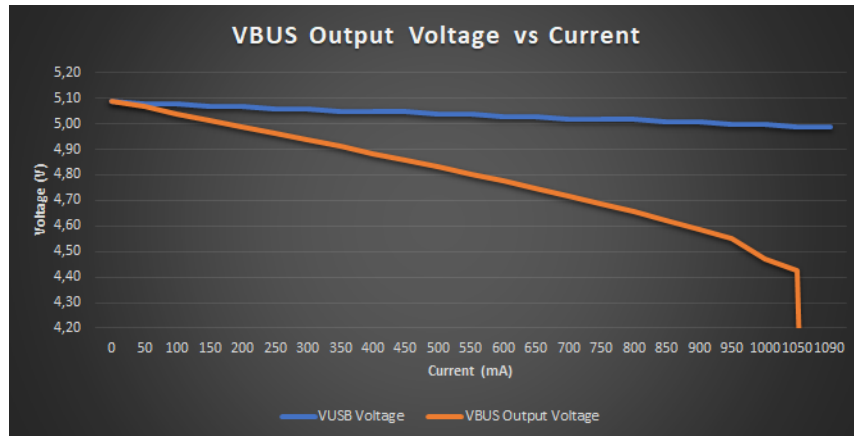
3.3.4 Power Supply Exceptions

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. This will 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 maximum output voltage is limited by the USB input voltage (specified to be between 4.4V to 5.25V), and 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 differ more than 100 mV over/under the voltage setting, it will be detected by the on-board debugger, and the internal voltage regulator will be 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 be lit normally 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 really 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.

VBUS Output Voltage is Low or Not Present

This is most lightly caused by a high-current drain on VBUS, 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 AVR128DA48 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 AVR128DA48 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 state.
5. Program the firmware into the AVR128DA48.

Figure 3-7. Target Power Strap
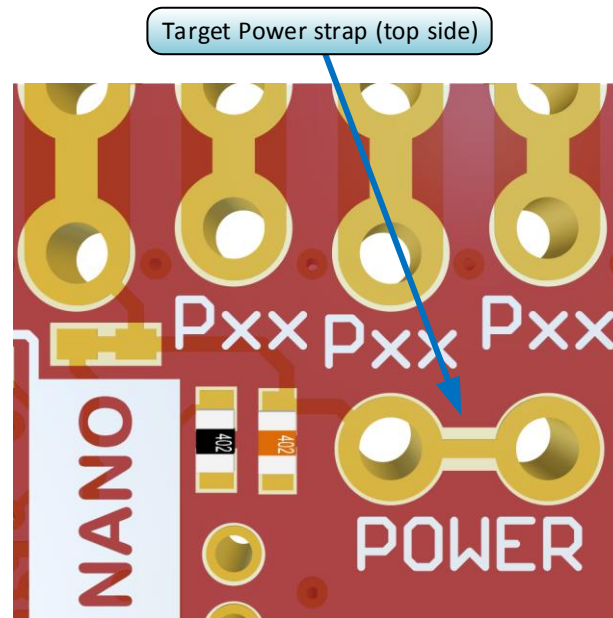
> **Tip:** A 100-mil pin header can be soldered into the *Target Power strap* (J101) footprint for easy connection of an ammeter. Once the ammeter is no longer needed, place a jumper cap on the pin header.

> **Info:** The on-board level shifters will draw a small amount of current even when they are not in use. A maximum of 2 µA can be drawn from each I/O pin connected to a level shifter for a total of 10 µA. Keep any I/O pin connected to a level shifter are tri-state to prevent leakage. All I/Os connected to the on-board debugger are listed in 4.2.4.1 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 AVR128DA48 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 Atmel Studio.

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

AVR128DA48 Curiosity Nano can program and debug external AVR128DA48 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, Atmel 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
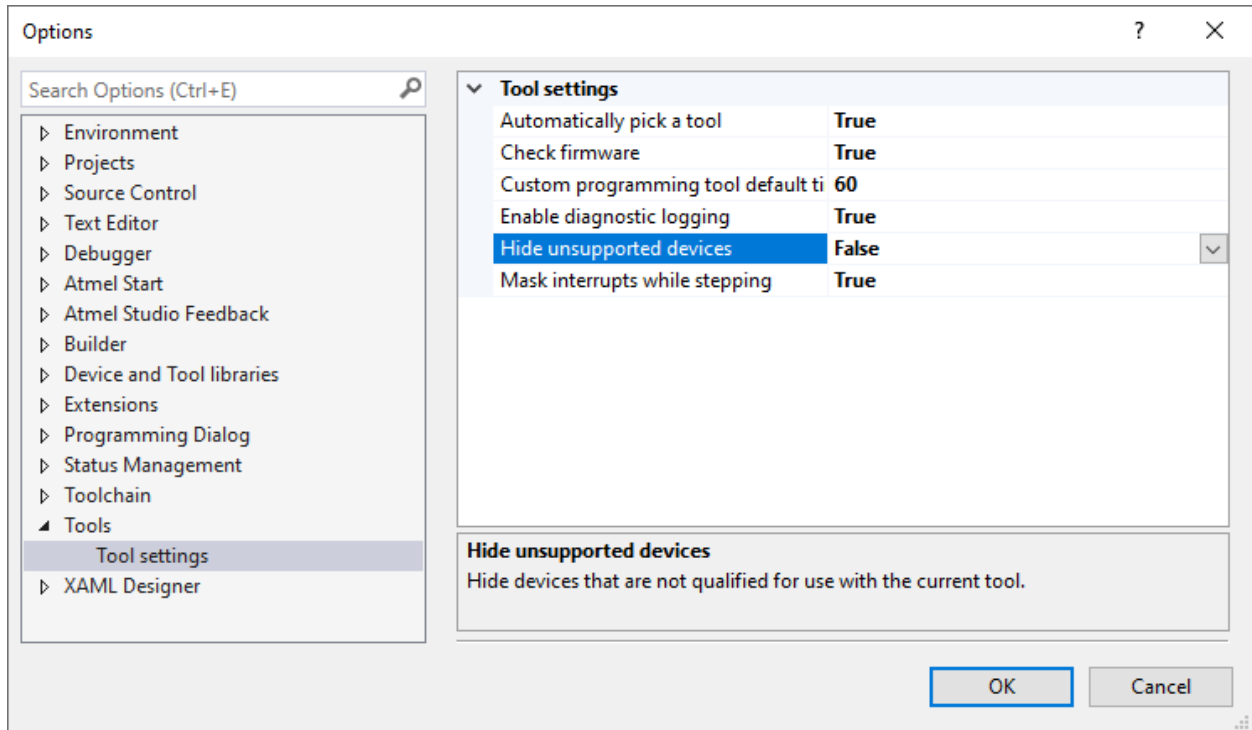
> **Info:** Atmel Studio allows any microcontroller and interface to be selected when **Hide unsupported devices** is set to **False**, also microcontrollers and interfaces which are not supported by the on-board debugger.

3.5.3   Hardware Modifications

The on-board debugger is connected to the AVR128DA48 by default. These connections must be removed 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 AVR128DA48 from the on-board debugger.

Figure 3-9. Programming and Debugging Connections to Debugger
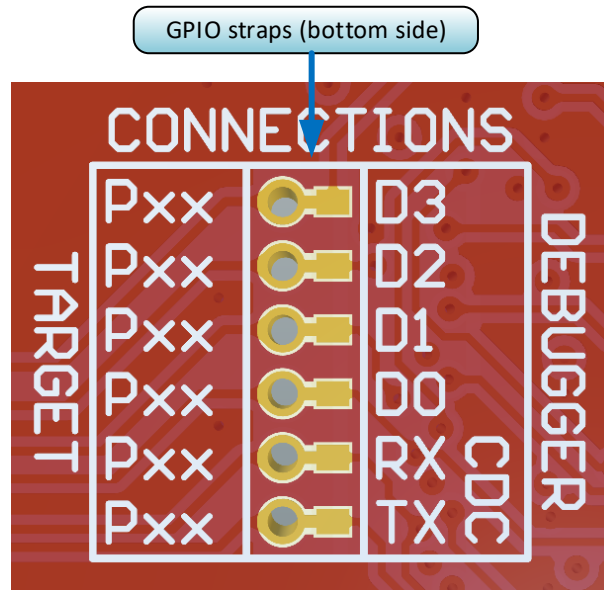
> **Info:** Cutting the connections to the debugger will disable programming, debugging, and data streaming from the AVR128DA48 mounted on the board.

> **Tip:** Solder in 0Ω resistors across the footprints or short-circuit them with solder to reconnect the signals between the on-board debugger and the AVR128DA48.

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 (DBG0, DBG1, and DBG2) used for programming and debugging, and in most cases, 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.

AVR128DA48 Curiosity Nano has a pull-up resistor R200 connected to its #RESET signal (DBG3). The location of the pull-up resistor 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 its own 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
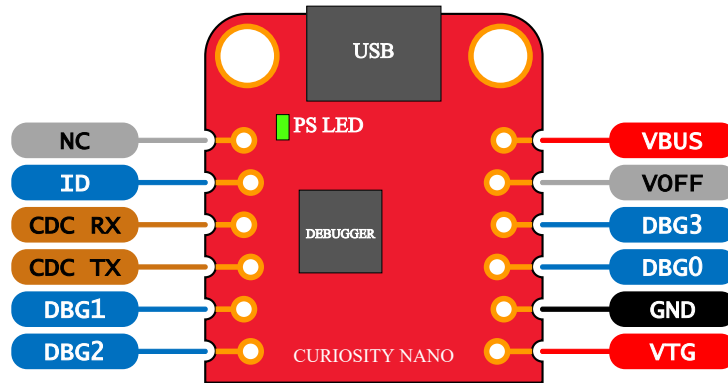
Table 3-4. Programming and Debugging Interfaces

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 AVR128DA48 Curiosity Nano to program/debug the AVR128DA48. The on-board debugger keeps all the pins connected to the AVR128DA48 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 AVR128DA48 Curiosity Nano
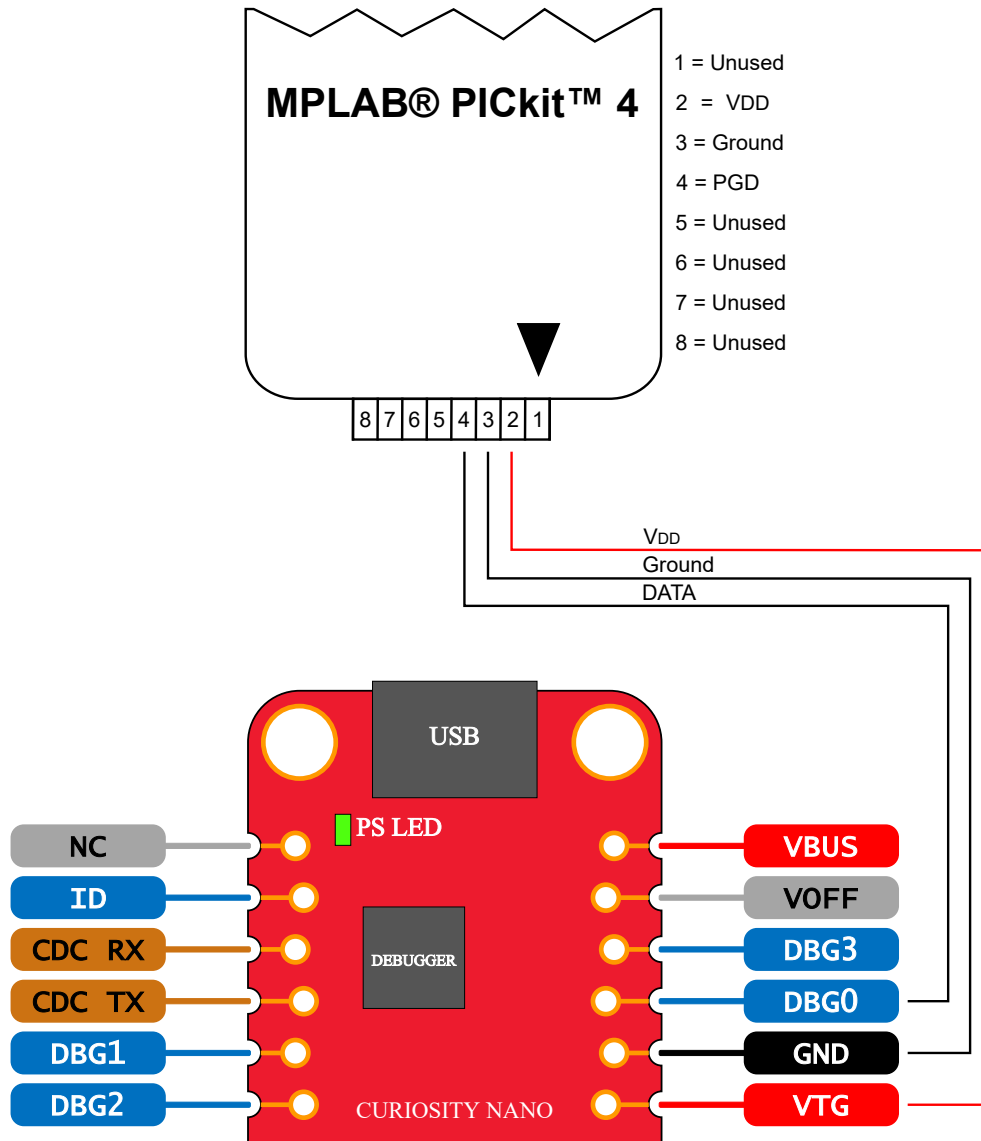
Figure 3-12. Connecting the Atmel-ICE to AVR128DA48 Curiosity Nano

1 = Unused   6 = Unused

2 = GND   7 = Unused

3 = UPDI   8 = Unused

4 = VTG   9 = Unused

5 = Unused   10 = Unused
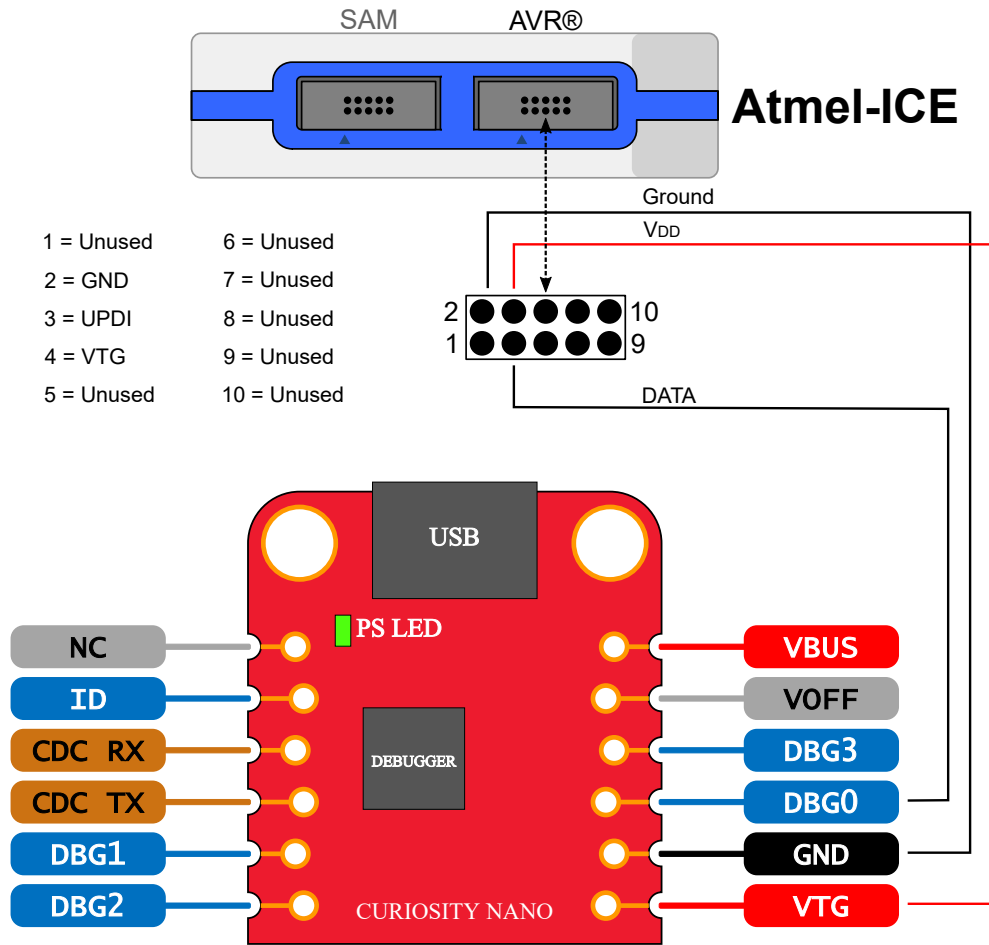
> **⚠ CAUTION** To avoid contention between the external debugger and the on-board debugger, do not start any programming/debug operation with the on-board debugger through Atmel Studio/Microchip MPLAB® X IDE or mass storage programming while the external tool is active.

4. Hardware User Guide

4.1 Connectors

4.1.1 AVR128DA48 Curiosity Nano Pinout

All the AVR128DA48 I/O pins are accessible at the edge connectors on the board. The image below shows the board pinout.

Figure 4-1. AVR128DA48 Curiosity Nano Pinout
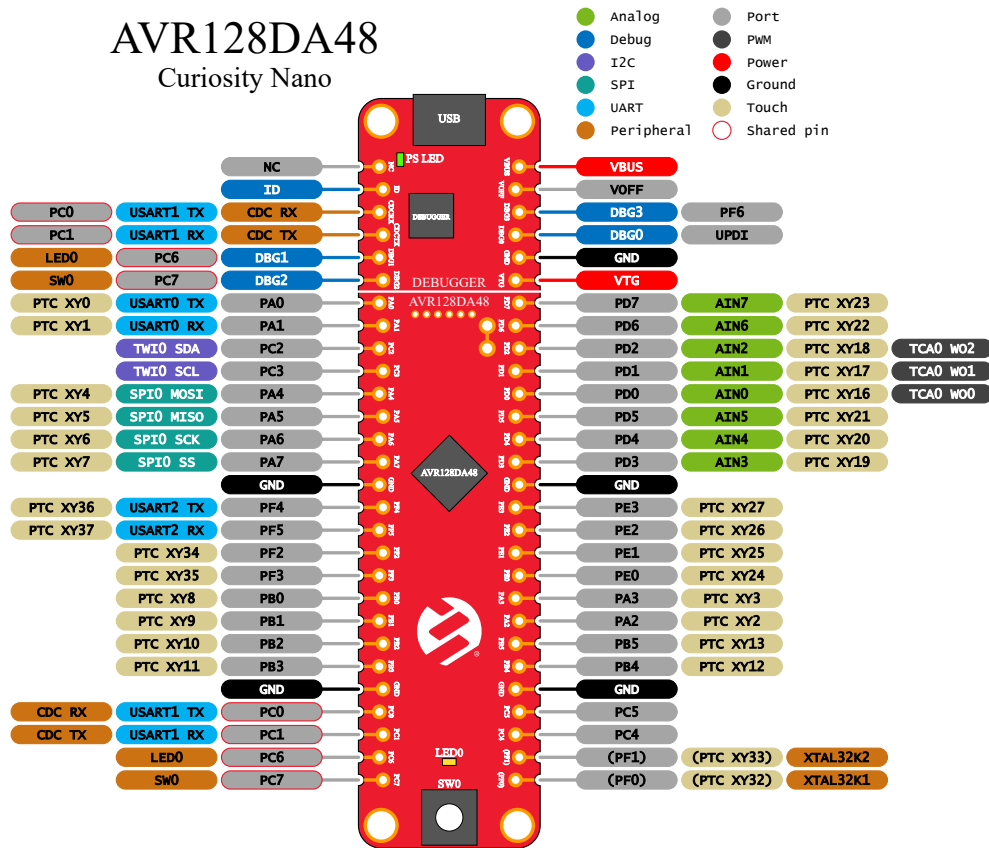
4.1.2 Using Pin Headers

The edge connector footprint on AVR128DA48 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.

> **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 AVR128DA48 Curiosity Nano Board that can be controlled by either GPIO or PWM. The LED can be activated by driving the connected I/O line to GND.

Table 4-1. LED Connection

AVR128DA48 Pin	Function	Shared Functionality
PC6	Yellow LED0	Edge connector, On-board debugger

4.2.2 Mechanical Switch

The AVR128DA48 Curiosity Nano has one mechanical switch. This is a generic user-configurable switch. When the switch is pressed, it will drive the I/O line to ground (GND).

> **Tip:** There is no externally connected pull-up resistor on the switch. To use the switch, make sure that an internal pull-up resistor is enabled on pin PC7.

Table 4-2. Mechanical Switch

AVR128DA48 Pin	Description	Shared Functionality
PC7	User switch (SW0)	Edge connector, On-board debugger

4.2.3 Crystal

The AVR128DA48 Curiosity Nano board has a 32.768 kHz crystal mounted.

The AVR128DA48 is connected to the crystal by default, but the GPIOs are also routed to the edge connector through two solder points. The two I/O lines routed to the edge connector are disconnected by default to reduce the chance of an external signal causing contention with the crystal, and to remove excessive capacitance on the lines. To use PF0 and PF1 as GPIO, some hardware modifications are required.

- Disconnect the crystal by cutting the two straps on the top side of the board next to the crystal (J210, J211). The crystal should be disconnected when using the pin as GPIO, as this might harm the crystal.
- Connect the I/O lines to the edge connector by placing solder blobs on the circular solder points marked PF0 and PF1 on the bottom side of the board (J207, J208)

The cut straps and solder points can be seen in Figure 4-2.

Table 4-3. Crystal Connections

AVR128DA48 Pin	Function	Shared Functionality
PF0	TOSC1 (Crystal input)	Edge connector
PF1	TOSC2 (Crystal output)	Edge connector

Figure 4-2. Crystal Connection and Cut Straps
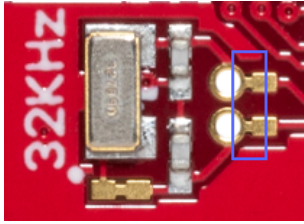
4.2.4 On-Board Debugger Implementation

AVR128DA48 Curiosity Nano features an on-board debugger that can be used to program and debug the AVR128DA48 using UPDI. The on-board debugger also includes a virtual serial port (CDC) interface over UART and debug GPIO. Atmel Studio/Microchip 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

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 4-4. On-Board Debugger Connections

AVR128DA48 Pin	Debugger Pin	Function	Shared Functionality
RF1	CDC TX	UART RX (AVR128DA48 RX line)	Edge connector
RF0	CDC RX	UART TX (AVR128DA48 TX line)	Edge connector
UPDI	DBG0	UPDI	Edge connector
PC6	DBG1	GPIO1	Edge connector, LED
PC7	DBG2	GPIO0	Edge connector, Mechanical Switch
PF6	DBG3	$\overline{\text{RESET}}$	Edge connector

5.   Hardware Revision History and Known Issues

This user guide is written to provide 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

The revision and product identifier of the AVR128DA48 Curiosity Nano Board can be found in two ways: Either by utilizing the Atmel Studio/Microchip MPLAB® X IDE Kit Window or by looking at the sticker on the bottom side of the PCB.

By connecting AVR128DA48 Curiosity Nano to a computer with Atmel Studio/Microchip MPLAB® X IDE running, the Kit Window will pop up. The first six digits of the serial number, which is listed under kit information, contain the product identifier and revision.

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

The same information can be 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 AVR128DA48 Curiosity Nano is A09-3280.

5.2   Revision 3

Revision 3 is the initially released revision. There are no known issues with this revision.

6. Document Revision History

Doc. rev.	Date	Comment
A	03/2020	Initial document release.

7. Appendix

7.1 Schematic

Figure 7-1. AVR128DA48 Curiosity Nano Schematic

TARGET ADJUSTABLE REGULATOR

DEBUGGER REGULATOR

DEBUGGER POWER/STATUS LED

DEBUGGER USB MICRO-B CONNECTOR

ID PIN

DEBUGGER TESTPOINT

STATUS LED

DBG3 OPEN DRAIN

CDC TX CTRL

CDC RX CTRL

DBG0 CTRL

DBG1 CTRL

DBG2 CTRL

VTG ADC

DAC

Interface table:

Interface Signal	ICSP TARGET	UPDI TARGET
CDC TX	UART RX	UART RX
CDC RX	UART TX	UART TX
DBG0	DAT	UPDI
DBG1	CLK	GPIO
DBG2	GPIO	GPIO
DBG3	MCLR	RESET
VCC	VCC	-

MIC5528:
Vin: 2.5V to 5.5V
Vout: Fixed 3.3V
Imax: 500mA
Dropout: 260mV @ 500mA

MIC5353:
Vin: 2.6V to 6V
Vout: 1.25V to 5.1V
Imax: 500mA
Dropout (typical): 50mV@150mA, 160mV @ 500mA
Accuracy: 2% initial
Thermal shutdown and current limit
Maximum output voltage is limited by the input voltage and the dropout voltage in the regulator.
(Vmax= Vin - dropout)

J100:
Cut-strap used for full separation of target power from the level shifters and on-board regulators.
- For current measurements using an external power supply, this strap could be cut for more accurate measurements. Leakage back through the switch is in the micro ampere range.

J101:
This is footprint for a 1x2 100mil pitch pin-header that can be used for easy current measurement to the target microcontroller and the LED / Button. To use the footprint:
- Cut the track between the holes, and mount a pin-header

Adjustable output and limitations:
- The DEBUGGER can adjust the output voltage of the regulator between 1.25V and 5.1V to the target.
- The voltage output is limited by the input (USB), which can vary between 4.40V to 5.25V
- The level shifters have a minimal voltage level of 1.65V and will limit the minimum operating voltage allowed for the target to still allow communication.
- The MIC94163 has a minimal voltage level of 1.70V and will limit the minimum voltage delivered to the target.
- Firmware configuration will limit the voltage range to be within the target specification.

R113 is required to pull the QI01 gate to a defined value when the U100 is not powered

Drawn By: PB
Engineer: AH, TF
Project Title: AVR128DA48 Curiosity Nano
Sheet Title: Debugger
Size: A3
PCB Assembly Number: A09-3280
PCB Number: A08-3002
PCBA Revision: 3
PCB Revision: 3
Page: 3 of 4
File: AVR128DA48 Curiosity Nano_Debugger.SchDoc

Designed with Altium
Altium.com
Date: 10/23/2019

MICROCHIP

7.2   Assembly Drawing

Figure 7-2. AVR128DA48 Curiosity Nano Assembly Drawing Top

Figure 7-3. AVR128DA48 Curiosity Nano Assembly Drawing Bottom

7.3 Curiosity Nano Base for Click boards™
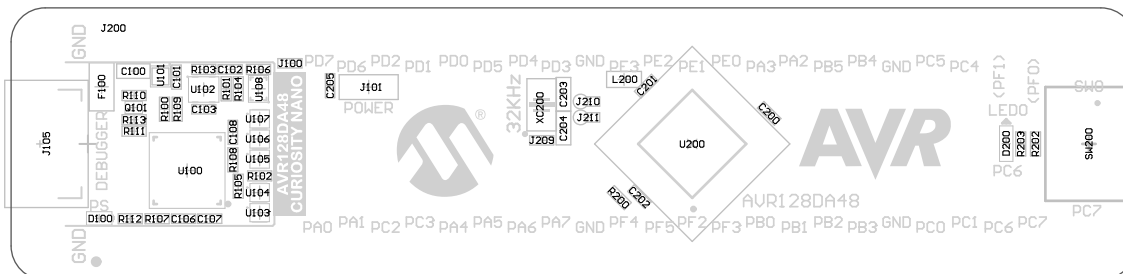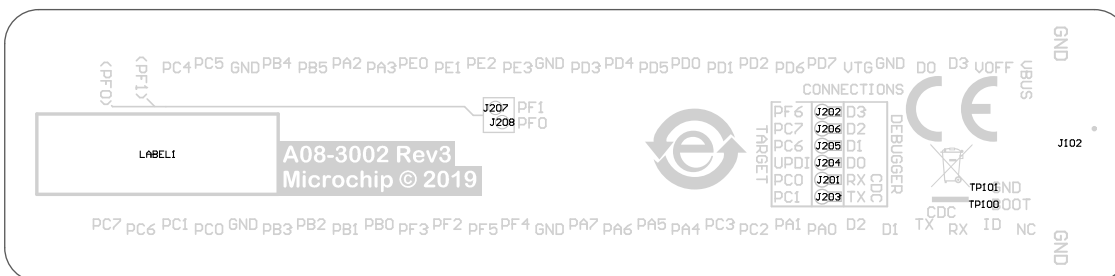
Figure 7-4. AVR128DA48 Curiosity Nano Pinout Mapping

7.4 Disconnecting the On-board Debugger

The on-board debugger and level shifters can be completely disconnected from the AVR128DA48.

The block diagram below shows all connections between the debugger and the AVR128DA48. 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:** Any connection that is cut can be reconnected using solder, alternatively, a 0Ω 0402 resistor can be mounted.

> **Tip:** When the debugger is disconnected, an external debugger can be connected to holes shown in Figure 7-6. Details about connecting an external debugger are described in 3.6 Connecting External Debuggers.

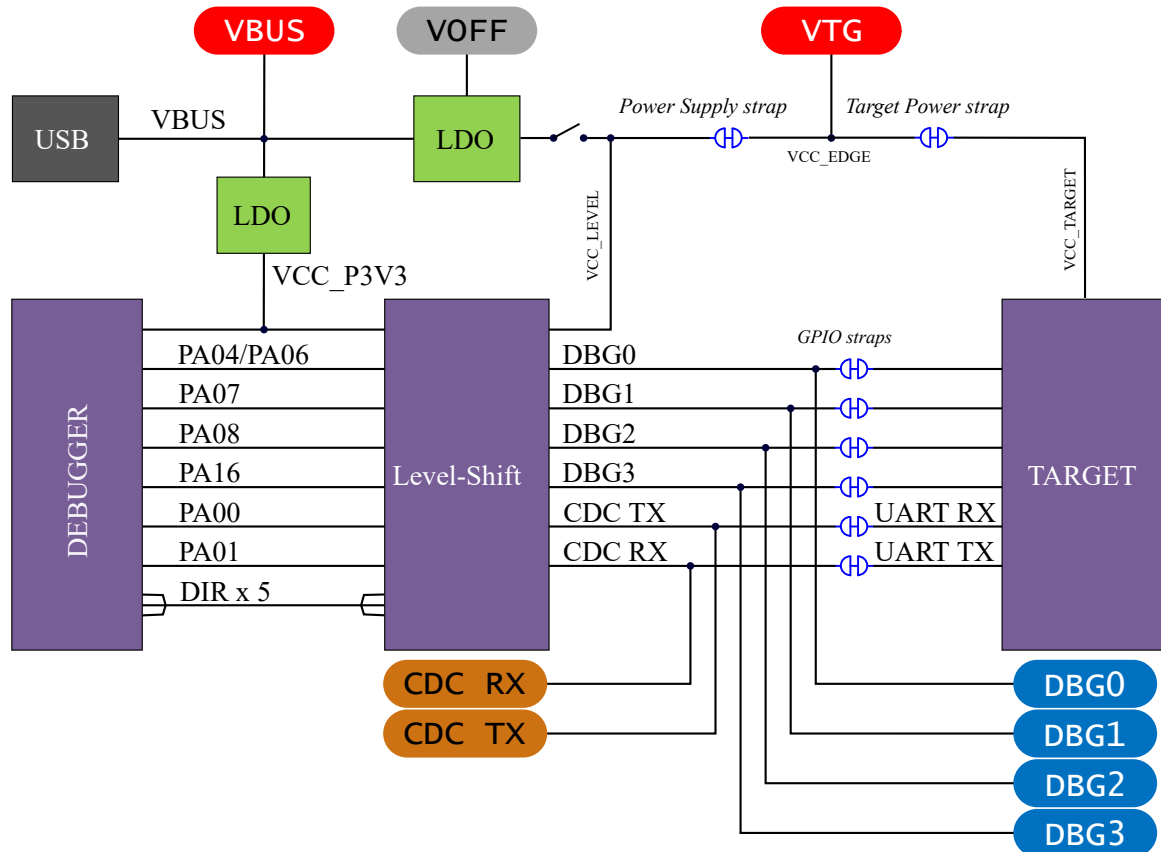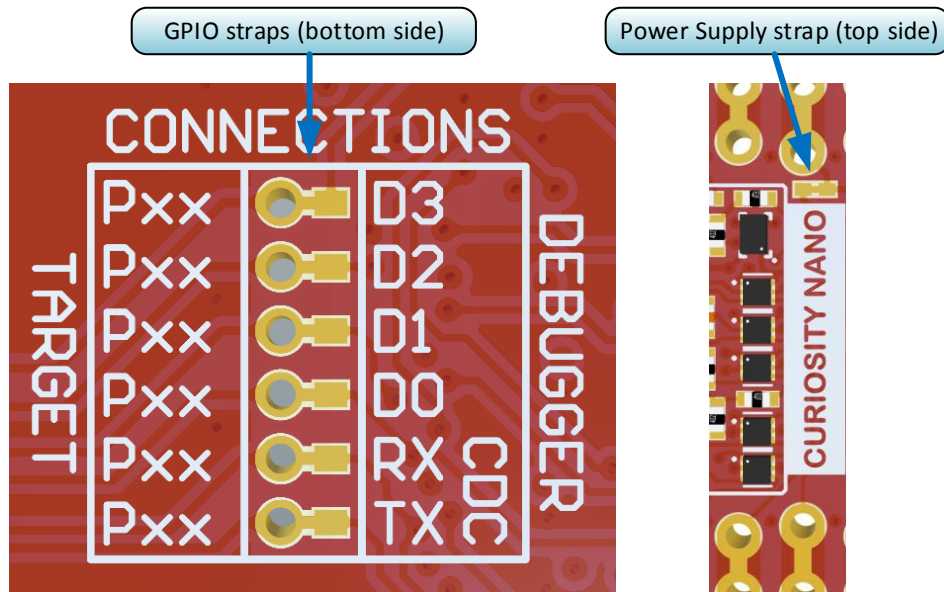**Figure 7-5. On-Board Debugger Connections Block Diagram**

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

7.5 Getting Started with IAR

IAR Embedded Workbench® for AVR® is a proprietary high-efficiency compiler not based on GCC. Programming and debugging of AVR128DA48 Curiosity Nano is supported in IAR™ Embedded Workbench for AVR using the Atmel-ICE interface. Some initial settings must be set up in the project to get the programming and debugging to work.

The following steps will explain how to get your project ready for programming and debugging:

1. Make sure you have opened the project you want to configure. Open the **OPTIONS** dialog for the project.
2. In the category **General Options**, select the **Target** tab. Select the device for the project, or if not listed, the core of the device, as shown in Figure 7-7.
3. In the category **Debugger**, select the **Setup** tab. Select **Atmel-ICE** as the driver, as shown in Figure 7-8.
4. In the category *Debugger > Atmel-ICE*, select the **Atmel-ICE 1** tab. Select **UPDI** as the interface and, optionally, select the **UPDI** frequency, as shown in Figure 7-9.

> **Info:** If the selection of Debug Port (mentioned in step 4) is grayed out, the interface is preselected, and the user can skip this configuration step.
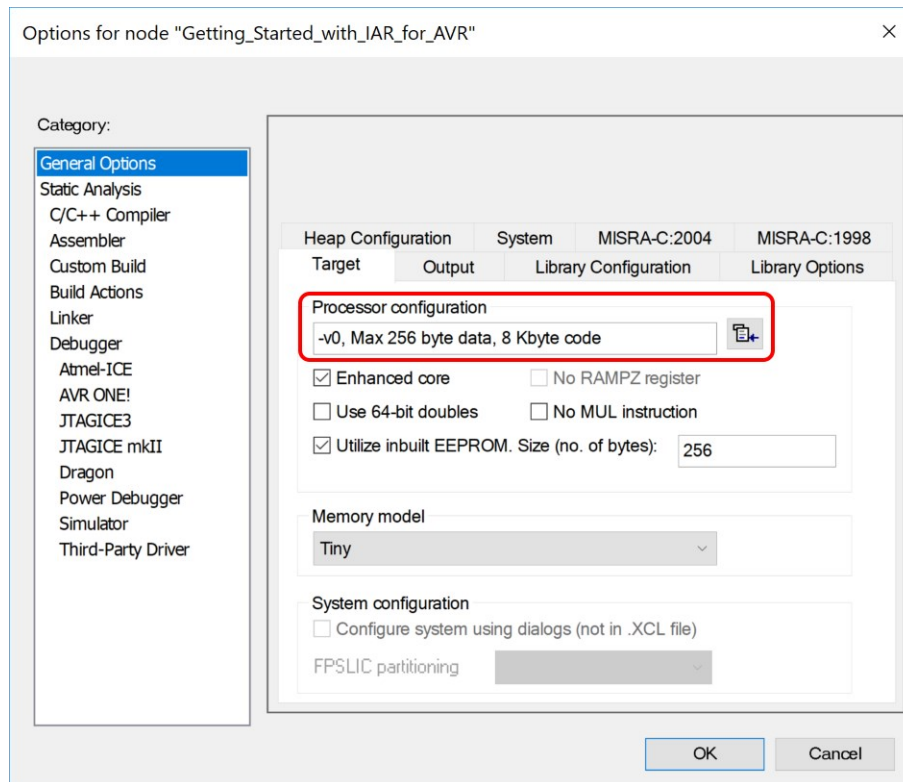
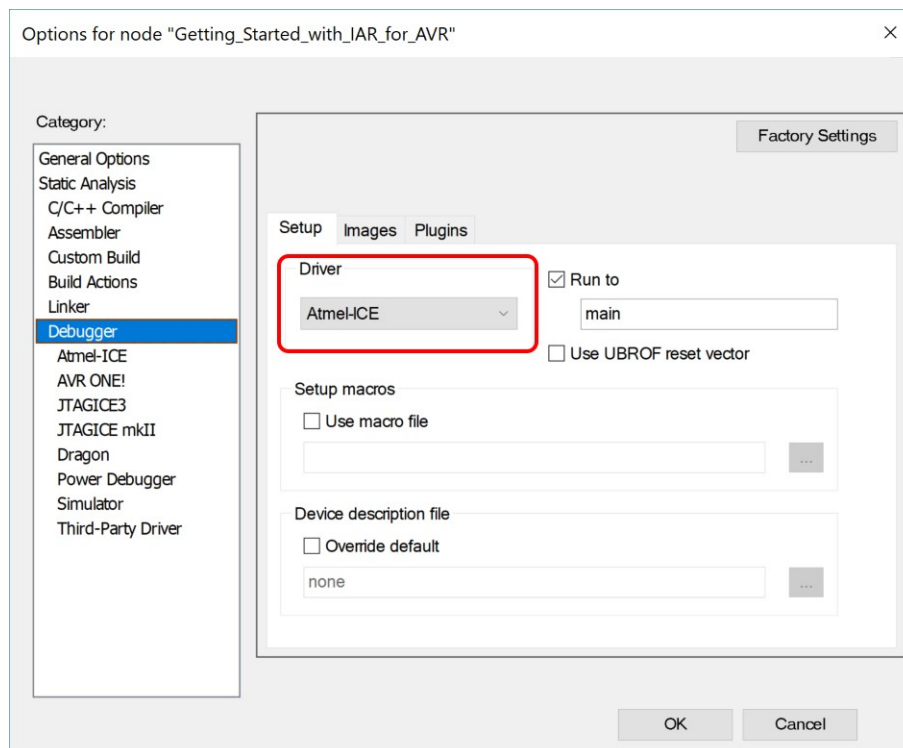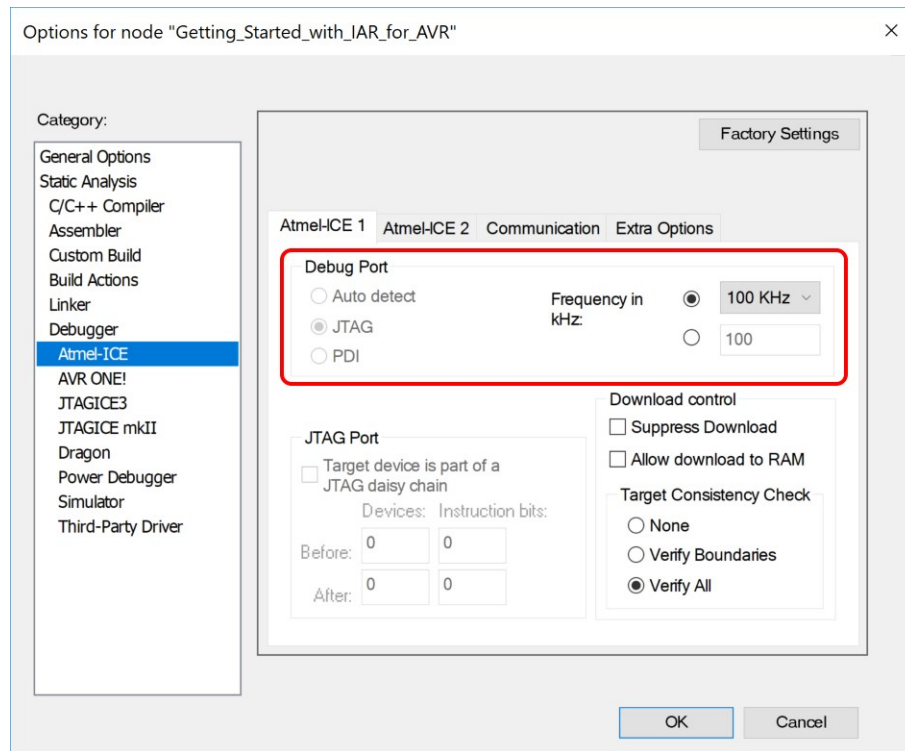**Figure 7-7. Select Target Device**

Figure 7-8. Select Debugger

Figure 7-9. Configure Interface

The Microchip Website

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

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

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with

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

Trademarks

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

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

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

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

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

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

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

© 2020, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-5760-2

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://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: http://www.microchip.com/support Web Address: http://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	**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	**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
Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455	**China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115	**Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200	**Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400
Austin, TX Tel: 512-257-3370	**China - Hong Kong SAR** Tel: 852-2943-5100	**Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906	**Germany - Heilbronn** Tel: 49-7131-72400
Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088	**China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355	**Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065	**Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44
Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075	**China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829	**Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366	**Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705
Dallas Addison, TX Tel: 972-818-7423 Fax: 972-818-2924	**China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526	**Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600	**Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781
Detroit Novi, MI Tel: 248-848-4000	**China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252	**Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340
Houston, TX Tel: 281-894-5983	**China - Xiamen** Tel: 86-592-2388138		**Norway - Trondheim** Tel: 47-72884388
Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380	**China - Zhuhai** Tel: 86-756-3210040		**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
Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800			**Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654
Raleigh, NC Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270			**UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820
Canada - Toronto Tel: 905-695-1980 Fax: 905-695-2078			