ATtiny1607 Curiosity Nano

ATtiny1607 Curiosity Nano Hardware User Guide

Preface

The ATtiny1607 Curiosity Nano evaluation kit is a hardware platform to evaluate the ATtiny1607 microcontroller.

Supported by Atmel Studio/Microchip MPLAB® X Integrated Development Environment (IDE), the kit provides easy access to the features of the ATtiny1607 to explore how to integrate the device into a custom design.

The Curiosity Nano series of evaluation kits include an on-board debugger. No external tools are necessary to program and debug the ATtiny1607.
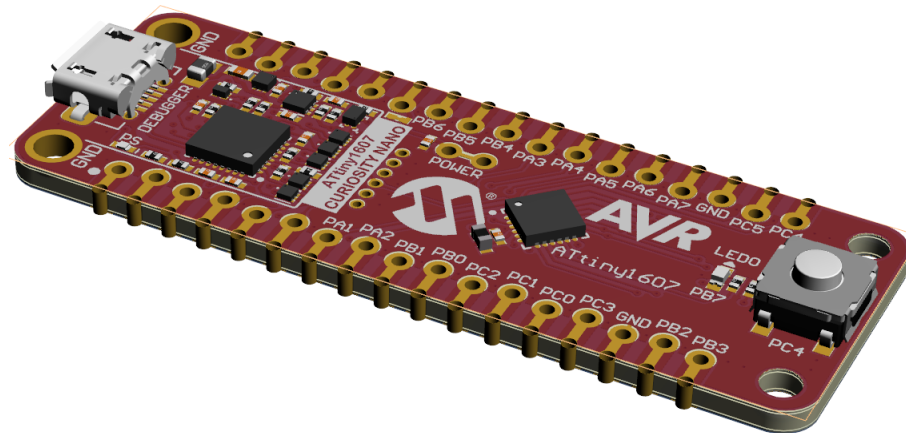
Table of Contents

1.   Introduction

1.1   Features

- ATtiny1607-MNR Microcontroller
- One Yellow User LED
- One Mechanical User Switch
- On-Board Debugger
  - Board identification in Atmel Studio/Microchip MPLAB® X
  - One green power and status LED
  - Programming and debugging
  - Virtual COM port (CDC)
  - Two logic analyzer 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 ATtiny1607 Curiosity Nano evaluation kit is a hardware platform to evaluate the ATtiny1607 microcontroller.

Figure 1-1. ATtiny1607 Curiosity Nano Evaluation Kit Overview
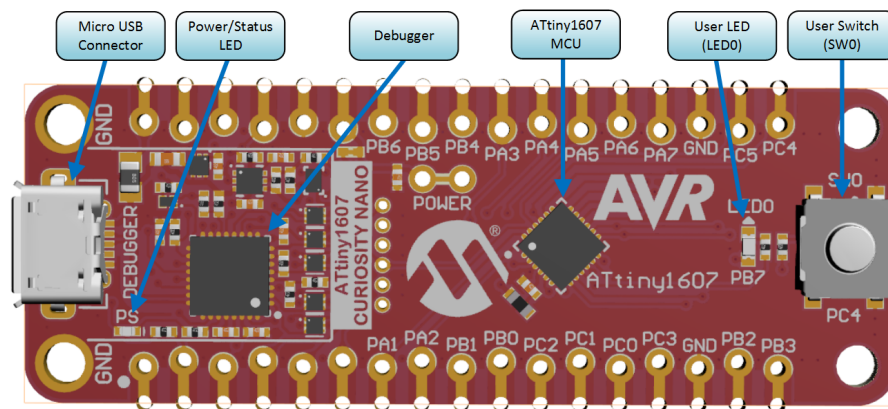
2. Getting Started

2.1 Curiosity Nano Quick Start

Steps to start exploring the Curiosity Nano platform:

1. Download Atmel Studio/Microchip MPLAB® X.
2. Launch Atmel Studio/Microchip MPLAB® X.
3. Connect a USB cable (Standard-A to Micro-B or Micro-AB) between the PC and the debug USB port on the kit.

When the Curiosity Nano kit 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 kit are included with Atmel Studio/Microchip MPLAB® X.

Once the Curiosity Nano board is powered the green status LED will be lit and Atmel Studio/Microchip MPLAB® X will auto-detect which Curiosity Nano board is connected. Atmel Studio/Microchip MPLAB® X will present relevant information like data sheets and kit documentation. The ATtiny1607 device is programmed by the on-board debugger and therefore no external programmer tool is required.

2.2 Design Documentation and Relevant Links

The following list contains links to the most relevant documents and software for the ATtiny1607 Curiosity Nano.

- **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.
- **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 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 8-bit AVR. There is a 30-day evaluation version as well as a 4 KB code-size-limited kick-start version available from their website.
- **Atmel START** - Atmel START is an online tool that 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.
- **Data Visualizer** - Data Visualizer is a program used for processing and visualizing data. The Data Visualizer can receive data from various sources such as the EDBG Data Gateway Interface found on Curiosity Nano and Xplained Pro boards and COM Ports.
- **ATtiny1607 Curiosity Nano website** - Kit information, latest user guide and design documentation.
- **ATtiny1607 Curiosity Nano on Microchip Direct** - Purchase this kit on Microchip Direct.

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. Each board is identified in the IDE, and relevant user guides, application notes, data sheets, and example code are easy to find. The on-board debugger features a Virtual COM port (CDC) for serial communication to a host PC, and a Data Gateway Interface (DGI) GPIO logic analyzer pin.

3.1 On-board Debugger

ATtiny1607 Curiosity Nano contains an on-board debugger for programming and debugging. The on-board debugger is a composite USB device of several interfaces: A debugger, a mass storage device, a data gateway, and a Virtual COM port (CDC).

Together with Atmel Studio/Microchip MPLAB® X, the on-board debugger can program and debug the ATtiny1607.

A Data Gateway Interface (DGI) is available for use with the logic analyzer channels for code instrumentation, to visualize the program flow. DGI GPIOs can be graphed using the Data Visualizer.

The Virtual COM port is connected to a UART on the ATtiny1607 and provides an easy way to communicate with the target application through terminal software.

The on-board debugger controls a Power and Status LED (marked PS) on ATtiny1607 Curiosity Nano. The table below shows how the LED is controlled in different operation modes.

Table 3-1. On-Board Debugger LED Control

Operation Mode	Status LED
Boot Loader mode	LED blink at 1 Hz during power-up.
Power-up	LED is ON.
Normal operation	LED is ON.
Programming	Activity indicator: The LED flashes slowly during programming/debugging.
Fault	The LED flashes fast if a power fault is detected.
Sleep/Off	LED is off. The on-board debugger is either in Sleep mode or powered down. This can occur if the kit is externally powered.

3.1.1 Virtual COM Port

The Virtual COM Port is a general purpose serial bridge between a host PC and a target device.

3.1.1.1 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 COM Port. The CDC can be used to stream arbitrary data in both directions between the host and the target: All characters sent from the host will be sent through a UART on the CDC TX pin, and UART characters sent into the CDC RX pin will be sent back to the host through the Virtual COM Port.

On Windows machines, the CDC will enumerate as *Curiosity Virtual COM Port* and appear in the Ports section of the device manager. The COM port number is shown here.

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

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:** On older Windows systems, a USB driver is required for CDC. This driver is included in MPLAB X and Atmel® Studio installations.

3.1.1.2 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 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.

3.1.1.3 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. This is a virtual control signal implemented on the USB interface, but not in hardware in the on-board debugger. Asserting DTR from the host will indicate to the on-board debugger that a CDC session is active, will 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:** Enable to set up your terminal emulator to assert the DTR signal. Without it, the on-board debugger will not send or receive any data through its UART.

3.1.1.4 Advanced Use

CDC Override Mode

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

Dropping a text file (with extension `.txt`) into the on-board debugger's mass storage drive can be used to send characters out of the CDC TX pin. 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 baud rate last used 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 CDC TX pin. Transferring a small amount of data per frame can be inefficient, particularly at low baud rates, since the on-board debugger buffers frames and not bytes. A maximum of 4 x 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.

When receiving data from the target, 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 8 x 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.

3.1.2 Mass Storage Disk

A simple way to program the target device is through drag and drop with `.hex` files.

3.1.2.1 Mass Storage Device

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

The CURIOSITY drive 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.

The on-board debugger enumerates as a Curiosity Nano USB device that can be found in the disk drives section of the Windows 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 a `.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 kit 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 kit firmware, name, serial number, and device.
- `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 not reflect the correct status.

3.1.2.2 Fuse Bytes

Fuse Bytes (AVR® MCU Targets)

The debugger does *not* mask any fuse bits or combinations when writing fuses. It is not possible to disable UPDI by fuse setting on devices with a dedicated UPDI pin. For devices with a shared/configurable UPDI pin, be sure **not** to select an alternate pin function for UPDI either by fuse setting in Programming mode or by using the I/O view or memory views to modify the memory-mapped fuse values. Disabling UPDI will render the debugger unable to contact the target device — an external programmer capable of 12V UPDI activation will be required.

3.2 Curiosity Nano Standard Pinout

The twelve edge connections closest to the USB connector on Curiosity Nano kits 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	UPDI Target	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 clock line/DGI GPIO.
DBG2	GPIO0	DGI GPIO.
DBG3	RESET	Reset line.
NC	-	No connect.
VBUS	-	VBUS voltage for external use.
VOFF	-	Voltage Off input.
VTG	-	Target voltage.
GND	-	Common ground.

Figure 3-1. Curiosity Nano Standard Pinout
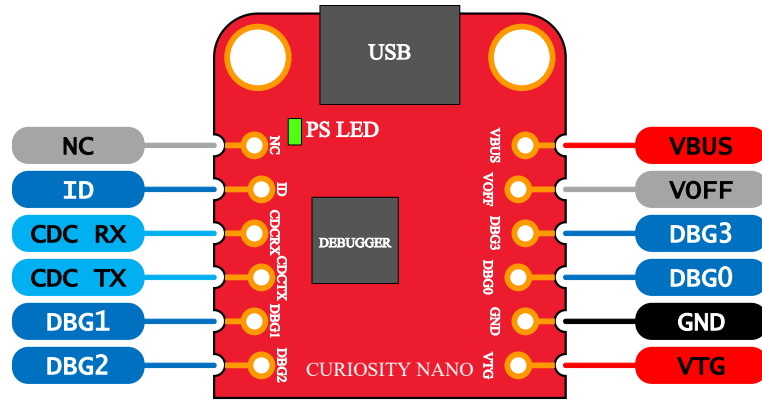
3.3 Power Supply

The kit 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 ATtiny1607 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 ATtiny1607 Curiosity Nano.

Figure 3-2. Power Supply Block Diagram
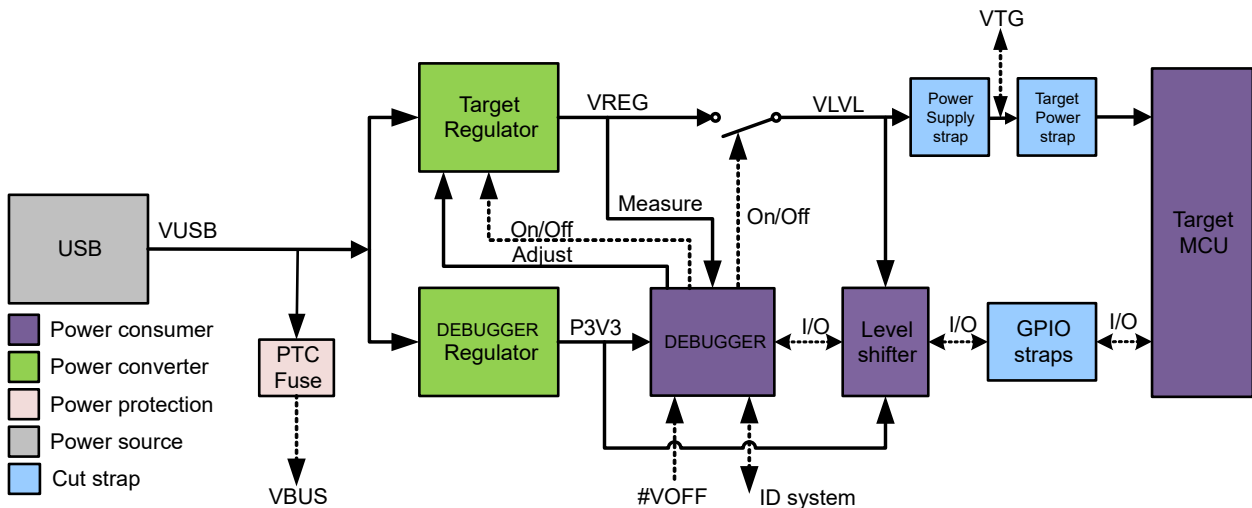
3.3.1 Target Regulator

The target voltage regulator is a MIC5353 variable output LDO. The on-board debugger can adjust the voltage output supplied to the kit 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 ATtiny1607 microcontroller. The voltage limits configured in the on-board debugger on ATtiny1607 Curiosity Nano are 1.8-5.1V.

> **Info:** The target voltage is set to 3.3V in production. It can be changed through MPLAB X project properties, and in the Atmel Studio device programming dialog. Any change to the target voltage is persistent, even through a power toggle.

The MIC5353 supports a maximum current load of 500 mA. It is an LDO regulator in a small package, placed on a small 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 3-3. Target Regulator Safe Operation Area
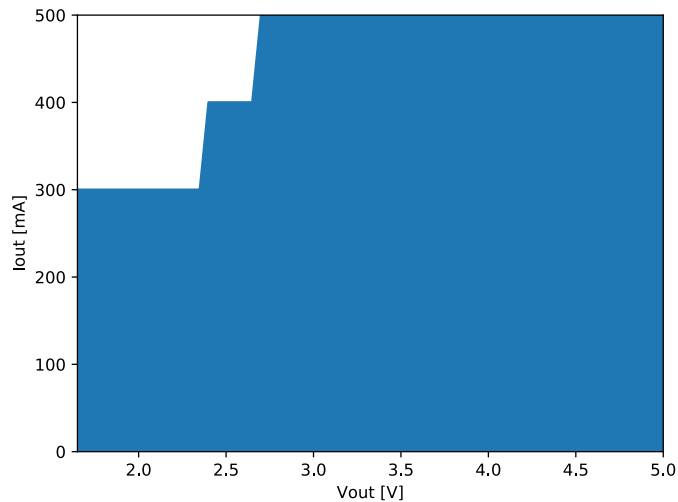
3.3.2 External Supply

ATtiny1607 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.

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

> ⚠ **WARNING** Absolute maximum external voltage is 5.5V for the on-board level shifters, and the standard operating condition of the ATtiny1607 is 1.8-5.5V. Applying a higher voltage may cause permanent damage to the kit.

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.

3.3.3 VBUS Output Pin

ATtiny1607 Curiosity Nano has a VBUS output pin which 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 3-4. VBUS Output Voltage vs. Current
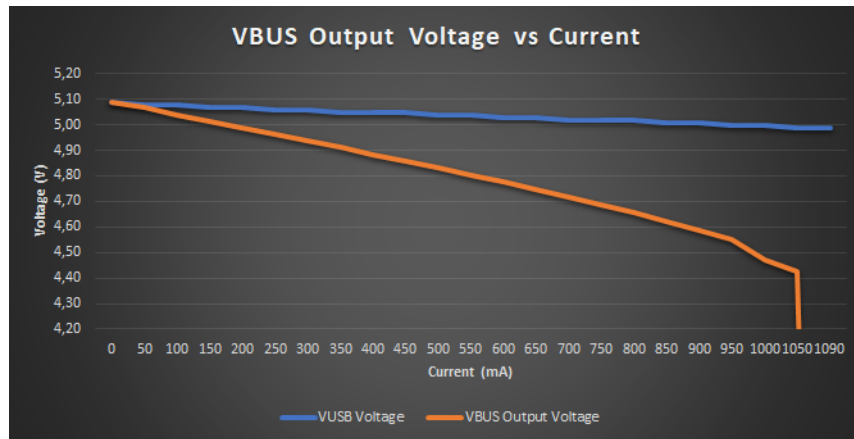
3.4 Target Current Measurement

Power to the ATtiny1607 is connected from the on-board power supply and *VTG* pin through a 100-mil pin header cut *Target Power strap* marked with "*POWER*" in silkscreen (J101). To measure the power consumption of the ATtiny1607 and other peripherals connected to the board, cut the *Target Power Strap* and connect an ammeter over the strap.

Figure 3-5. Target Power Strap
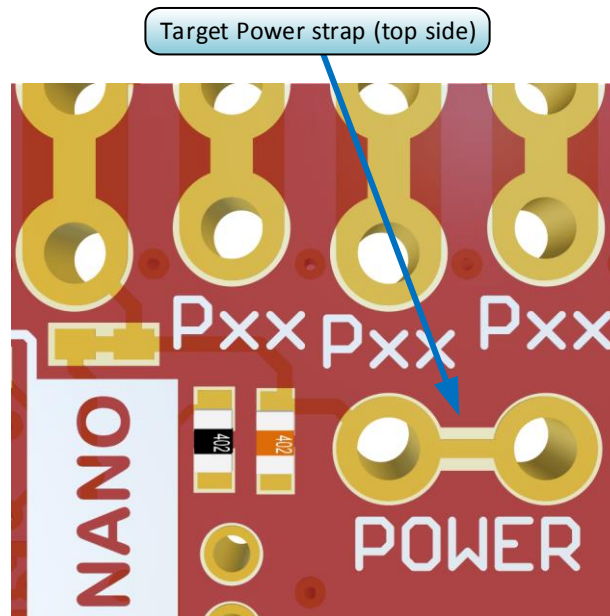
> **Tip:** A 100-mil pin header can be soldered into the *Target Power strap* (J101) footprint for easy connection of an ammeter. Once the ammeter is not needed anymore, place a jumper-cap on the pin header.

> **Info:** The on-board level shifters will draw a small amount of current even when they are not in use. A maximum of 10 µA can be drawn from the target power net, and an additional 2 µA can be drawn from each I/O pin connected to a level shifter for a total of 20 µA. Disconnect the on-board debugger and level shifters as described in Section 3.5 Disconnecting the On-Board Debugger and keep any I/O pin connected to a level shifter in tri-state to prevent leakage.

3.5   Disconnecting the On-Board Debugger

The block diagram below shows all connections between the debugger and the ATtiny1607 microcontroller. The rounded boxes represent connections to the board edge on ATtiny1607 Curiosity Nano. The signal names shown in Figure 3-1 are printed in silkscreen on the bottom side of the board.

Figure 3-6.   On-Board Debugger Connections to the ATtiny1607
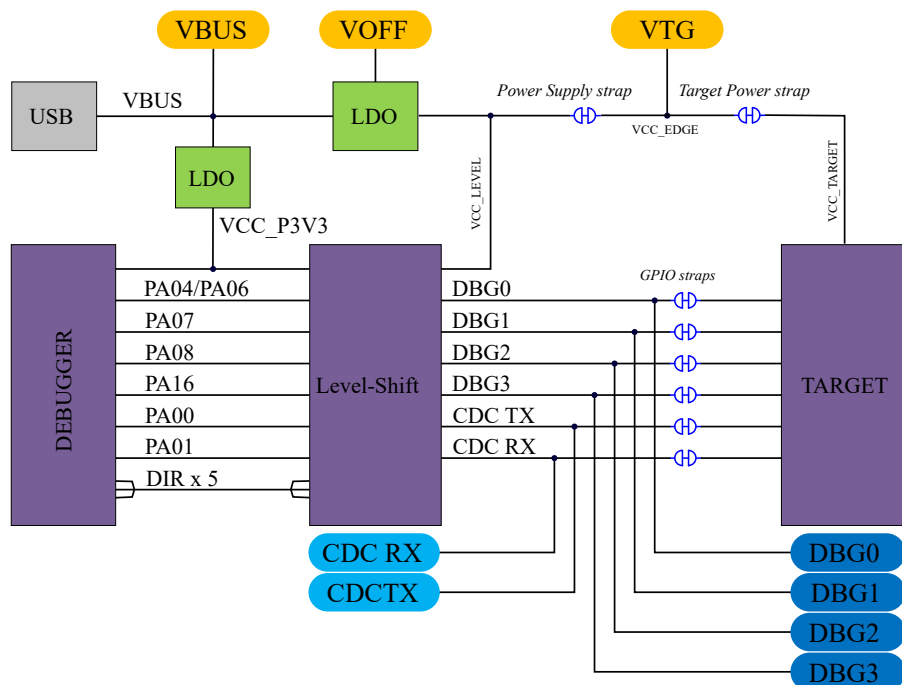
By cutting the *GPIO straps* with a sharp tool, as shown in Figure 3-7, all I/Os connected between the debugger and the ATtiny1607 are completely disconnected. To completely disconnect the target regulator and level shifter power from the target, cut the *Power Supply strap* (J100) as shown in Figure 3-7.

> **Info:** Cutting the connections to the debugger will disable programming, debugging, data streaming, and the target power supply. The signals will also be disconnected from the board edge next to the on-board debugger section.

> **Tip:** Solder in 0Ω resistors across the footprints or short-circuit them with tin solder to reconnect any cut signals.
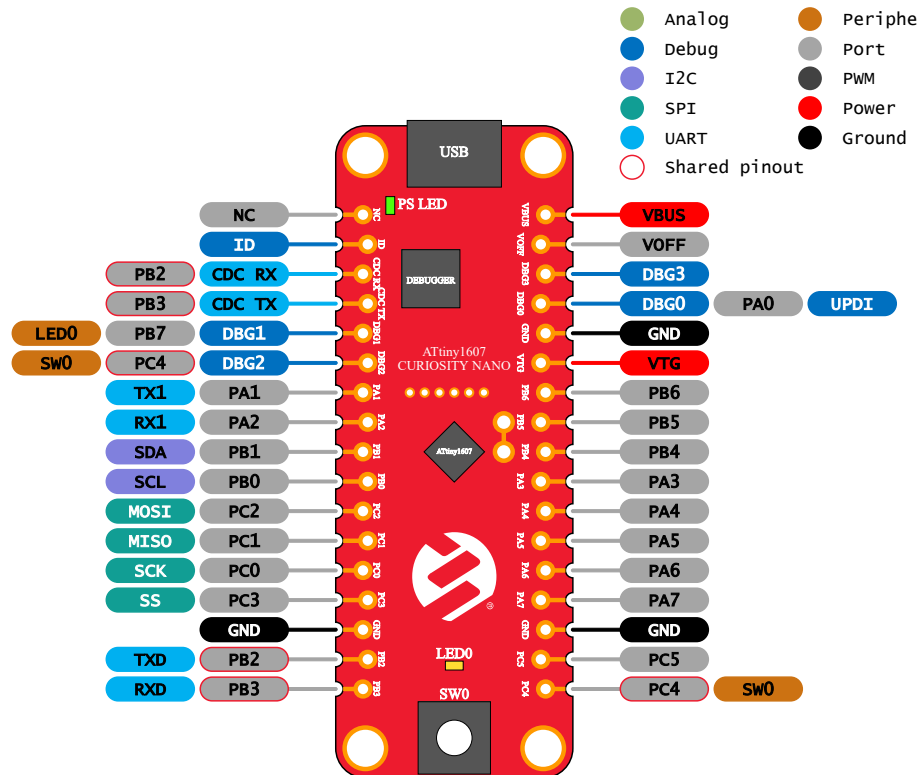
Figure 3-7. Kit Modifications
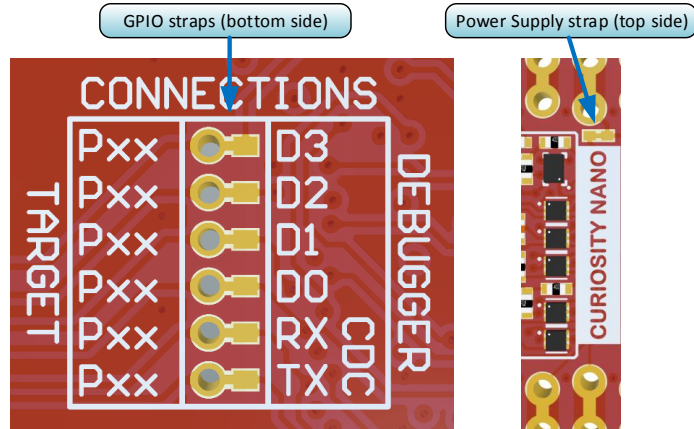
4. Hardware User Guide

4.1 Connectors

4.1.1 ATtiny1607 Curiosity Nano Pinout

All the ATtiny1607 I/O pins are accessible at the edge connectors on the board. The image below shows the kit pinout.

Figure 4-1. ATtiny1607 Curiosity Nano Pinout

4.1.2 Using Pin Headers

The edge connector footprint on ATtiny1607 Curiosity Nano has a staggered design where each of the holes is shifted 8 mil (~0.2 mm) off center. The hole shift allows the use of regular 100-mil pin headers on the kit 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 all the way 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 printed circuit board.

4.2 Peripherals

4.2.1 LED

There is one yellow user LED available on the ATtiny1607 Curiosity Nano kit 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

ATtiny1607 Pin	Function	Shared Functionality
PB7	Yellow LED0	Edge connector

4.2.2 Mechanical Switch

ATtiny1607 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 PC4.

Table 4-2. Mechanical Switch

ATtiny1607 Pin	Description	Shared Functionality
PC4	User switch (SW0)	Edge connector

4.2.3 On-Board Debugger Implementation

ATtiny1607 Curiosity Nano features an on-board debugger that can be used to program and debug the ATtiny1607 using UPDI. The on-board debugger also includes a Virtual Com port interface over UART and DGI GPIO. Atmel Studio/Microchip MPLAB® X can be used as a front-end for the on-board debugger for programming and debugging. Data Visualizer can be used as a front-end for the CDC and DGI GPIO.

4.2.3.1 On-Board Debugger Connections

The table below shows the connections between the target and the debugger section. All connections between the target and the debugger are tri-stated as long as the debugger is not actively using the interface. Hence there is little contamination 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 Section 3. Curiosity Nano.

Table 4-3. On-Board Debugger Connections

ATtiny1607 Pin	Debugger Pin	Function	Shared Functionality
PB3	CDC TX	UART RX (ATtiny1607 RX line)	Edge connector
PB2	CDC RX	UART TX (ATtiny1607 TX line)	Edge connector
UPDI	DBG0	UPDI	
PB7	DBG1	GPIO	Edge connector
PC4	DBG2	GPIO	Edge connector
UPDI	DBG3	RESET (*J202*, not connected by default)	Edge connector

5. Hardware Revision History and Known Issues

This user guide provides the latest available revision of the kit. This section contains 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 ATtiny1607 Curiosity Nano can be found in two ways; either through Atmel Studio/Microchip MPLAB® X or by looking at the sticker on the bottom side of the PCB.

By connecting a ATtiny1607 Curiosity Nano to a computer with Atmel Studio/Microchip MPLAB® X running, an information window will pop up. The first six digits of the serial number, which is listed under kit details, contain the product identifier and revision.

The same information can be found on the sticker on the bottom side of the PCB. Most kits will print the identifier and revision in plain text as A09-nnnn\rr, where "nnnn" is the identifier and "rr" is the revision. The boards with limited space have a sticker with only a QR-code, containing the product identifier, revision and the serial number.

The serial number string has the following format:

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

The product identifier for ATtiny1607 Curiosity Nano is A09-3252.

5.2 Chip Erase at Low Voltage

This kit supports variable voltage from 1.8-5.1V. When a chip erase of the ATtiny1607 is started, the BOD is enabled and uses the BODLEVEL set in the FUSE.BODCFG fuse byte. If the BODLEVEL is set higher than the selected target voltage on the kit, chip erase will fail.

> **Info:** BODLEVEL0 in the ATtiny1607 has a maximum trigger voltage of 2.0V, which means that chip erase is likely to fail if the kit voltage is set to 1.8V.

5.3 Revision 2

Revision 2 adds the *Target Power strap* and staggers the holes along the edge of the PCB for convenient use of pin headers without soldering.

5.4 Revision 1

Revision 1 is the initially released revision with limited distribution.

The holes along the edge of revision 1 are not staggered as described in Using Pin Headers and requires that any pin headers must be soldered into the board for use.

Revision 1 does not have the *Target Power strap* described in 3.4 Target Current Measurement. Instead, the current can be measured across the *Power Supply strap*, as described in 3.5 Disconnecting the On-Board Debugger.

Figure 5-1. ATtiny1607 Curiosity Nano Revision 1
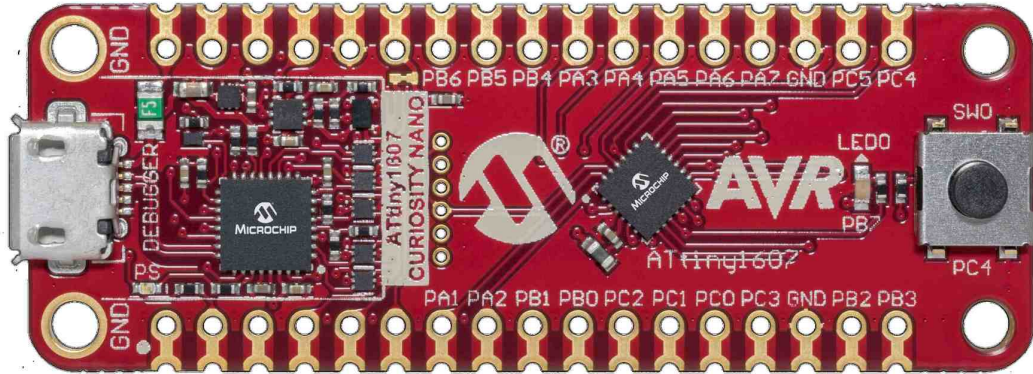
6.   Document Revision History

Doc. rev.	Date	Comment
A	06/2019	Initial document release.

7. Appendix

7.1 Schematic

Figure 7-1. ATtiny1607 Curiosity Nano Schematic

TARGET ADJUSTABLE REGULATOR

DEBUGGER REGULATOR

DEBUGGER POWER/STATUS LED

DEBUGGER USB MICRO-B CONNECTOR

ID PIN

DEBUGGER TEST POINTs

DBG3 OPEN DRAIN

7.2 Assembly Drawing

Figure 7-2. ATtiny1607 Curiosity Nano Assembly Drawing Top
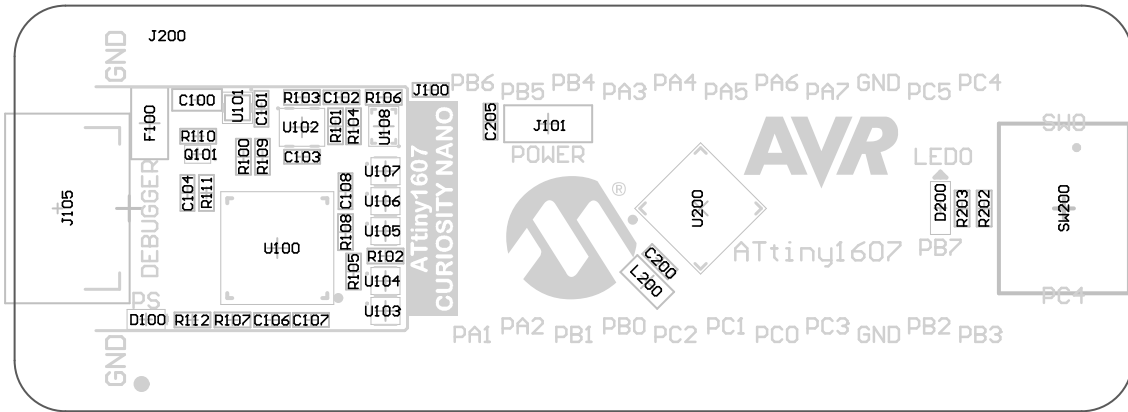
Figure 7-3. ATtiny1607 Curiosity Nano Assembly Drawing Bottom
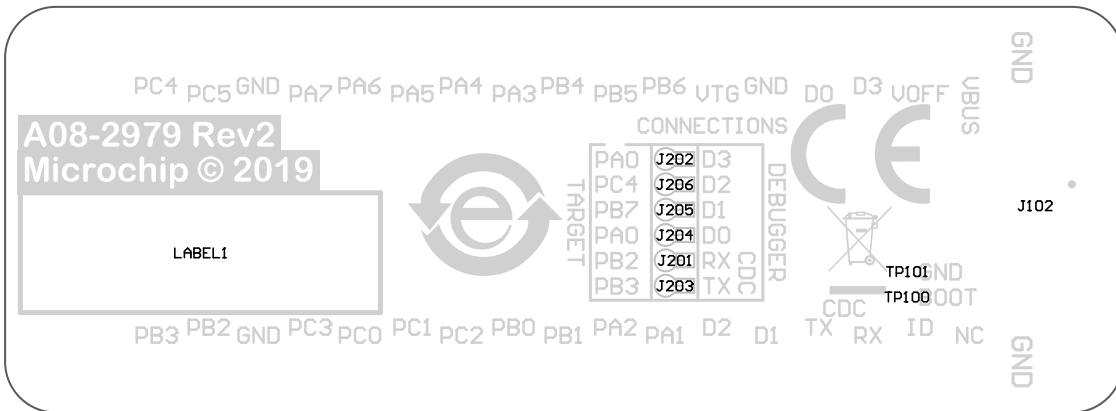
7.3 Curiosity Nano Base for Click boards™

Figure 7-4. ATtiny1607 Curiosity Nano Pinout Mapping
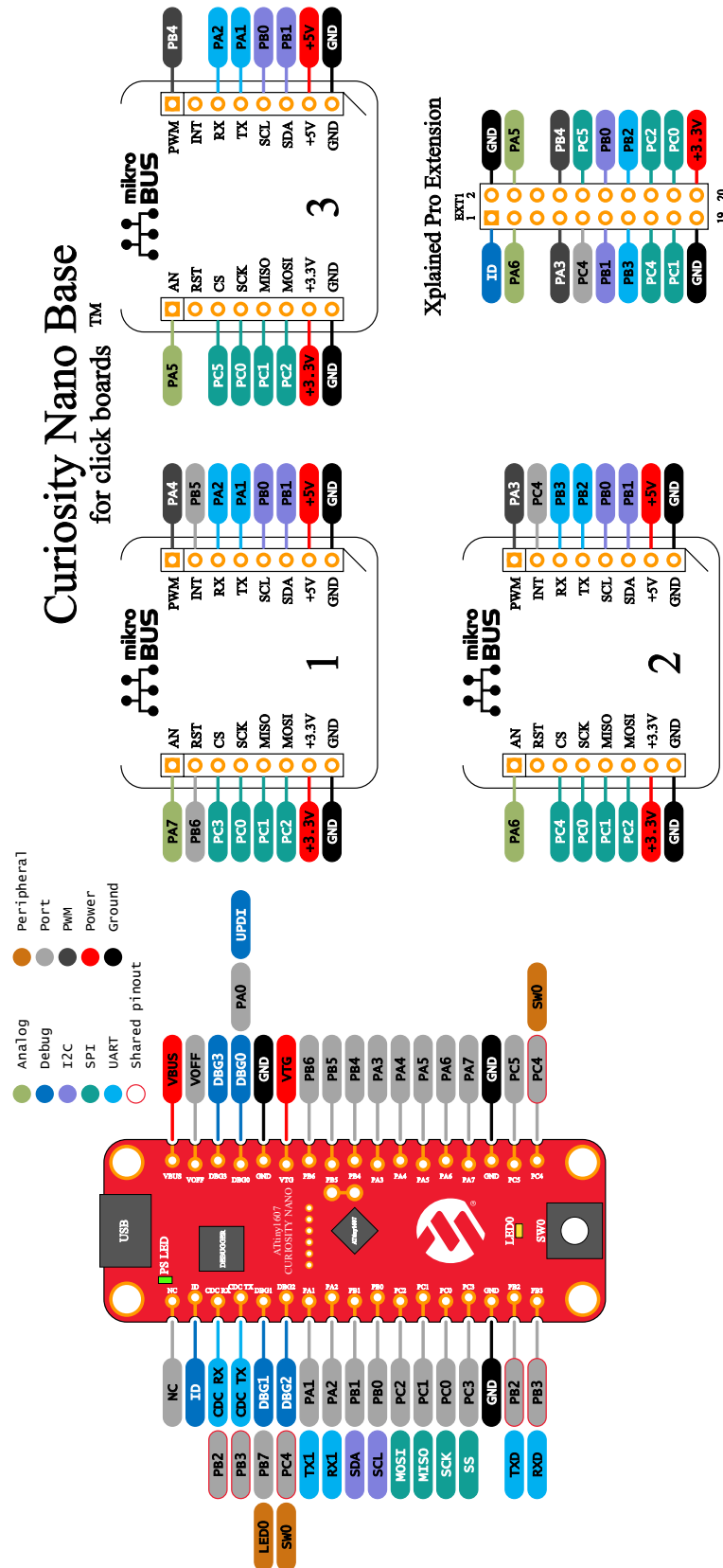
7.4 Connecting External Debuggers

Even though there is an on-board debugger, external debuggers can be connected directly to ATtiny1607 Curiosity Nano to program/debug the ATtiny1607. The on-board debugger keeps all the pins connected to the ATtiny1607 and board edge in tri-state when not actively used. Therefore, the on-board debugger will not interfere with any external debug tools.

Figure 7-5. Connecting the MPLAB PICkit™ 4 In-Circuit Debugger/Programmer to ATtiny1607 Curiosity Nano
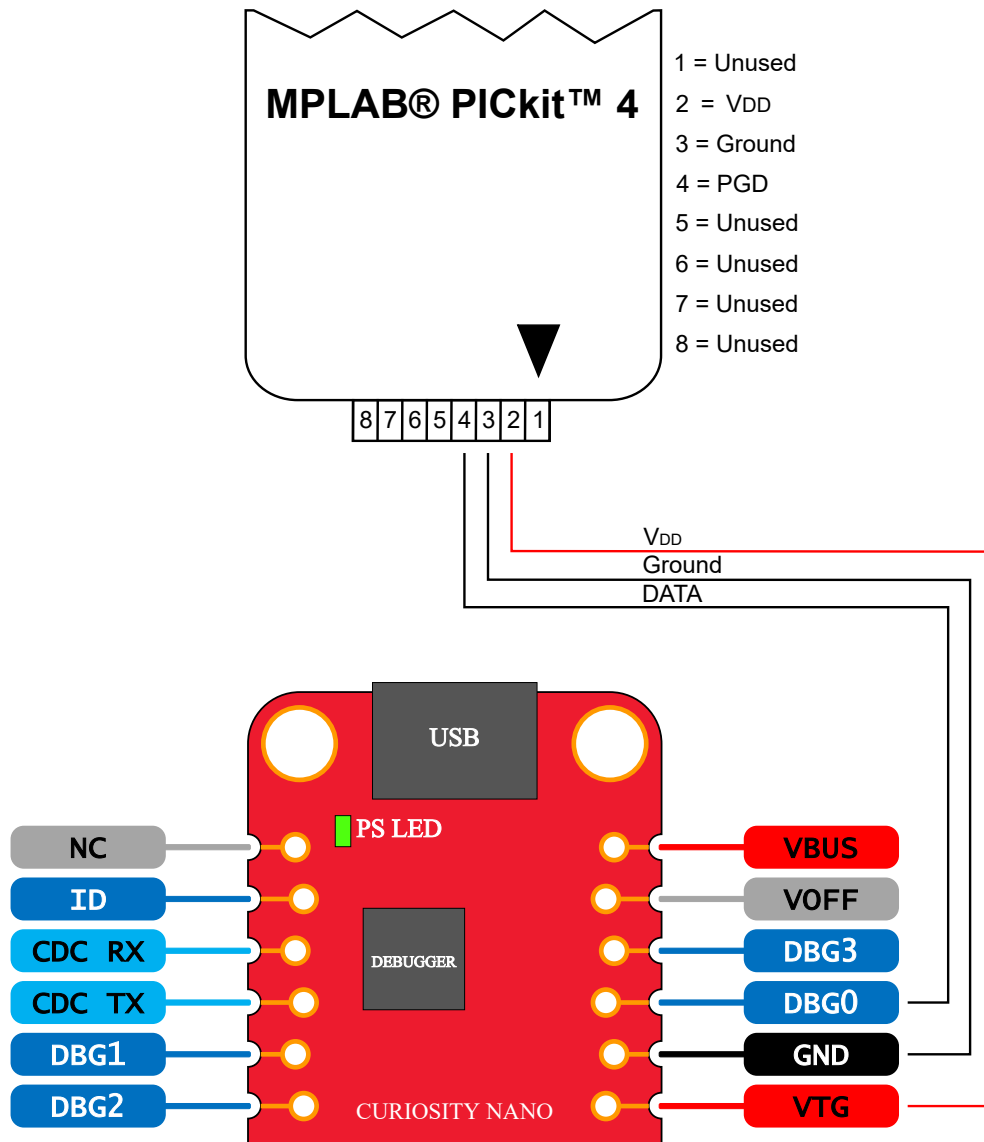
Figure 7-6. Connecting the Atmel-ICE to ATtiny1607 Curiosity Nano

1 = Unused   6 = Unused
2 = GND   7 = Unused
3 = UPDI   8 = Unused
4 = VTG   9 = Unused
5 = Unused   10 = Unused
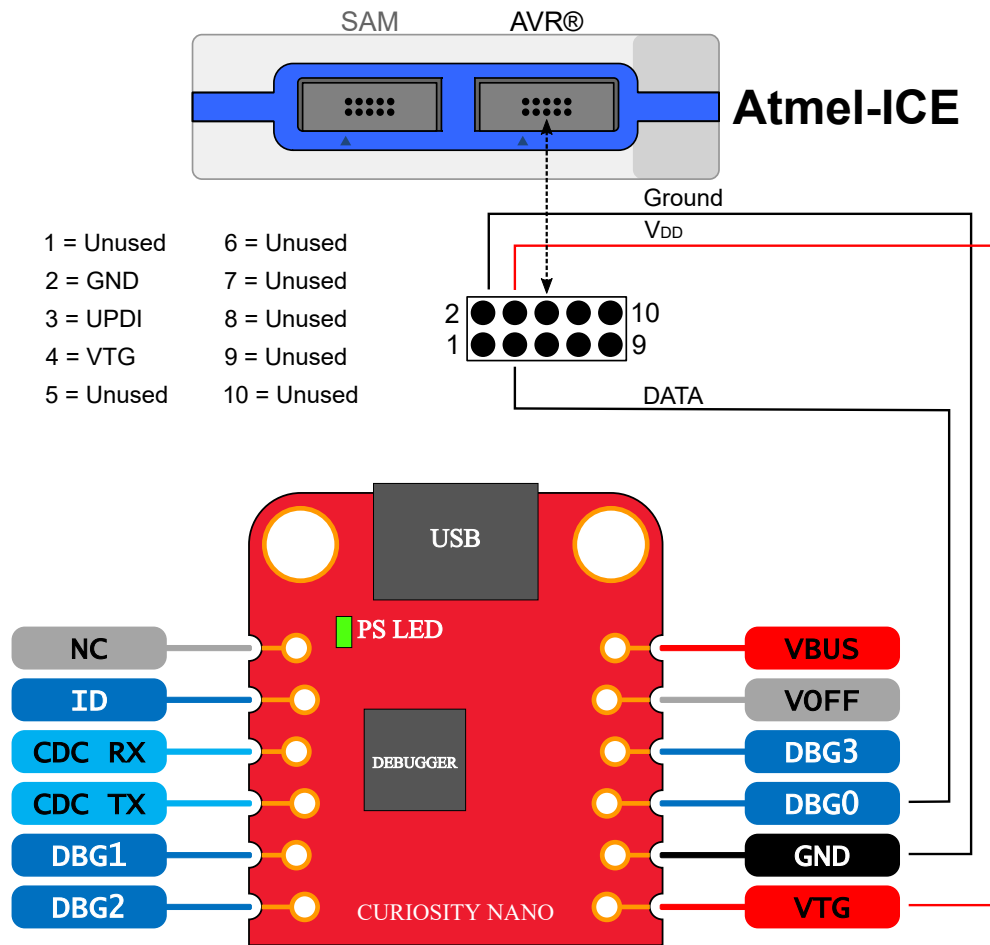
⚠ **CAUTION**   To avoid contention between the external debugger and the on-board debugger, do not start any programming/debug operation with the on-board debugger through Atmel Studio/Microchip MPLAB® X or mass storage programming while the external tool is active.

7.5 Getting Started with IAR

IAR Embedded Workbench® for AVR® is a proprietary high-efficiency compiler which is not based on GCC. Programming and debugging of ATtiny1607 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.
3. In the category **Debugger**, select the **Setup** tab. Select **Atmel-ICE** as the driver.

4. In the category **Debugger** > **Atmel-ICE**, select the **Atmel-ICE 1** tab. Select UPDI as the interface and, optionally, select the UPDI frequency.

> **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
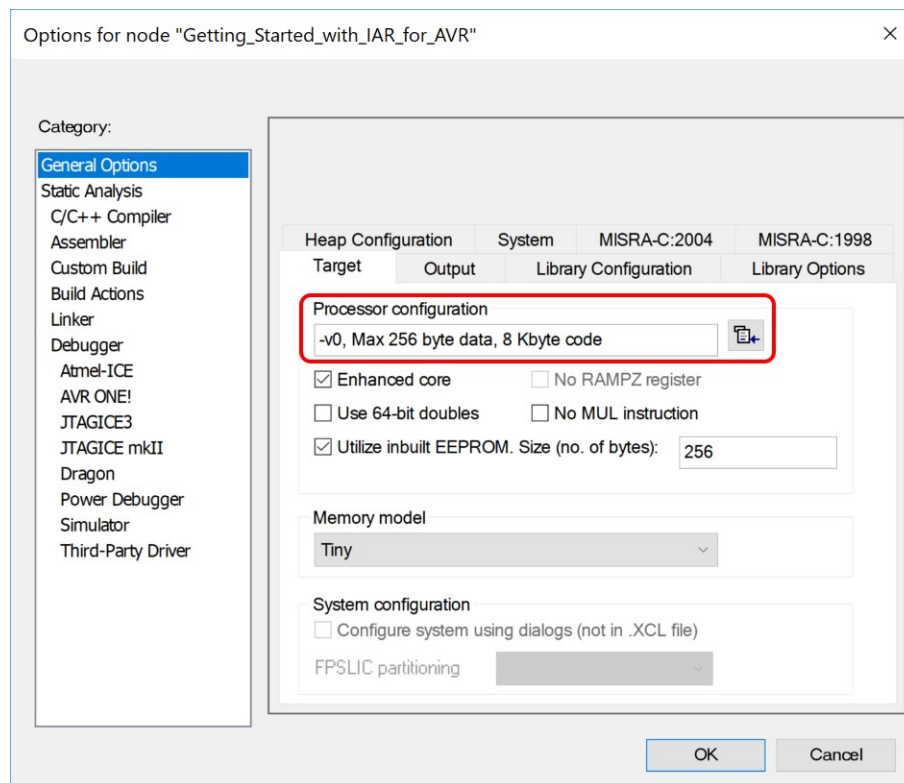
Figure 7-8. Select Debugger
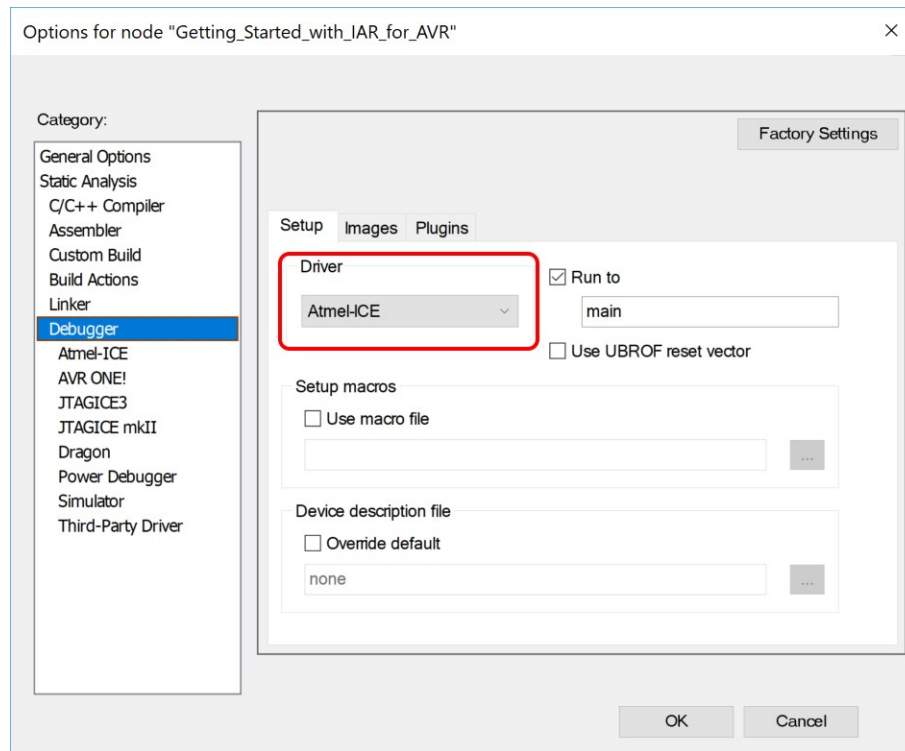
Figure 7-9. Configure Interface
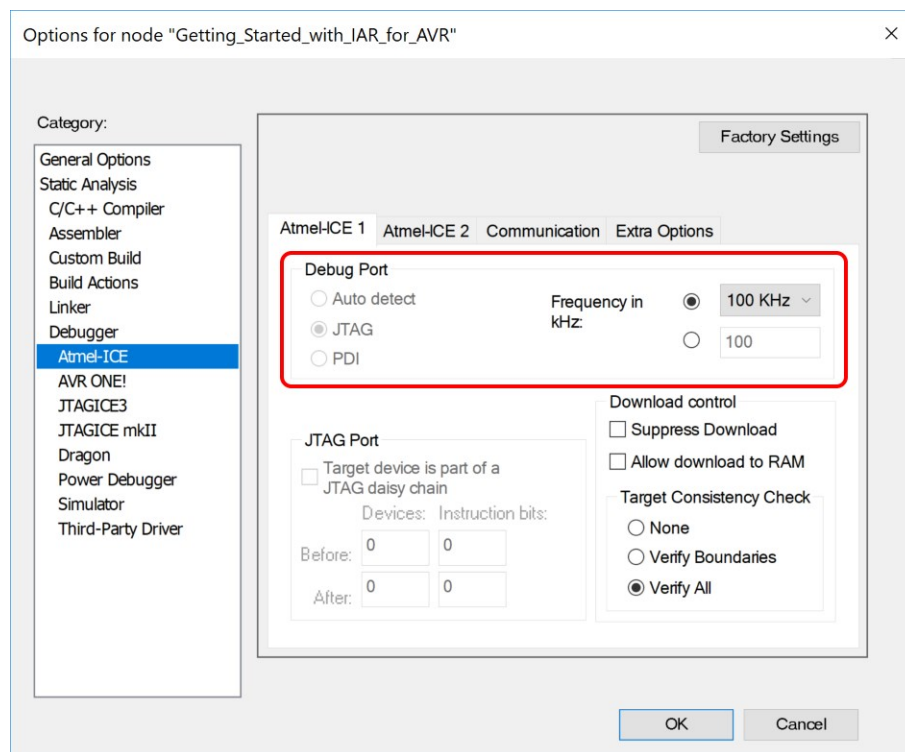
The Microchip Website

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 web site 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."

Legal Notice

Trademarks

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: 45-4450-2828 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	**Thailand - Bangkok** Tel: 66-2-694-1351	**Italy - Padova** Tel: 39-049-7625286
Houston, TX Tel: 281-894-5983	**China - Xian** Tel: 86-29-8833-7252	**Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340
Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380	**China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040		**Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737
Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800			**Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91
Raleigh, NC Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000			**Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654
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			