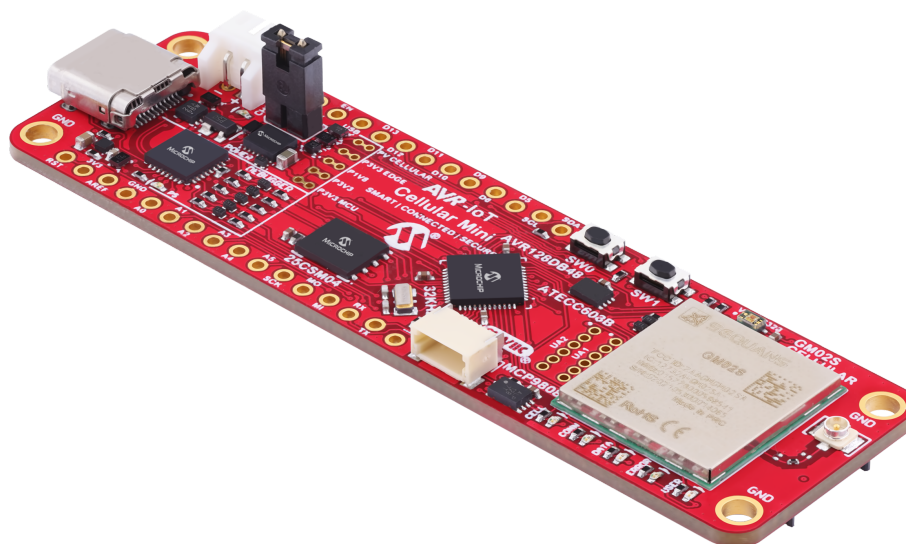

AVR-IoT Cellular Mini Hardware User Guide

Preface

Introduction

The AVR-IoT Cellular Mini Development Board is a small and easily expandable demonstration and development platform for IoT solutions based on the AVR® microcontroller architecture using cellular technology. It is designed to demonstrate that the design of a typical IoT application can be simplified by partitioning the problem into three blocks:

- Smart – represented by the [AVR128DB48](#) microcontroller
- Connected – represented by the [GM02S](#) cellular module, with the included antenna and SIM card
- Secure – represented by the [ATECC608B](#) secure element



In addition, the AVR-IoT Cellular Mini Development Board features the following elements:

- Full [Arduino](#) support through a [library](#) built on top of the open-source [DxCore](#)
- The on-board debugger (PKOB nano) supplies full programming support for the Arduino IDE. There is no need for any external tools, and it also provides access to a serial port interface (serial to USB bridge) and two logic analyzer channels (debug GPIO).
- The on-board debugger enumerates on the PC as a mass storage interface device for easy 'drag-and-drop' programming.
- A [Qwiic](#)® connector and the [Feather](#) compatible edge connector allow for easy expansion from a wide selection of add-on boards from Sparkfun and Adafruit
- Two on-board sensors; Color Sensor ([VEML3328](#)) and Temperature sensor ([MCP9808](#))
- Microchip [MCP73830](#) Li-Ion/LiPo battery charger
- Microchip [25CSM04](#) 4 Mb Serial EEPROM
- [Microchip MPLAB® X](#) and [Microchip Studio](#) support bare-metal development
- [AVR128DB48 website](#) - Find documentation, data sheets, samples, and purchase microcontrollers.
- [AVR-IoT Cellular Mini website](#) - Kit information, latest user guide, and design documentation.

Table of Contents

Preface.....	1
1. Introduction.....	3
1.1. Features.....	3
1.2. Board Overview.....	3
2. Getting Started.....	6
2.1. Quick Start.....	6
2.2. Design Documentation and Relevant Links.....	8
3. Hardware Description.....	9
3.1. Power.....	9
3.2. Peripherals.....	13
3.3. Sensors.....	17
3.4. External Connectors.....	18
4. On-Board Debugger.....	21
4.1. On-Board Debugger Overview.....	21
4.2. On-Board Debugger Implementation.....	27
5. Hardware Revision History and Known Issues.....	28
5.1. Identifying Product ID and Revision.....	28
5.2. Revision 5.....	28
6. Document Revision History.....	29
7. Appendix.....	30
7.1. Schematics.....	30
7.2. Assembly Drawing.....	34
7.3. AVR-IoT Cellular Mini w/ MPLAB X and Microchip Studio.....	34
Microchip Information.....	36
The Microchip Website.....	36
Product Change Notification Service.....	36
Customer Support.....	36
Microchip Devices Code Protection Feature.....	36
Legal Notice.....	36
Trademarks.....	37
Quality Management System.....	38
Worldwide Sales and Service.....	39

1. Introduction

1.1 Features

- AVR128DB48 AVR® Microcontroller
- ATECC608B CryptoAuthentication™ Device
- 25CSM04 Serial EEPROM – 4 Mb
- GM02S Cellular Module
- 824-2170 MHz Cellular Flexible Antenna
- Prepaid SIM Card from Truphone – 150 MB Data Included
- Nano SIM Card Holder
- eSIM Footprint
- Five User LEDs
- Two Mechanical Buttons
- VMK3-9002 32.768 kHz Crystal
- VEML3328 Color Sensor
- MCP9808 Temperature Sensor
- Adafruit Feather Compatible Edge Connector
- Qwiic® I²C Connector
- On-board Debugger
 - Board identification in Microchip Studio and Microchip MPLAB® X
 - One green power and status LED
 - Virtual serial port (USB CDC)
 - Two logic analyzer channels (debug GPIO)
- Arduino Compatible with the [Cellular Arduino Library](#)
- USB, Battery, and External-Input Power Options
- MCP73830 Li-Ion/LiPo Battery Charger, with Charge Status LED
- Two Fixed Power Domains, 1.8V and 3.3V

1.2 Board Overview

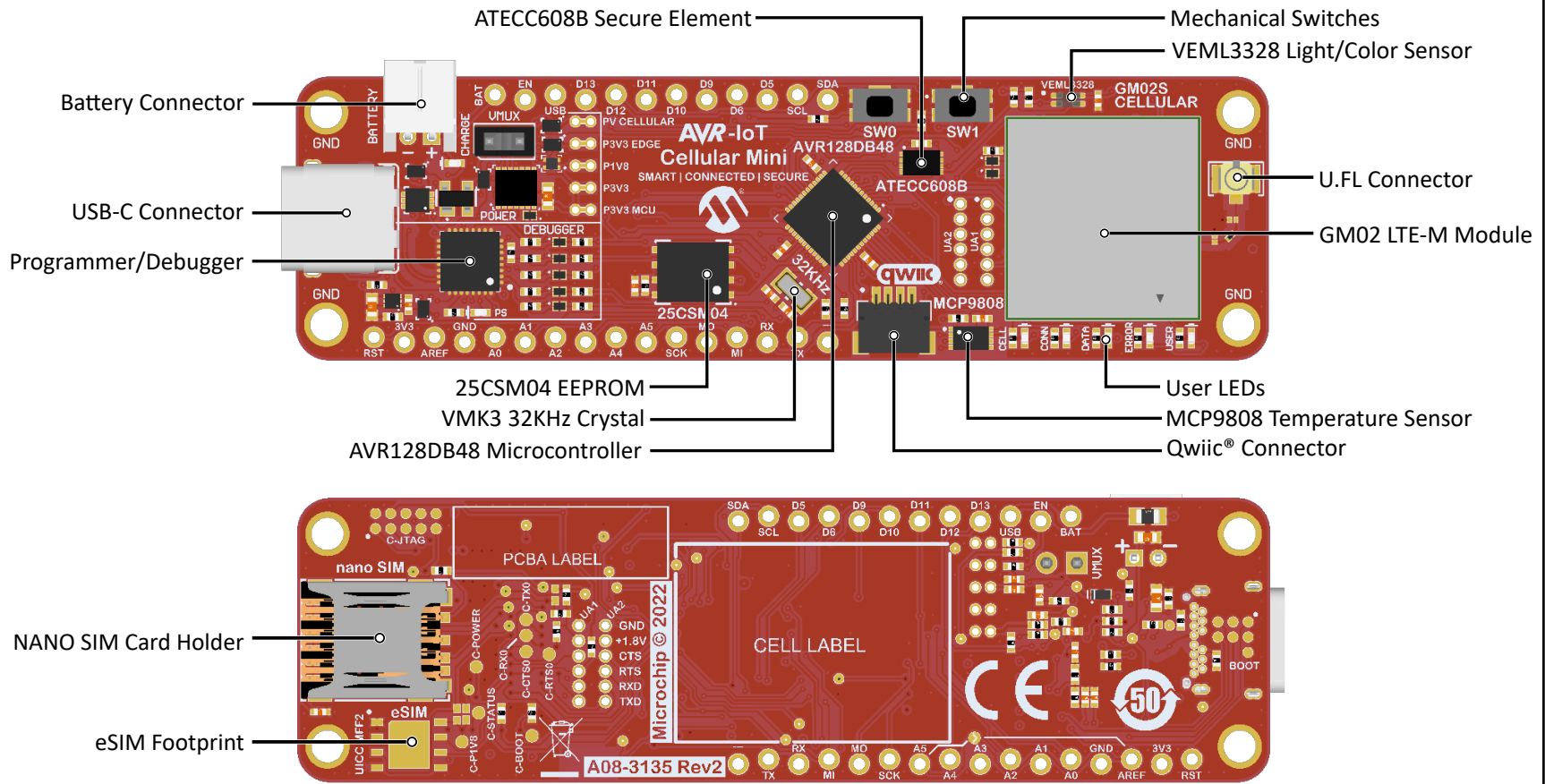
The AVR-IoT Cellular Mini development board is a hardware platform to evaluate and develop IoT solutions with the Microchip AVR128DB48 AVR microcontroller, ATECC608B secure element, and the GM02S cellular module. Included with the development board is a SIM card from Truphone with 150 MB of data ready for [activation](#).

The GM02S requires a connection to an LTE-M or NB-IoT capable cellular network. With a connection to a capable cellular network, the ATECC608B can be used to authenticate the hardware with the cloud to identify every unique board.

The development board comes preprogrammed with a demo application, the Microchip Sanbox, to get started with the AVR-IoT Cellular Mini.

The figure below shows the main features of the board.

Figure 1-1. AVR-IoT Cellular Mini Development Board Overview



1.2.1 Arduino Compatability

The AVR-IoT Cellular Mini has complete [Arduino](#) support through a [library](#) built on the top of the open-source [DxCore](#).

To get started with development in the Arduino IDE, follow the steps at iot.microchip.com to install all the required dependencies.



Tip: When the required dependencies are installed, development for the AVR-IoT Cellular Mini board is also supported in Visual Studio Code with the [extension for Arduino](#).

1.2.2 IoT Provisioning Tool

The [IoT Provisioning Tool](#) is an easy-to-use solution for configuring an AVR-IoT Cellular Mini to connect to a supported cloud provider, set the network provider, and select cellular frequency bands.



Tip: The AVR-IoT Cellular Mini must be provisioned for the AWS Microchip sandbox for the demo firmware to work.

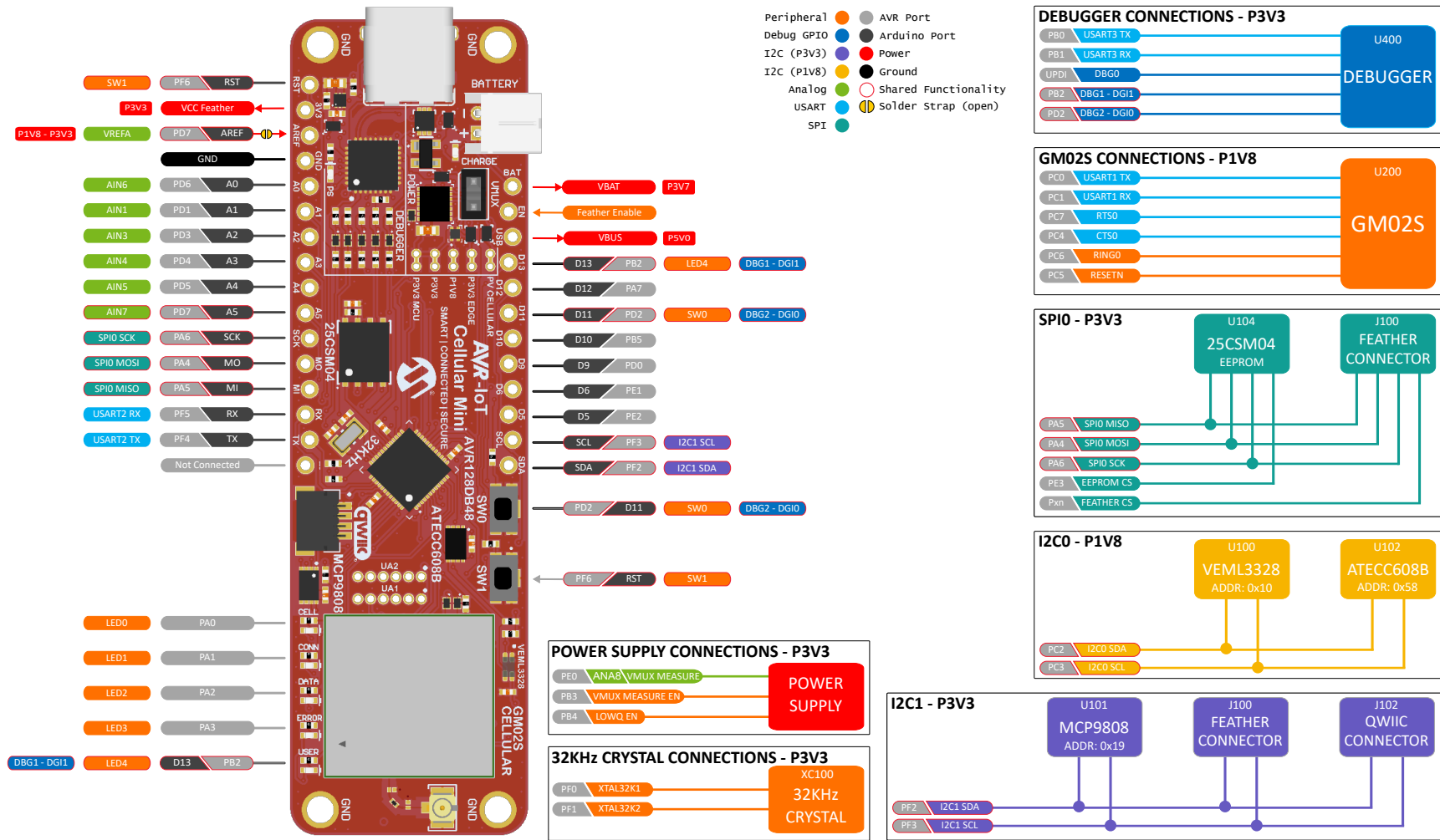
2. Getting Started

2.1 Quick Start

The AVR128DB48 device on the AVR-IoT Cellular Mini board is programmed and debugged by the on-board debugger, and therefore, no external programmer or debugger tool is required. The AVR-IoT Cellular Mini board comes programmed with a demo application. To get started with the development board, follow these steps:

1. [Activate](#) the included SIM card.
2. Insert the included SIM card in the SIM card holder, and connect the external antenna to the board.
3. Connect the debug USB-C port on the board to your PC.
4. Scan the QR code on the bottom of the board, or open the [mass-storage device](#) and follow the CLICK-ME.HTM to the kit web page.
5. Follow the steps on the web page to get started developing your IoT application.

Figure 2-1. AVR-IoT Cellular Mini Pinout



2.2 Design Documentation and Relevant Links

The following list contains links to the most relevant documents and software for the AVR-IoT Cellular Mini board:

- [AVR-IoT Cellular Mini website](#) - Kit information, latest user guide, and design documentation.
- [AVR-IoT Cellular Mini on Microchip Direct](#) - Purchase this kit on Microchip Direct.
- [Microchip IoT Documentation](#) - Find the latest guides and documentation for getting started with Microchip IoT boards
- [Arduino IDE](#) - The Arduino IDE is a set of tools to develop, compile and upload Arduino code onto compatible devices.
- [DxCore](#) - Arduino support for the AVR Dx-series
- [AVR-IoT Cellular Arduino Library](#) - Offers an API to communicate with the systems and components on the board.
- [IoT Provisioning Tool](#) - Provision the AVR-IoT Cellular Mini board for use with a cloud provider and/or a network provider for cellular SIM-card.
- [Microchip Sample Store](#) - Microchip sample store where one can order samples of devices.
- [MPLAB Data Visualizer](#) - MPLAB Data Visualizer is a program used for processing and visualizing data. The Data Visualizer can receive data from various sources such as serial ports and on-board debugger's Data Gateway Interface, as found on Curiosity Nano and Xplained Pro boards.
- [Microchip PIC® and AVR® Solutions](#) - Microchip PIC and AVR Device Solutions contain complete applications for use with Microchip development boards, ready to be adapted and extended.

3. Hardware Description

3.1 Power

3.1.1 Power Supply

The board can be powered through the USB port, a Li-Ion/LiPo battery, or by an external power source. The board contains one buck converter for 3.3V with LDO standby mode and one LDO for 1.8V. The debugger is supplied from its own LDO and will only be powered when the board is connected to USB power. The [Sequans GM02S](#) module is powered directly from the input power and generates a 1.8V output voltage.



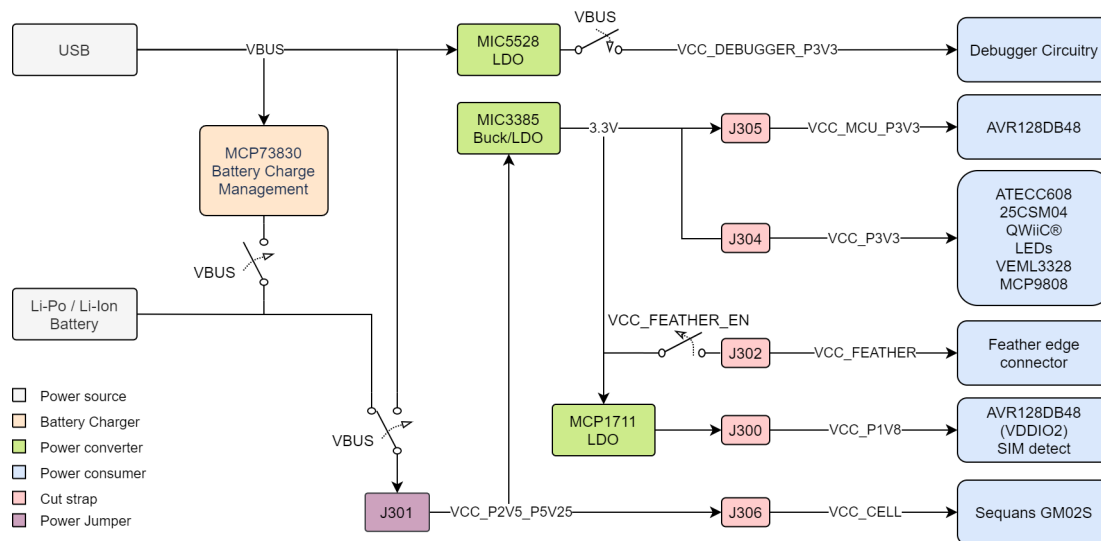
Info: The maximum available current from the USB is limited to 500 mA. The current is shared between charging the battery (if connected) and the system.



Important: When not powered from a USB, it is important to leave the AVR128DB48 pins connected to the CDC UART in Tri-State (Input) mode to prevent the debugger from getting powered through its GPIO.

The power supply is designed with four power switches ([MIC94165](#)) to prevent leakage when powered by a battery. All three switches are open when not connected to USB power, disabling battery charging and the on-board debugger. The Feather edge connector is powered-on by default.

Figure 3-1. Power Supply Block Diagram



WARNING Any external power applied to the board must be in the range of 2.5–5.25V. Any voltage Exceeding 5.25V may damage the board.

When the board is supplied by a battery, the only limitation regarding current is caused by the voltage regulators. The maximum output current from the two regulators, MIC3385 and MCP1711, is listed in the table below.

Table 3-1. Output Current Limit

Regulator	Output Current Limit [mA]
MCP1711	150
MIC3385 (PWM)	900
MIC3385 (LDO)	120

3.1.2 Battery Charger

AVR-IoT Cellular Mini features an [MCP73830](#) Li-Ion/LiPo charger and a JST battery connector on board. The charger is configured to limit the charge current to 500 mA to prevent overcharging of small-capacity batteries. The charge cycle is configured to terminate when the charge current falls below 37 mA or if the charge cycle is longer than the safety timer of four hours.



Info: The minimum recommended battery capacity is 400 mAh.



The MCP73830 has a battery charge voltage of 4.2V. Make sure your battery is rated for the same charge voltage.

The battery charger will enter a preconditioning mode if the voltage at the V_{BAT} pin is less than 3.5V, and in this mode only 10% of the fast charge current is supplied to the battery. The preconditioning mode has a separate safety timer of one hour. If a charge cycle does not exit this mode within the timer window, the charge cycle is terminated. If the preconditioning safety timer is triggered, it may indicate a faulty battery.



Make sure to check your battery's polarity before connecting it. There is no reverse current/battery protection circuitry on the board.

The battery charging circuitry includes one red LED to indicate the charge status. See the table below:

Table 3-2. Charge LED

LED	Charge Status
On	Preconditioning, constant-current fast charging, or constant voltage charging
Off	No battery present, shutdown, charge complete, or timer fault
Flashing	A preconditioning timer fault may indicate a faulty battery

3.1.3 Multi-Voltage I/O

The AVR128DB48 MCU has support for Multi-Voltage I/O (MVIO). Port C is reserved for MVIO and is powered from VDDIO2 at 1.8V. The MVSYS_CFG fuse controls the AVR128DB48's supply state. If the fuse is set to single-supply mode, all pins in port C will be tri-stated as the VDDIO2 supply voltage will be below the acceptable range for operation. The default setting for the MVSYS_CFG fuse is the dual-supply mode.



Info: 1.8V is selected for VDDIO2 to satisfy the Sequans GM02S module's requirement of 1.8V I/O without the need for external level shifters.

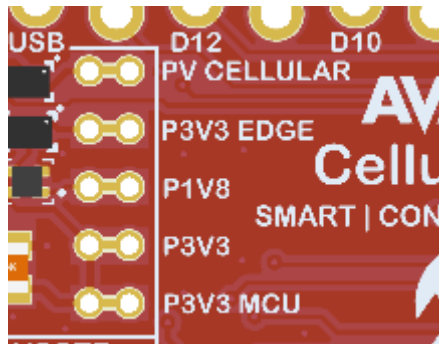
Table 3-3. MVIO pins

AVR128DB48 Pin	Function	Connected Peripheral
PC0	USART1 TX	GM02S
PC1	USART1 RX	GM02S
PC2	I2C0 SDA	ATECC608B, VEML3328
PC3	I2C0 SCL	ATECC608B, VEML3328
PC4	CTS0	GM02S
PC5	RESETEN	GM02S
PC6	RING0	GM02S
PC7	RTS0	GM02S

3.1.4 Power Measurement

Power to all the board parts are connected through five cut-straps with 50-mil holes, and marked in silkscreen. These are intended for current measurement purposes. Do not leave these unconnected as the MCU, sensors, or other connected circuitry might get powered through the I/Os.

Figure 3-2. Cut-Straps



WARNING Before doing any hardware modifications, make sure the board is disconnected from USB, battery or external power.

To measure power on the desired net, cut the cut-strap and connect an ammeter to the 50-mill holes.

3.1.4.1 System Voltage Measurement

The AVR-IoT Cellular Mini board has a system voltage measurement circuit using the [MIC94163](#) switch and a voltage divider connected to an ADC pin on the AVR128DB48, enabling on demand measurement and preventing leaking power through the voltage divider.

Table 3-4. System Voltage Measurement Pins

AVR128DB48 Pin	Function
PE0	ADC0 (AIN8) - System voltage measurement
PB3	GPIO - System voltage measurement enable

The system voltage is available as an input on PE0 and is divided by four. To measure the system voltage, follow these steps:

1. Configure the voltage reference for the ADC.
2. Set PB3 high to enable the voltage divider.

3. Set PE0 as input for the ADC.
4. Run a single-ended ADC conversion.
5. Calculate the voltage using the following equation:
 - a.
$$V = \frac{\text{ADC Result} \times V_{\text{REF}} \times 4}{\text{ADC Resolution}}$$



Tip: For power conservation, disable PE0, PB3 and the ADC after the measurement.

3.1.4.2 Measuring Supply Voltage

VDD and VDDIO2 are available internal input channels to the AVR128DB48's ADC. The voltage is divided by ten. To measure VDD or VDDIO2, follow these steps:

1. Configure the voltage reference for the ADC.
2. Select VDD or VDDIO2 as the positive input to the ADC.
3. Run a single-ended ADC conversion.
4. Calculate the voltage using the following equation:
 - a.
$$V = \frac{\text{ADC Result} \times V_{\text{REF}} \times 10}{\text{ADC Resolution}}$$

3.1.5 Low-Power Operation

Take the following considerations to achieve the lowest power consumption possible:

- [MCP9808](#) in Low-Power mode (shutdown)
 - Set bit 8 (SHDN) in the 16-bit configuration register (address 0x01)
- [VEML3328](#) powered off
 - Send shut down command (low byte 0x11, high byte 0x80) to command register (address 0x00)
- [ATECC608N](#) in sleep mode
 - Send the sleep command (0x01) to the ATECC608B
- [GM02S](#) in Low-Power mode
 - Low-Power mode is active by default,
 - To enable/disable Low-Power mode, issue the following AT commands:

```
AT+CFUN=5
AT+SQNHWCFG="lpm"[, ("enable", "disable")]
AT^RESET
```

- The module must be set to “Manufacturing Mode” before applying the AT command, and the settings are latched by issuing the reset AT command.
- Set unused AVR128DB48 I/O pins as input and disable the digital input buffer
 - Make sure to disable PB3 as it is used to enable the circuitry for system voltage measurement
- AVR128DB48 in Power-Down sleep mode
- Enable LOWQ® LDO mode for the 3.3V regulator ([MIC3385](#)) by setting PB4 high

3.1.6 External Power

There are two ways to add external power to the AVR-IoT Cellular Mini board:

1. Through the USB pin on the [edge connector](#).
2. Through the VMUX 100-mil pin header.



WARNING Before doing any hardware modifications, make sure the board is disconnected from USB, battery or external power.



When powering the board with an external source, make sure that the voltage is between 2.5V and 5.25V. Any voltage exceeding this may damage the board.



When using external power, make sure that the board is not connected to a battery or USB power. If it is, it may damage your board.

3.2 Peripherals

3.2.1 AVR128DB48

Microchip's [AVR128DB48](#) microcontroller features Intelligent Analog, Core Independent Peripherals (CIPs), three integrated OPAMPS, and the Multi-Voltage I/O (MVIO) system. The features of the AVR128DB48 make it well suited for Low-Power IoT applications.

3.2.2 ATECC608B

The [ATECC608B](#) is a secure element from the Microchip CryptoAuthentication™ portfolio with advanced Elliptic Curve Cryptography (ECC) capabilities. With ECDH and ECDSA built right in, this device is ideal for the rapidly growing IoT market by supplying the full range of security such as confidentiality, data integrity, and authentication to systems with MCU or MPUs running encryption/decryption algorithms. Identical to all Microchip CryptoAuthentication products, the new ATECC608B employs ultra-secure hardware-based cryptographic key storage and cryptographic countermeasures which eliminate potential backdoors linked to software weaknesses.

Use the ATECC608B CryptoAuthentication device on the AVR-IoT Cellular Mini board can to authenticate the board with other hardware for secure IoT communication.



Info: The 7-bit I²C address is: 0x58.

Table 3-5. ATECC608B Connections

ATECC608B Pin	AVR128DB48 Pin	Function	Shared Functionality
SDA	PC2	TWI0 SDA	VEML3328
SCL	PC3	TWI0 SCL	

3.2.3 25CSM04 EEPROM

The Microchip [25CSM04](#) is a 4 Mb Serial EEPROM with a Serial Peripheral Interface (SPI) capable of operation across a broad voltage range. The device is organized as one block of 524,288 bytes of 8-bits each (512 KB), including a security register with a 128-bit unique serial number and a 256-byte (2048-bit) user-programmable lockable ID page, built-in Error Correction Code, and a configurable write protection scheme for all bytes.

Additional Features:

- Data Retention > 100 years
- Operating Voltage Range: 2.5-5.5V
- High-Speed Clock Frequency of 8 MHz at $V_{cc} > 3.0V$; 5MHz at $V_{cc} > 2.5V$
- Write current: 3.0 mA at 5V; Read current: 3.0 mA at 5V, 8 MHz, Standby Current 1.0 μA at 2.5 V

Table 3-6. 25CSM04 Connections

25CSM04 Pin	AVR128DB48 Pin	Function	Shared Functionality
CS	PE3	SPI0 SS	—
SO	PA5	SPI0 MISO	Feather Connector
SI	PA4	SPI0 MOSI	
SCK	PA6	SPI0 SCK	

3.2.4 32.768 kHz Crystal

The [VMK3-9002-32K7680000](#) is a 32.768 kHz crystal accurate to within 20 ppm. Using the External 32.768 kHz Crystal Oscillator (XOSC32K) module in AVR128DB48 with the mounted 32.768 kHz crystal generates a clock signal that is significantly more accurate than what the internal OSC32K module can generate alone. With the AVR128DB48's Auto-Tune feature, the external crystal can be used as a reference to improve the accuracy of the internal high-frequency oscillator, which is also an ideal reference clock for the Real-Time Counter module in the AVR128DB48.

Table 3-7. 32.768 kHz Crystal Connections

AVR128DB48 Pin	Function	Shared Functionality
PF0	XTAL32K1 (Crystal input)	—
PF1	XTAL32K2 (Crystal output)	—

3.2.5 LEDs

Five user LEDs are available on the AVR-IoT Cellular Mini board that can be controlled by either GPIO or PWM. The LEDs can be activated by driving their connected I/O lines to GND.

Table 3-8. LED Connections

LED	AVR128DB48 Pin	LED Color	Suggested Function	Shared Functionality
LED0	PA0	Blue	Cellular status	—
LED1	PA1	Green	Connection status	—
LED2	PA2	Yellow	Data status	—
LED3	PA3	Red	Error	—
LED4	PB2	Yellow	User LED	On-board debugger

3.2.6 Mechanical Switches

The AVR-IoT Cellular Mini board has two mechanical switches. SW0 is a generic user-configurable switch that will drive the connected I/O line to ground (GND) when pressed. SW1 is connected to an input-only pin on the AVR128DB48 and can be fused to function as `RESET`. There are no external pull-up resistors for the mechanical switches. Only external series resistors are present to prevent contention. It is recommended to enable internal pull-ups for PD2 and PF6.



Info: PF6 is by default fused to function as input. Set the RSTPINCFG fuse high to use SW1 as `RESET`.

Table 3-9. Mechanical Switch Connections

Switch	AVR128DB48 Pin	Function	Shared Functionality
SW0	PD2	GPIO / EXT_INT	On-board debugger
SW1	PF6	Input/RESET	—

3.2.7 GM02S Module

Sequans [GM02S](#) is a low-power consumption cellular module suitable for small form factor designs. The Monarch 2 GM02S is Sequans' Single-SKU™ RF front end and this enables the use of the module in any band worldwide. An integrated EAL5+ secure enclave enables secure storage of SIM (iSIM).

The module is powered from the VCC_P2V5_P5V25 net and outputs 1.8V on the VCC_CELL_P1V8 net, which can be used to provide power for small devices up to 100mA maximum current draw. The VCC_CELL_P1V8 is routed as power for the eSIM footprint.



The I/O for the GM02S operates at 1.8V, with a maximum input of 3.3V. Any voltage exceeding 3.3V may damage the GM02S module.

The GM02S module's power consumption varies depending on the state of the module:

- Active/Standby mode – 30 mA
- Light sleep mode – 200 µA
- Deep sleep mode – 1 µA

Table 3-10. GM02S Connections

GM02S Pin	Connection	Description
12 - Status LED/GPIO1	Testpoint - TP204	Status LED
13 - Power Save Status/GPIO2	Testpoint - TP205	Power Status
16 - RING0	AVR128DB48 - PC6	UART0 ring line
39 - ESIM Data	eSIM footprint	Routed to eSIM footprint
40 - ESIM CLK		
41 - ESIM RST		
42 - SIM CLK	SIM card holder	Connected to the SIM card holder
43 - SIM RST		
44 - SIM Data		
45 - SIM Detect		Pulled high when no SIM card is present, low when SIM is present
73 - SIM VCC		1.8V supply for SIM card holder
46 - RESETEN	AVR128DB48 - PC5	Module reset signal
66 - JTAG TMS	JTAG pins are available on the bottom of the kit in a 10 pin testpoint array .	JTAG interface for the GM02S
67 - JTAG TDI		
68 - JTAG TDO		
69 - JTAG TCK		
70 - JTAG TRSTN		



Info: The GM02S JTAG pins are routed out to a 50-mil test point array on the bottom of the board.

Figure 3-3. GM02S JTAG Test point Array



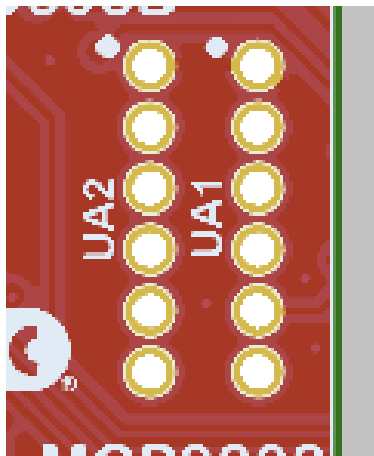
3.2.7.1 GM02S UARTs

The module features three UARTs, these are configured as shown in the table below:

Table 3-11. GM02S UARTs

GM02S UART	Connections	Function	Baud rate	Flow control	Form at	Parity	Stopb it(s)	Mode
UART0	Connected to AVR128DB48 port C	Data and control from external MCU	115200	Enabled (RTS/CTS)	8-bit	None	1	AT
UART1	Available as 50-mil test points	Debugging and firmware upgrade	921600					DCP
UART2		Modem console	115200	Unsupported				Conso le

Figure 3-4. UART1 and UART2



UART1 and UART2 are available on 50-mil test points on both the top and bottom sides of the development board.

3.2.7.2 GM02S Factory Configuration

The Sequans GM02S module is configured in the factory so that all unused GPIO pins and peripherals are disabled. Only the features available on the board are enabled. The SIM0 (SIM card) interface is enabled without polling to conserve power. The SIM card will only be powered if the modem detects a SIM card. The SIM1 interface is disabled

AVR-IoT Cellular Mini Hardware User...

Hardware Description

by default. Enable the SIM1 interface in the hardware configuration of the GM02S to use an eSIM with the AVR-IoT Cellular Mini.

All boards come with a restore point created during manufacturing of the AVR-IoT Cellular Mini. Using the factory reset functionality will reset all nonvolatile parameters of the module back to the last restoration point. Send the following commands to the GM02S via UART to perform a factory restore:

```
AT+CFUN=5
AT+SQNSFACTORYRESET
```



Issuing a factory reset also flushes any data cached by the modem and erases any certificate stored in the modem. Provisioning of cloud service provider required after a factory reset.

To check if the factory restore point is present, issue the following commands to the modems UART:

```
AT+CFUN=5
AT+SQNSFACTORYSAVE?
```

The response, if the restore point is present, is:

```
+SQNSFACTORYSAVE=microchip
```



Important: Configure the UART for AT mode to issue an AT command to any GM02S UARTs.

3.2.7.3 External Antenna

Included with the development board is a flexible antenna with a U.FL connector from [Molex](#). The antenna is suitable for any band in the 824~960/1710~2170 MHz range.

3.3 Sensors

3.3.1 Temperature Sensor

The [MCP9808](#) digital temperature sensor converts temperatures between -20°C and +100°C to a digital word with $\pm 0.25^\circ\text{C}/\pm 0.5^\circ\text{C}$ (typical/maximum) accuracy.

Additional features:

- Accuracy:
 - $\pm 0.25^\circ\text{C}$ (typical) from -40°C to $+125^\circ\text{C}$
 - $\pm 0.5^\circ\text{C}$ (maximum) from -20°C to $+100^\circ\text{C}$
- User Selectable Measurement Resolution:
 - 0.5°C , 0.25°C , 0.125°C , 0.0625°C
- User Programmable Temperature Limits:
 - Temperature Window Limit
 - Critical Temperature Limit
- User Programmable Temperature Alert Output
- Operating Voltage Range: 2.7V to 5.5V
- Operating Current: 200 μA (typical)
- Shutdown Current: 0.1 μA (typical)

The MCP9808 temperature sensor is connected to the AVR128DB48 through I²C and a GPIO for the user-configurable alert output.



Info: The 7-bit I²C address is: 0x19.

Table 3-12. MCP9808

MCP9808 Pin	AVR128DB48 Pin	Function	Shared Functionality
SDA	PF2	TWI1 SDA	Feather Connector, Qwiic
SCL	PF3	TWI1 SCL	

3.3.2 VEML3328 Color Sensor

A [VEML3328](#) RGB-CIR Color sensor is mounted on the AVR-IoT Cellular Mini board for measuring red, green, blue, clear and IR light. The sensor uses photodiodes, amplifiers and analog/digital circuitry to sense differences in color temperature and brightness.

Additional features:

- 16-bit resolution for each channel (R, G, B, C, and IR)
- Temperature compensation: -40°C to +85°C
- Operation voltage range: 2.6V to 3.6V
- Operating Current: 580 µA (typical)
- Shutdown Current: 800 nA (typical)

The VEML3328 color sensor is connected to the AVR128DB48 through an 1.8V I²C bus.



Info: The 7-bit I²C address is: 0x10.

Table 3-13. VEML3328 Connections

VEML3328 Pin	AVR128DB48 Pin	Function	Shared Functionality
SDA	PC2	TWI0 SDA	ATECC608B
SCL	PC3	TWI0 SCL	

3.4 External Connectors

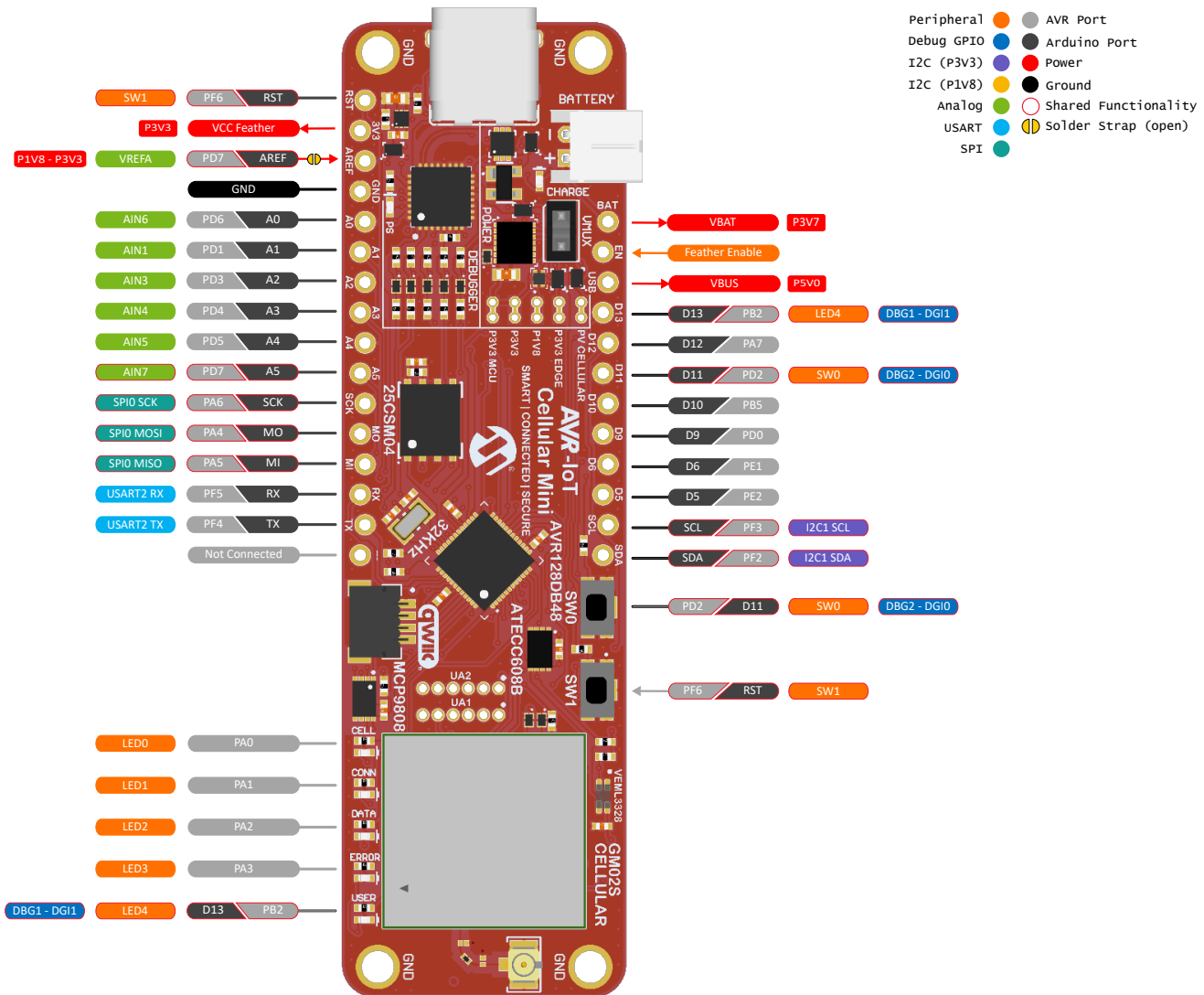
3.4.1 Feather Connector

The AVR-IoT Cellular Mini development board features a Feather style edge connector. The connector is designed to be compatible with Adafruit's [Feather/FeatherWings](#).

AVR-IoT Cellular Mini Hardware User...

Hardware Description

Figure 3-5. Feather Connector Pinout



Some connections are shared with other peripherals. See the table below:

Table 3-14. Shared connections

Pins	Description	Shared Functionality
AREF	Reference analog voltage input for AVR128DB48	Shared through open solder-strap with PD7 (A5)
SPI	—	25CSM04
I2C	—	MCP9808, Qwiic® connector
DBG1, DBG2	Debug GPIO channels	On-Board Debugger Connections

3.4.2 Qwiic®

The AVR-IoT Cellular Mini board features a 4-pin Qwiic® connector for expanding the development board using the SparkFun Qwiic® system. The Qwiic® system allows daisy-chaining of modules to connect two or more modules.

Figure 3-6. Qwiic® Connector

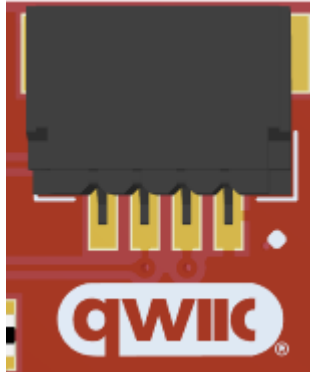


Table 3-15. Qwiic® Connector Pinout

Qwiic® Pin	AVR128DB48 Pin	Function	Shared Functionality
1 - GND	—	GND	—
2 - VCC	—	VCC_P3V3	—
3 - SDA	PF2	TWI1 SDA	MCP9808 , Feather connector
4 - SCL	PF3	TWI1 SCL	

4. On-Board Debugger

The following section describes the functionality of the on-board debugger (PKOB nano). Not all its features may be supported in the Arduino IDE.

4.1 On-Board Debugger Overview

AVR-IoT Cellular Mini 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 AVR128DB48 in both MPLAB® X IDE and Microchip Studio
- A mass storage device that allows drag-and-drop programming of the AVR128DB48
- A virtual serial port (CDC) that is connected to a Universal Asynchronous Receiver/Transmitter (UART) on the AVR128DB48, 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 AVR-IoT Cellular Mini board. The table below shows how the LED is controlled in different Operation modes.

Table 4-1. 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.

4.1.1 Debugger

The on-board debugger on the AVR-IoT Cellular Mini 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 AVR128DB48 using both MPLAB® X IDE and Microchip Studio, and some third-party IDEs.



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

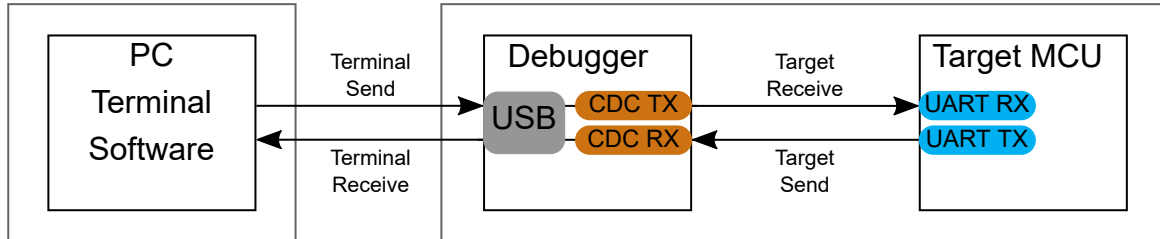
4.1.2 Virtual Serial Port (CDC)

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

4.1.2.1 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 4-1. CDC Connection



Info: As shown in [Figure 4-1](#), 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.

4.1.2.2 Operating System Support

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



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

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



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

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



Info: For all operating systems, use a terminal emulator that supports DTR signaling. See [4.1.2.4. Signaling](#).

4.1.2.3 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.

4.1.2.4 Signaling

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

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

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

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

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

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



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



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

4.1.2.5 Advanced Use

CDC Override Mode

In ordinary 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 will start with the characters:

```
CMD:SEND_UART=
```

Debugger firmware version 1.20 or earlier has the following limitations:

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

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

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

```
CMD:SEND_UART=
```

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

```
CMD:SEND_9600=
```

```
CMD:SEND_115200=
```

```
CMD:SEND_460800=
```

USB-Level Framing Considerations

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

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

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

4.1.3 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

4.1.3.1 Mass Storage Device Implementation

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

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

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

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

It is not possible to format the CURIOSITY drive. The filename may appear in the disk directory listing when programming a file to the target. 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 previously programmed application.

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

By default, the CURIOSITY drive contains several read-only files for generating icons 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
- PUBKEY.TXT – a text file containing the public key for data encryption
- 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.

4.1.3.2 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 MPLAB® X IDE or Microchip Studio.

Enabling CRC Check in Fuses

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

4.1.3.3 Special Commands

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

Table 4-2. 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:SEND_9600= CMD:SEND_115200= CMD:SEND_460800=	Sends a string of characters to the CDC UART at the baud rate specified. Note that only the baud rates explicitly specified here are supported! See “CDC Override Mode.” (Debugger firmware v1.21 or newer.)
CMD:RESET	Resets the target device by entering Programming mode and exiting Programming mode immediately afterward. The exact timing can vary according to the programming interface of the target device. (Debugger firmware v1.16 or newer.)



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

4.1.4 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 any debug GPIO data. It is available as a plug-in for MPLAB® X IDE or a stand-alone application that can be used in parallel with MPLAB® X IDE or Microchip Studio.

Although DGI encompasses several physical data interfaces, the AVR-IoT Cellular Mini implementation includes logic analyzer channels:

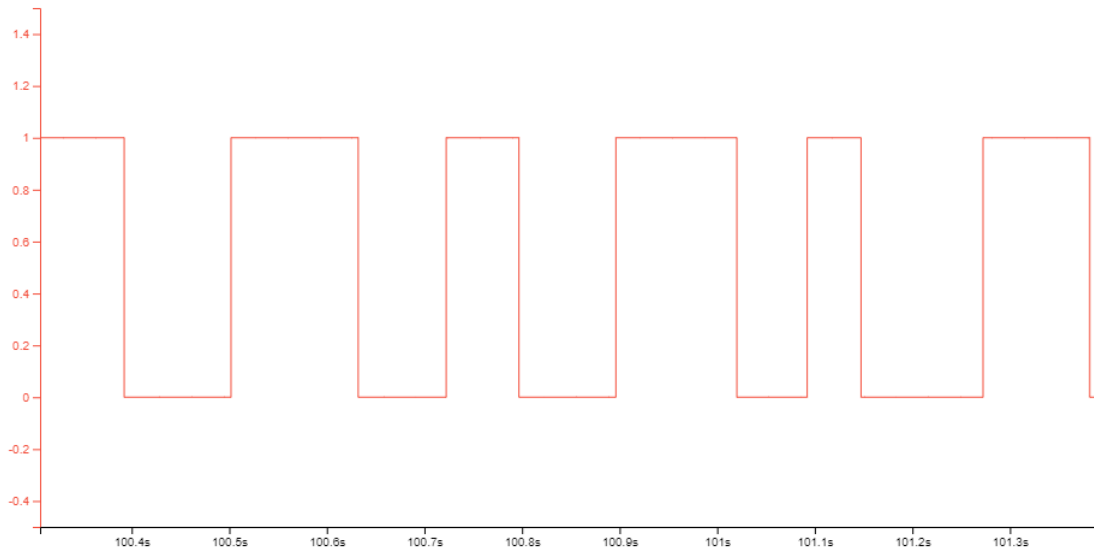
- Two debug GPIO channels (also known as DGI GPIO)

4.1.4.1 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 low-frequency events on a time axis – for example, when given Application state transitions occur.

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

Figure 4-2. 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.

4.1.4.2 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.

4.2 On-Board Debugger Implementation

AVR-IoT Cellular Mini features an on-board debugger that can be used to program and debug the AVR128DB48 using UPDI. The on-board debugger also includes a virtual serial port (CDC) interface over UART and debug GPIO. Both MPLAB® X IDE and Microchip Studio 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.

4.2.1 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 when the debugger is not using the interface. Hence, there are few 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 [4.1. On-Board Debugger Overview](#).

Table 4-3. On-Board Debugger Connections

AVR128DB48 Pin	Debugger Pin	Function	Shared Functionality
PB0	CDC RX	USART3 TX (AVR128DB48 TX line)	—
PB1	CDC TX	USART3 RX (AVR128DB48 RX line)	—
UPDI	DBG0	UPDI	—
PB2	DBG1	DGI GPIO1/LED4	3.2.5. LEDs
PD2	DBG2	DGI GPIO0/SW0	3.2.6. Mechanical Switches

5. Hardware Revision History and Known Issues

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

5.1 Identifying Product ID and Revision

There are two ways to find the revision and product identifier of the AVR-IoT Cellular Mini: Either by utilizing the MPLAB® X IDE or Microchip Studio Kit Window or by looking at the sticker on the bottom side of the PCB.

The Kit Window will pop up when connecting AVR-IoT Cellular Mini to a computer with MPLAB® X IDE or Microchip Studio running. The first six digits of the serial number, listed under kit information, contain the product identifier and revision.



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

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

The serial number string has the following format:

"nnnnrrssssssssss"

n = product identifier

r = revision

s = serial number

The product identifier for AVR-IoT Cellular Mini is A09-3437.

5.2 Revision 5

Revision 5 is the initially released revision.

GM02S delivered with modem firmware version: LR8.0.5.10-55042-cs0.

Some boards are delivered with too long pin-headers. These must be shortened before mounting to the board.

The battery charger is configured for 500 mA charge current. The AVR-IoT Cellular Mini and may use more current than the USB 2.0 specification allows.

Known issue: The crystal load capacitors are not tuned optimally. For the best results C106 may be replaced with a 6.8 pF capacitor

6. Document Revision History

Doc. Rev.	Date	Comments
A	05/2022	Initial document release

Appendix

Schematics

Figure 7-1. AVR-IoT Cellular Mini MCU Schematic

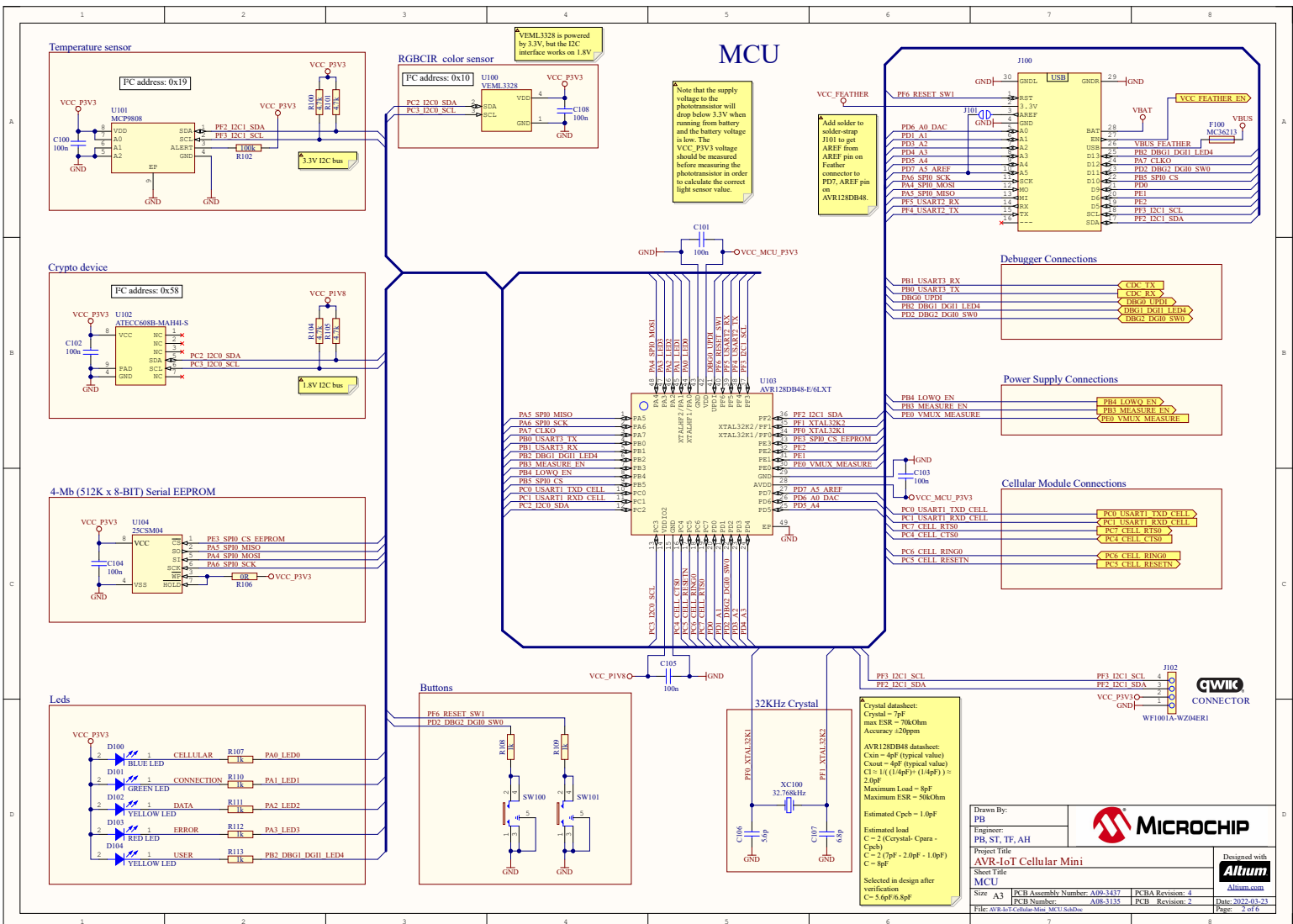


Figure 7-2. AVR-IoT Cellular Mini LTE Schematic

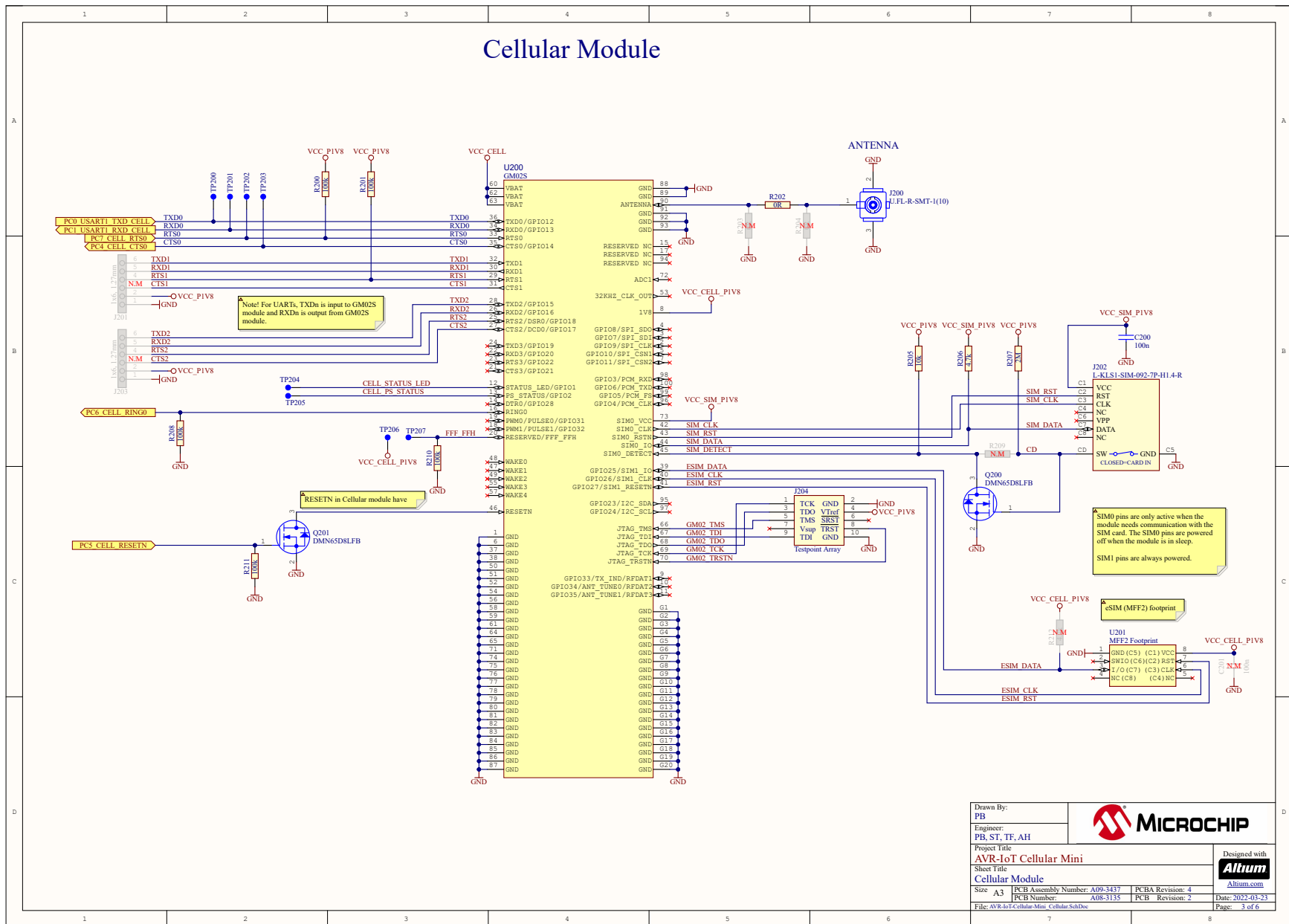


Figure 7-3. AVR-IoT Cellular Mini Power Schematic

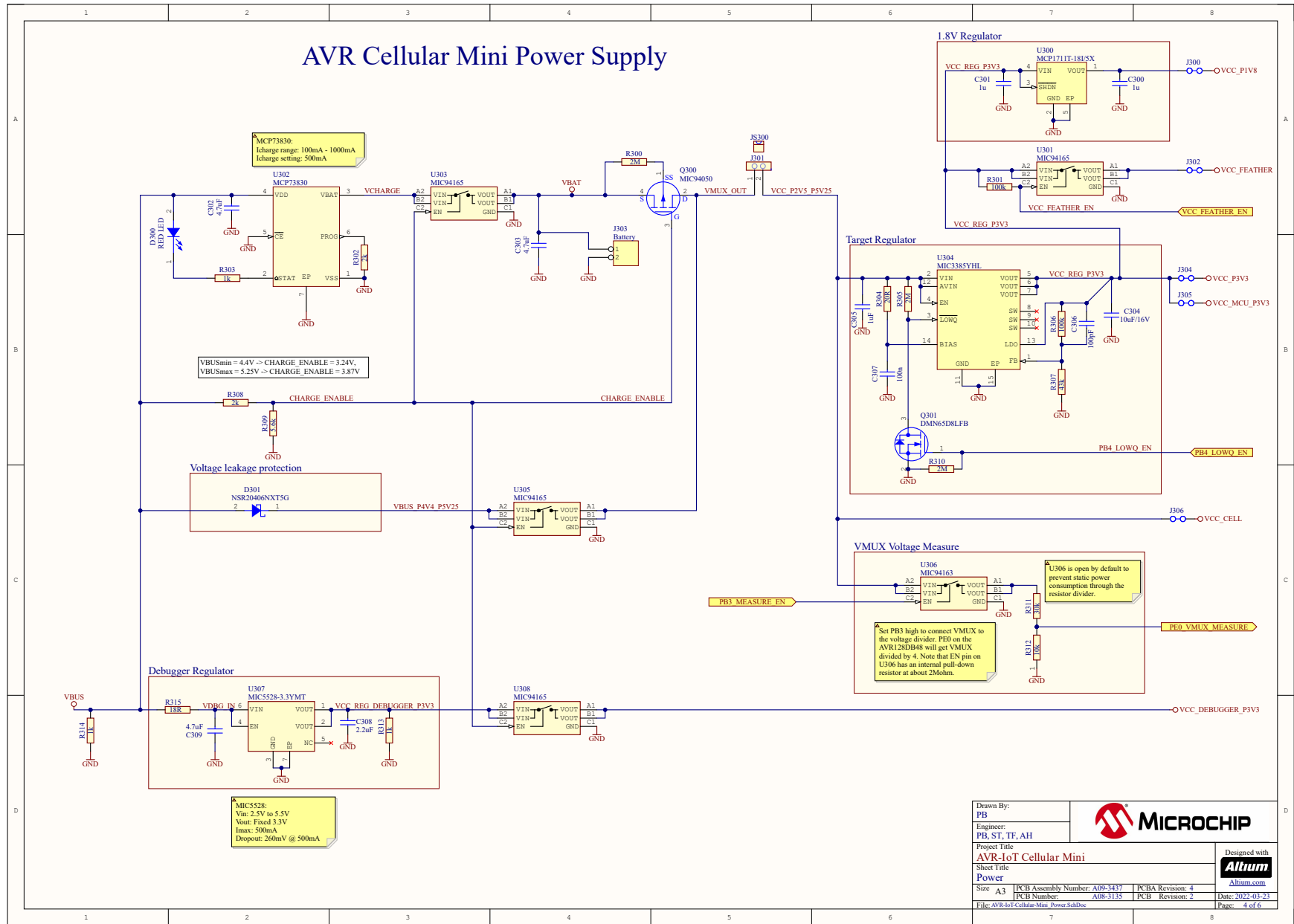
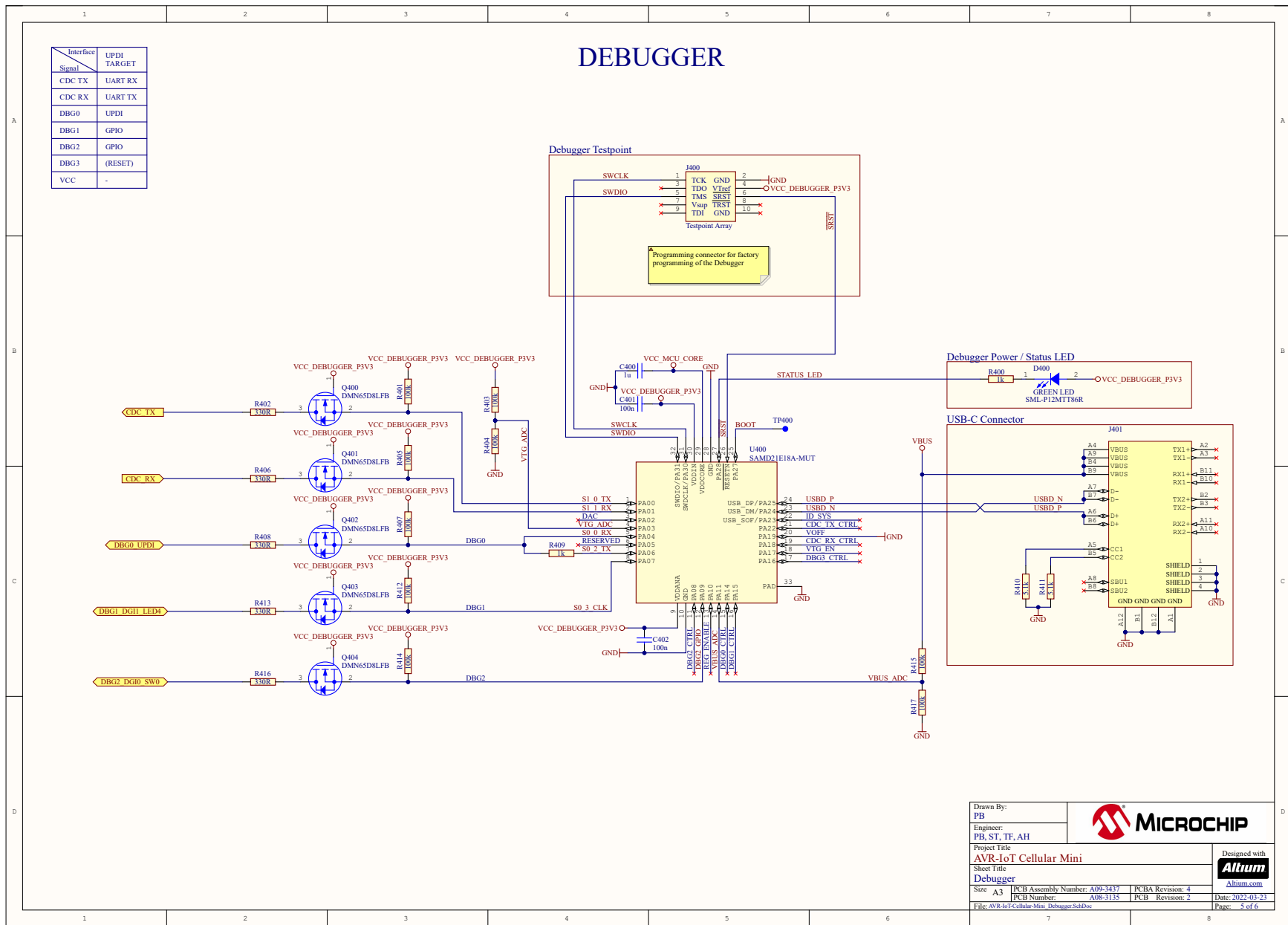


Figure 7-4. AVR-IoT Cellular Mini Debugger Schematic



7.2 Assembly Drawing

Figure 7-5. AVR-IoT Cellular Mini Assembly Drawing Top

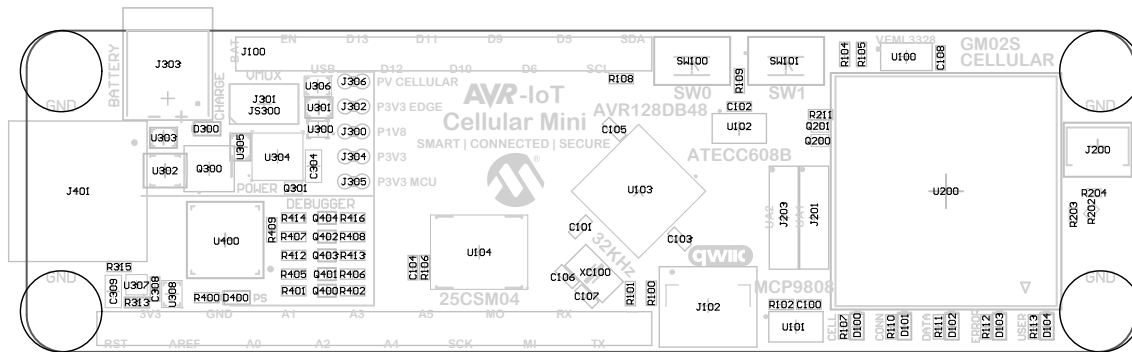
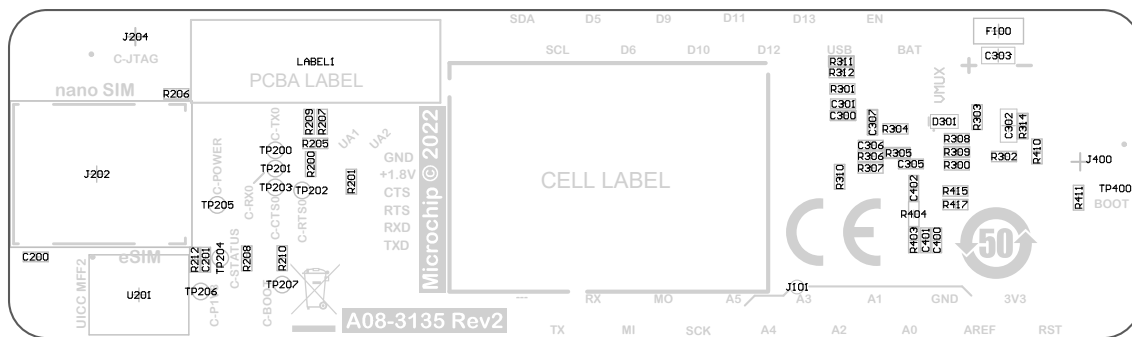


Figure 7-6. AVR-IoT Cellular Mini Assembly Drawing Bottom



7.3 AVR-IoT Cellular Mini w/ MPLAB X and Microchip Studio

The AVR-IoT Cellular Mini is compatible with both MPLAB X and Microchip Studio for bare-metal development, with the on-board debugger supporting programming and debugging in both IDEs.

7.3.1 Driver Installation

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

7.3.2 Kit Window

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



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

7.3.3 MPLAB® X IDE Device Family Packs

Microchip MPLAB® X IDE requires specific information to support devices and tools. This information is contained in versioned packs. For the AVR-IoT Cellular Mini board, MPLAB® X version 5.40 with device family pack AVR-Dx

version 1.2.52 and tool pack nEDBG_TP version 1.2.x or newer is required. For more information on packs and how to upgrade them, refer to the [MPLAB® X IDE User's guide - Work with Device Packs](#).



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

7.3.4 Microchip Studio Device Family Pack

Microchip Studio requires specific information to support devices. This information is contained in versioned packs. For the AVR-IoT Cellular Mini board, the device family pack AVR-Dx version 1.1.45 or newer is required. For more information on packs and how to upgrade them, refer to the [Microchip Studio User Guide - Device Pack Manager](#).



Tip: The latest device family packs are available through **Tools > Device Pack Manager** in Microchip Studio or online at [Microchip Studio Packs Repository](#).

Microchip Information

The Microchip Website

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

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

Product Change Notification Service

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

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

Customer Support

Users of Microchip products can receive assistance through several channels:

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

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

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

Microchip Devices Code Protection Feature

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

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded

by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

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

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

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICTail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQL, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, Symmcom, and Trusted Time 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.

© 2022, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-6683-0360-3

Quality Management System

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

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com	Australia - Sydney Tel: 61-2-9868-6733 China - Beijing Tel: 86-10-8569-7000 China - Chengdu Tel: 86-28-8665-5511 China - Chongqing Tel: 86-23-8980-9588 China - Dongguan Tel: 86-769-8702-9880 China - Guangzhou Tel: 86-20-8755-8029 China - Hangzhou Tel: 86-571-8792-8115 China - Hong Kong SAR Tel: 852-2943-5100 China - Nanjing Tel: 86-25-8473-2460 China - Qingdao Tel: 86-532-8502-7355 China - Shanghai Tel: 86-21-3326-8000 China - Shenyang Tel: 86-24-2334-2829 China - Shenzhen Tel: 86-755-8864-2200 China - Suzhou Tel: 86-186-6233-1526 China - Wuhan Tel: 86-27-5980-5300 China - Xian Tel: 86-29-8833-7252 China - Xiamen Tel: 86-592-2388138 China - Zhuhai Tel: 86-756-3210040	India - Bangalore Tel: 91-80-3090-4444 India - New Delhi Tel: 91-11-4160-8631 India - Pune Tel: 91-20-4121-0141 Japan - Osaka Tel: 81-6-6152-7160 Japan - Tokyo Tel: 81-3-6880-3770 Korea - Daegu Tel: 82-53-744-4301 Korea - Seoul Tel: 82-2-554-7200 Malaysia - Kuala Lumpur Tel: 60-3-7651-7906 Malaysia - Penang Tel: 60-4-227-8870 Philippines - Manila Tel: 63-2-634-9065 Singapore Tel: 65-6334-8870 Taiwan - Hsin Chu Tel: 886-3-577-8366 Taiwan - Kaohsiung Tel: 886-7-213-7830 Taiwan - Taipei Tel: 886-2-2508-8600 Thailand - Bangkok Tel: 66-2-694-1351 Vietnam - Ho Chi Minh Tel: 84-28-5448-2100	Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4485-5910 Fax: 45-4485-2829 Finland - Espoo Tel: 358-9-4520-820 France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 Germany - Garching Tel: 49-8931-9700 Germany - Haan Tel: 49-2129-3766400 Germany - Heilbronn Tel: 49-7131-72400 Germany - Karlsruhe Tel: 49-721-625370 Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 Germany - Rosenheim Tel: 49-8031-354-560 Israel - Ra'anana Tel: 972-9-744-7705 Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781 Italy - Padova Tel: 39-049-7625286 Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340 Norway - Trondheim Tel: 47-72884388 Poland - Warsaw Tel: 48-22-3325737 Romania - Bucharest Tel: 40-21-407-87-50 Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 Sweden - Gothenberg Tel: 46-31-704-60-40 Sweden - Stockholm Tel: 46-8-5090-4654 UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820

Mouser Electronics

Authorized Distributor

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

[Microchip:](#)

[EV70N78A](#)