



PIC-IoT WG User Guide

PIC-IoT WG Development Board User Guide

Preface

The PIC-IoT WG development board is a small and easily expandable demonstration and development platform for IoT solutions, based on the PIC[®] microcontroller architecture using Wi-Fi[®] technology. It was designed to demonstrate that the design of a typical IoT application can be simplified by partitioning the problem into three blocks:

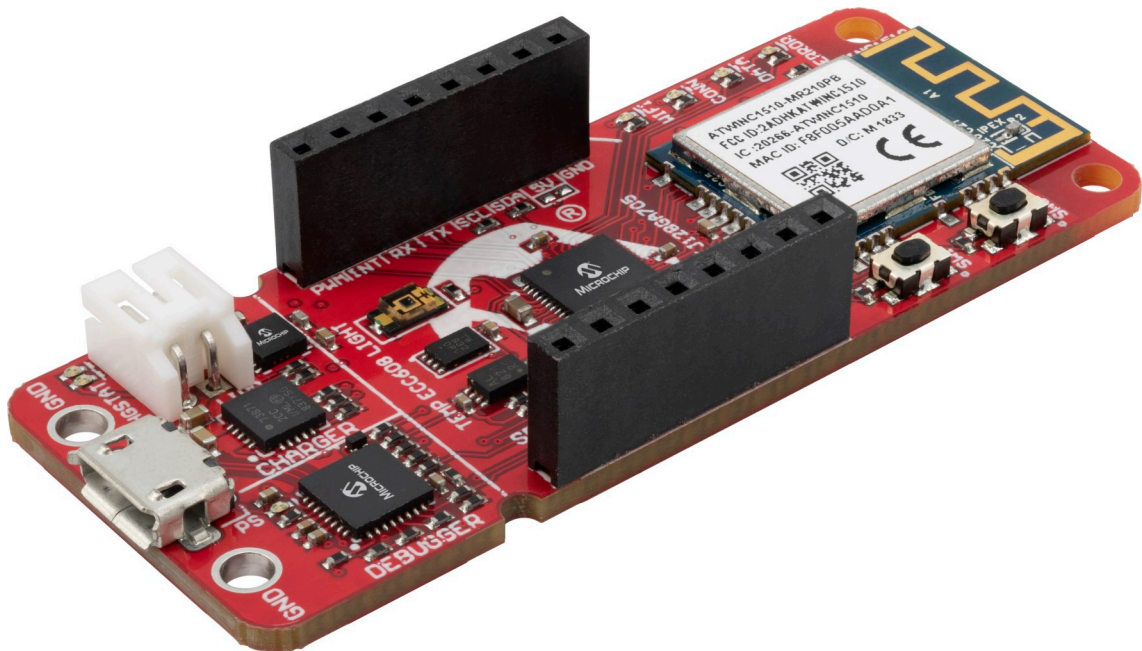
- Smart - represented by the PIC24FJ128GA705 microcontroller
- Secure - represented by the ATECC608A secure element
- Connected - represented by the WINC1510 Wi-Fi controller module

The PIC-IoT WG development board feature the following elements:

- The PKOB nano provides access to a serial port interface (serial to USB bridge)
- The PKOB nano enumerates on the PC as a mass storage interface device for easy 'drag and drop' programming, Wi-Fi[®] configuration, and full access to the microcontroller application Command Line Interface (CLI)
- An interface allowing for full programming and debugging support through MPLAB[®] X IDE
- mikroBUS[™] connector allows for the ability to expand the board capabilities with selection from 450+ sensors and actuator options offered by MikroElektronika (www.mikroe.com) via a growing portfolio of Click boards[™]
- A light sensor used to demonstrate published data
- A high-accuracy temperature sensor used to demonstrate published data

The PIC-IoT WG development board comes pre-programmed and configured for demonstrating the connectivity to the Google Cloud IoT Core.

Figure 1. PIC-IoT WG



The MPLAB® Code Configurator (MCC) with supporting PIC-IoT WG library can be used for generating the firmware through MPLAB X®. To download/modify the firmware, the system requirements are the following:

Table 1. Software Requirements

Required for	PIC-IoT
IDE	MPLAB X IDE v 5.15 or later
Compiler	XC16 Compiler v 1.35 or later
Code Generation	MCC Plug-in v 3.75
	MCC Foundation Services v 0.1.32
	MCC PIC-IoT WG Sensor Node v 1.1.1

Table of Contents

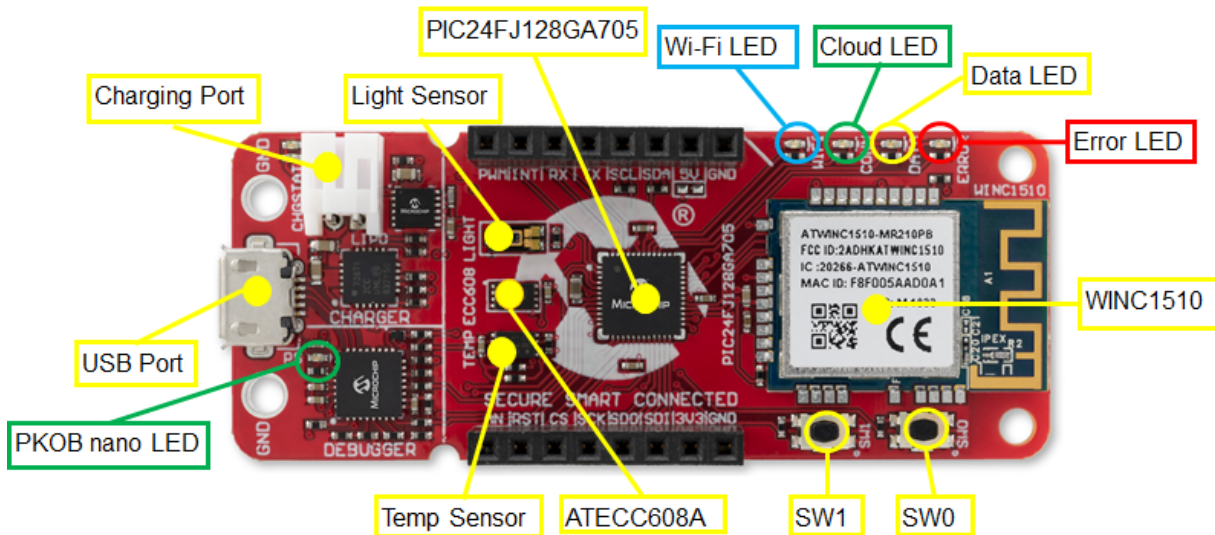
Preface.....	1
1. Chapter 1: Overview.....	5
1.1. Board Layout.....	5
1.2. LED Indicators.....	5
1.3. Switch Button Use Cases.....	6
2. Chapter 2: Getting Started.....	7
2.1. Connecting the Board to the Host PC.....	7
2.2. The MCHP-IoT Webpage.....	8
2.3. Connecting the Board to Wi-Fi Networks.....	9
2.4. Visualizing Cloud Data in Real Time.....	11
2.5. Configuring Other Settings.....	13
3. Chapter 3: Code Generation.....	16
3.1. Generating code from MCC.....	16
4. Chapter 4: Hardware Guide.....	26
4.1. Technical Features.....	26
4.2. Hardware Description.....	26
4.3. Hardware Revision.....	27
5. FAQs, Tips and Troubleshooting.....	29
5.1. FAQs and Tips.....	29
5.2. LED Status Troubleshooting.....	30
6. Appendix.....	33
6.1. Relevant Links.....	33
6.2. Board Layouts.....	33
7. Document Revision History.....	36
The Microchip Web Site.....	37
Customer Change Notification Service.....	37
Customer Support.....	37
Microchip Devices Code Protection Feature.....	37
Legal Notice.....	38
Trademarks.....	38
Quality Management System Certified by DNV.....	39

Worldwide Sales and Service.....	40
----------------------------------	----

1. Chapter 1: Overview

1.1 Board Layout


The PIC-IoT WG development board layout can be seen below.






1.2 LED Indicators

The development board features four LEDs that the demo code uses to provide diagnostic information as represented in the table below.

Table 1-1. LED Indicators

LED Color	Type		Indication	Details
	Label	Pattern		
 Blue	WIFI	Solid Blue	Wi-Fi Network Connection	Indicates a successful connection to the local Wi-Fi network.
		Blinking Blue	Soft AP Mode	Indicates that the board can be detected and used as a Wi-Fi access point. For details refer to the 2.3.3 Via Soft AP section.

.....continued				
LED Color	Type		Indication	Details
	Label	Pattern		
 Green	CONN	Solid Green	Google Cloud Connection	Indicates a successful MQTT connection to the Google Cloud .
		Blinking Green	Default Wi-Fi credentials	Indicates that the board is trying to establish a Wi-Fi connection using the default credentials.
 Yellow	DATA	Blinking Yellow	Data Publication to the Cloud	Indicates that sensor data in the form of MQTT packet has been successfully published to the Google Cloud.
		Solid Yellow for extended time	State of “Toggle” sent within MQTT publish packet	Indicates the state of the “Toggle” switch, received as part of the packet published by Google Cloud on the subscribed topic.
 Red	ERROR	Solid Red	Error Status	Indicates an error in the application.

1.3 Switch Button Use Cases

The following section gives details on hold buttons at power-up:

- Hold SW0 until two LED cycles to enter Soft AP mode (Refer to the [2.3.3 Via Soft AP](#) section).
- Hold both SW0 and SW1 to use default Wi-Fi credentials. The default credentials are configurable through MCC, and the application uses the following default values:
-

Table 1-2. Wi-Fi Credentials

SSID	Password
MCHP.IOT	microchip

2. Chapter 2: Getting Started

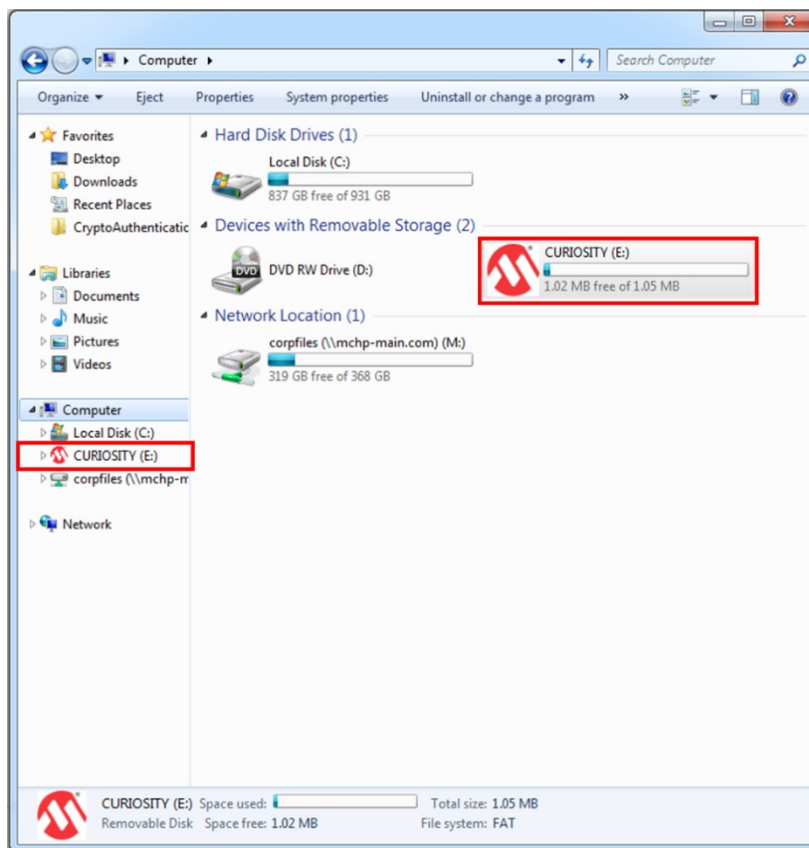
2.1 Connecting the Board to the Host PC

The PIC-IoT boards can be connected to a computer using a standard micro-USB cable. Once plugged in, the LED array at the top right-hand corner of the board should flash in the following order twice: Blue->Green->Yellow->Red. When the board is not connected to Wi-Fi, the red LED will light up.

The board will appear as a removable storage device on the host PC, as shown in the figure below. Double-click the **CURIOSITY** drive to open it and get started.

Note: All procedures are the same for Windows®, Mac OS®, and Linux® environments.

Figure 2-1. Curiosity Board as Removable Storage



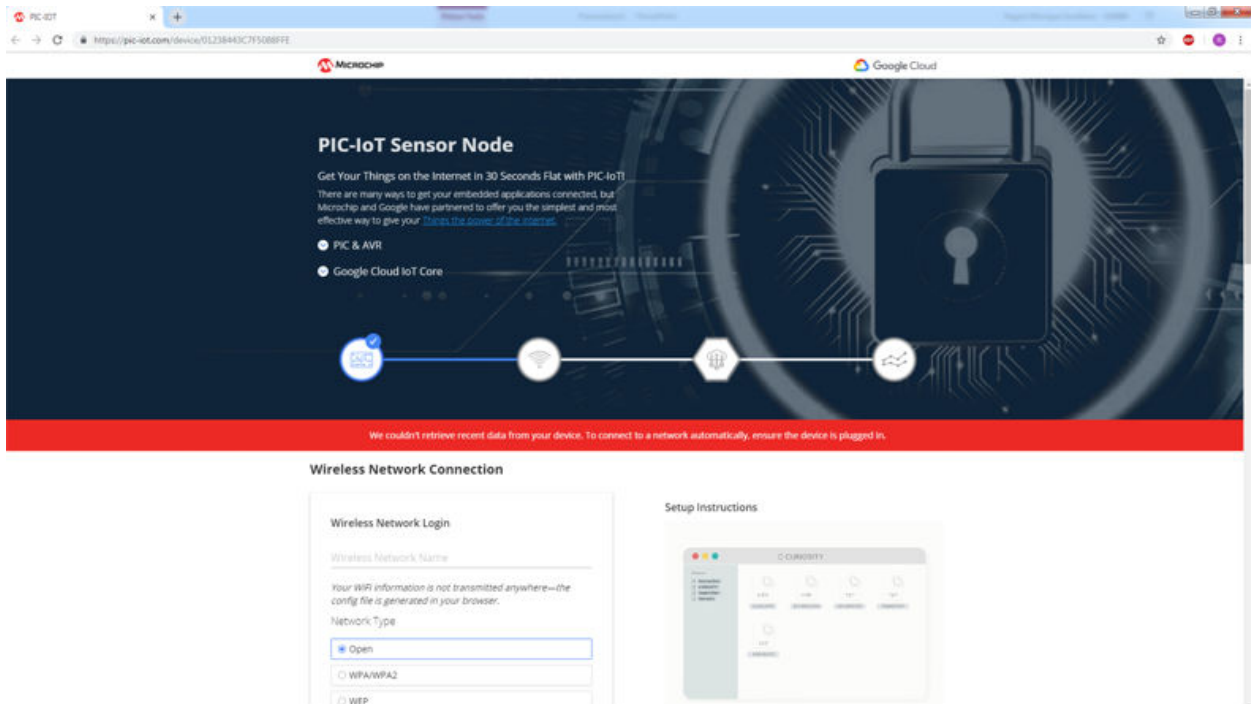
The CURIOSITY drive should contain the following five files:

- CLICK-ME.HTM - redirects the user to the PIC-IoT web demo application
- KIT-INFO.HTM- redirects the user to a site containing information and resources about the board
- KIT-INFO.TXT - a text file with details about PKOB nano firmware and the board's serial number
- PUBKEY.TXT - a text file containing the public key used for data encryption
- STATUS.TXT - a text file containing the status condition of the board

Double-click on the **CLICK-ME.HTM** file to go to the dedicated webpage to access the Google Cloud sandbox account.

2.2 The MCHP-IoT Webpage

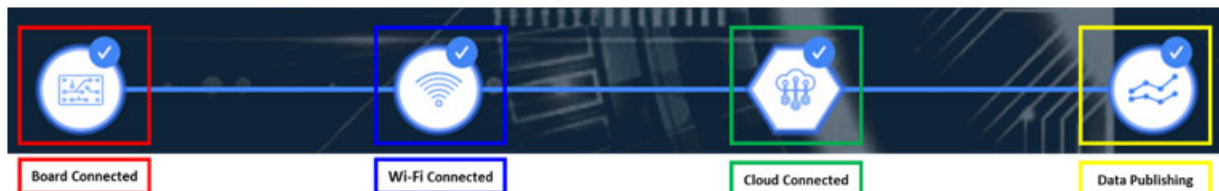
Figure 2-2. PIC-IoT WG Webpage



The figure below shows an image of the PIC-IoT WG webpage. This page displays the sensor data and allows the user to regenerate the Wi-Fi credentials as a file labeled WIFI.CFG. This can be loaded onto the board acting as a storage device to re-configure the access point parameters.

The status markers at the middle of the page, as shown in the following figure, indicate the progress of the system setup. These markers will light up once each stage is completed successfully.

Figure 2-3. Webpage Status Indicators



The leftmost marker indicates if the board is connected to the host PC. Next to this, the Wi-Fi marker lights up once the board is connected to a Wi-Fi network. The blue LED will turn on to indicate the board connection state. To the right of the Wi-Fi marker, the Google Cloud MQTT marker is found, indicating the status of the TCP socket connection and MQTT connection to the Google Cloud. The corresponding green LED will turn on to indicate the board connection state. Finally, the rightmost marker lights up signifying that data is streaming from the board to the server; this is shown by the blinking of the yellow LED on the board for each successful MQTT publication of data.

2.3 Connecting the Board to Wi-Fi Networks

2.3.1 Via PIC-IoT Webpage

There are several ways to connect the PIC-IoT development board to the Internet. The easiest of these methods is through the PIC-IoT webpage (www.pic-iot.com). The lower left-hand corner of the site will show a wireless network connection window where the user can choose to connect to an open (no password required) network or enter the credentials for a password protected (WPA/WPA2/WEP) Wi-Fi network. The figure below shows how to enter the Wi-Fi credentials on the website.



Important: The Wi-Fi network SSID and password are limited to 19 characters. Avoid using quotation marks, names, or phrases that begin or end in spaces. The PIC-IoT development board supports only 2.4 GHz networks inline, thus using mobile hotspots to connect the board to the Internet is recommended.

Figure 2-4. Entering Wi-Fi Credentials in PIC-IoT Webpage

Once the required details are entered, click the **Download Configuration** button. This will download the WIFI.CFG (text) file to the host PC. From the WIFI.CFG's download location, drag and drop the file to the CURIOSITY drive to update the Wi-Fi credentials of the board. The blue LED will light up once a successful connection to the Wi-Fi Access Point is made. Refer to [Chapter 3](#) to troubleshoot any board issues.



Important: Any information entered in the SSID and password fields is not transmitted over the web or to the Microchip or Google servers. Instead, the information is used locally (within the browser) to generate the WIFI.CFG file.

2.3.2 Via Command Line Interface (CLI)

Another way of connecting to the Wi-Fi is through the Serial Command Line Interface (CLI). This interface can be accessed through any serial terminal application. Using the UART settings defined in the [2.5.2 Serial USB Interface](#) section, the user can reconfigure the board to a Wi-Fi network by entering the `wifi` command. [Figure 2-5](#) and [Figure 2-6](#) show examples of trying to connect to open, or secured networks, respectively. For more details on the `wifi` command and its parameters, refer to the [2.5.2 Serial USB Interface](#) section.

Figure 2-5. Wi-Fi Configuration via Serial Command Line (Open Network)

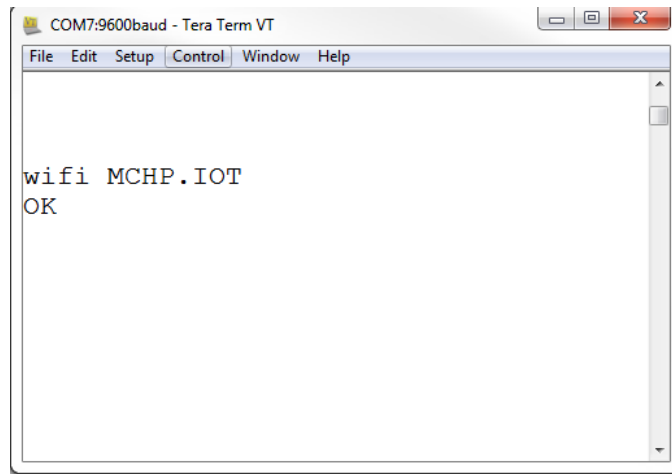
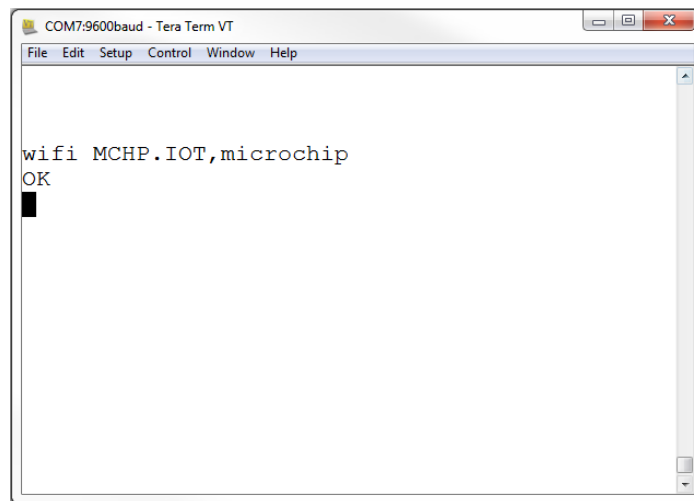


Figure 2-6. Wi-Fi Configuration via Serial Command Line (Secured Network)



2.3.3 Via Soft AP

The last method to connect to the Wi-Fi is through the advanced Software Access Point (Soft AP) mode, which is a feature of the WINC module on board. This method is ideal if the user is only using a mobile device, such as a mobile phone or tablet, instead of a laptop or PC. The Soft AP mode can be entered by pressing and holding the SW0 push button for most of the start-up time between initial power-up LED cycling. When the Soft AP mode has been successfully entered, the board can be detected as a Wi-Fi access point named MCHP.IOT.ACCESSPOINT; the blue LED will begin to blink when Soft AP is available. Using a mobile device such as a mobile phone or tablet, connect to the MCHP.IOT.ACCESSPOINT hotspot. It will redirect to a sign-in page where the user can enter the SSID and password of the network to which the board will connect. The device name will not be considered,

and the authorization type will always be WPA/WPA2 (2). Once these details are entered, click the **Connect** button to connect the board to the network. The figure below shows how the sign-in page will look.

Figure 2-7. Connecting via Soft AP

2.4 Visualizing Cloud Data in Real Time

Out of the box, all PIC-IoT development boards are pre-registered to Microchip's Google Cloud sandbox account. This account is set up for demonstration purposes only. All data gathered by the sensors of the PIC-IoT WG development boards are published to the Microchip sandbox account and can be identified by the following details:

Table 2-1. Project details

Project ID	Region
pic-IoT	us-central1

There is no permanent storage, or collection of the data published by the boards connected to the Microchip sandbox account. The full storage catalog of the Google Cloud features, such as data storage/retention, can be available to the user with use of the board once removed from the demo environment and the associated Device ID/Public Key has been migrated to a private account.

2.4.1 Publishing Data to the Google Cloud

A MQTT PUBLISH packet is always sent to the MQTT broker using a specific topic. The PIC-IoT development board publishes messages using the topic '/devices/{deviceId}/events' in communication to the Google Cloud. The messages published on this topic contain the real-time data obtained from the on-board light and temperature sensors. It does not perform any averaging of data, which is done to allow instantly visible changes on the webpage. The frequency of sending a PUBLISH packet can be decided

by the user application. The application is written such that the sensor data is published to the Cloud every one second.

2.4.1.1 Viewing the published messages

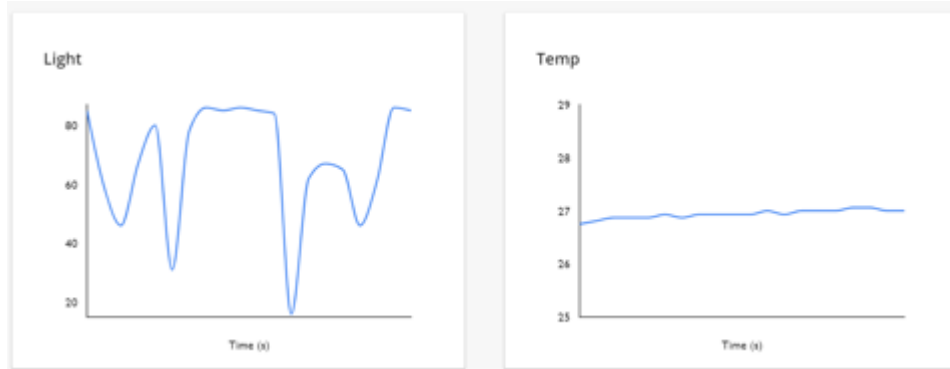
Once the board is connected to a Wi-Fi access point, and has established a socket connection to the Cloud, the PIC-IoT webpage will show a real-time graph of the data captured from the on-board light and temperature sensors. Data is sent as a MQTT PUBLISH packet from the board to the cloud through a JSON object.

The ASCII string is formatted as follows:

```
{'Light': XXX, 'Temp': YYY },
```

where XXX and YYY are numerical values expressed in decimal notation. The yellow LED on the board is turned on for 250 ms, every one second to indicate that the board is publishing data.

Figure 2-8. Real-Time Data on the PIC-IoT Webpage



2.4.2 Subscribing to topics

In addition to publishing its own data, the PIC-IoT development boards are also capable of subscribing to a topic, after which it will receive data from the Google Cloud whenever data with that topic is published to the broker server. Subscribing to topics is desired when the receiver is interested in the information sent to the broker by other connected client devices publishing data using the subscribed topic. After sending a SUBSCRIBE packet, all the messages published on the specific topic of subscription are received by the board. As of now, the board subscribes to the '/devices/{deviceId}/config' topic. This is the only topic provided by the Google Cloud for subscribing using the MQTT connection.

2.4.3 Sending the messages

The pic-iot.com webpage URL displays a section "Control your device" below the Light and Temperature graphs. The **Toggle** button is used to send the switch value to the PIC-IoT board.

Similarly, the "Text" section can be used to send a text string to the board. The toggle and text fields can be modified individually or at the same time. These values are only published over '/devices/{deviceId}/config' topic upon pressing the **Send to device** button. Since the board subscribes to this topic by default, all the published messages are received by the board.

Figure 2-9. Sending Messages on the Subscribed Topic

Control your device

If your device is listening, you can use these controls to send it a base64 encoded JSON message, with any or all of these fields:

Toggle

Sends values of 0 or 1 to device as "toggle" field

☒

Text

Sends string to device as "text" field

microchip

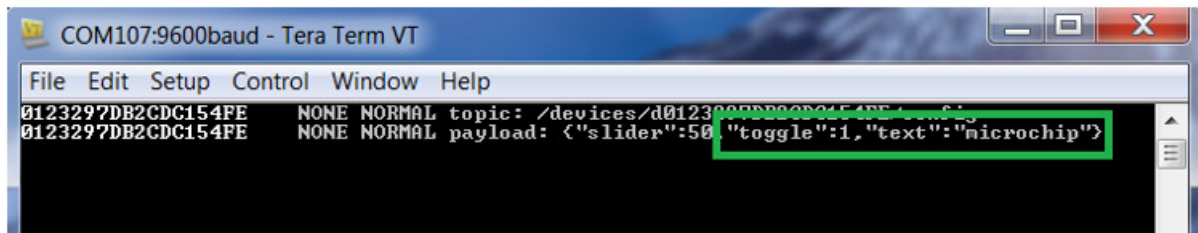
Send to device

2.4.4 Viewing Messages Received on Subscribed Topic

The toggle switch value corresponds to a short forced ON/OFF state to the yellow LED on the PIC-IoT board. The LED will stay ON/OFF for a short time depending on the position of the toggle switch. After, the LED will resume normal behavior, blinking to indicate the transmission sensor data through PUBLISH packets.

The message typed in the text field is transmitted in the form of a string to the board. In addition to the yellow LED behavior, the values of the toggle and text field values can be viewed on a Serial Terminal application (such as Tera Term, Realterm, PuTTY, etc.).

Figure 2-10. Viewing Messages on a Serial Terminal



There is no permanent storage, or collection of the data published by the boards connected through the Microchip sandbox account. The full storage features available by the Google Cloud are available to the user after the board has been removed from the demo environment (Microchip Sandbox) and migrated to a private account.

2.5 Configuring Other Settings

While the PIC-IoT development board comes out of the box fully programmed and provisioned, the user can still control aspects of the application firmware behavior through the USB interface. There are three methods to do this: Hex File (reprogram) or WIFI.CFG (reconfigure credentials) drag and drop using the mass storage feature, commands through the serial command line interface (CLI), or using MPLAB X[®] IDE, and the on-board programmer/debugger PKOB nano.

2.5.1 Mass Storage Drag and Drop

One way to program the embedded device is to drag and drop a .hex file into the CURIOSITY drive. The C compiler tool chain generates a .hex file for each project it builds. This .hex file contains the code of

the project. The Nano Embedded Debugger (PKOB nano) also provides access to a serial port interface (serial to USB bridge). This facilitates the user to drag and drop a modified `.hex` file which contains the firmware updates. This feature does not require any USB driver to be installed and works in all major OS environments. Alternative application `'example.hex'` files for the board firmware will be available for selection from the downloads section at the bottom of the PIC-IoT webpage as they become available.

2.5.2 Serial USB Interface

The Wi-Fi Access Point credentials can be re-configured through a serial command line interface on PIC-IoT development boards. This interface may also be used to provide application diagnostic information. To access this interface, use any preferred serial terminal application (i.e. Teraterm, Coolterm, PuTTY) and open the serial port labeled Curiosity Virtual COM port, with the following settings:

Table 2-2. Serial USB Interface Settings

Baud Rate	9600
Data	8-Bit
Parity Bit	None
Stop Bit	1 Bit
Flow Control	None
Local Echo	ON
Transmit Protocol	CR+LF (Carriage Return + Line Feed)

Note: For users of the Windows environment, the USB serial interface requires the installation of a USB serial port driver, included in the installation of the MPLAB X IDE.

The user can control the board by typing the command keywords, listed in the table below:

Table 2-3. Serial Command Line Commands

Command	Arguments	Description
<code>reset</code>	-	Reset the settings on the device
<code>device</code>	-	Print the unique device ID of the board
<code>key</code>	-	Print the public key of the board
<code>reconnect</code>	-	Re-establish connection to the Cloud
<code>version</code>	-	Print the firmware version of the serial port user interface
<code>cli_version</code>	-	Print the command line interface firmware version of the serial port user interface
<code>wifi</code>	<Network SSID>, <Password>, <Security Option*>	Enter Wi-Fi network authentication details
<code>debug</code>	<Debug Options**>	Print debug messages to see status of board operation

*- Authorization Type options are available by typing one of the following three numbers to determine network security option used:

1. Open - Password and Security option parameters are not required.
2. WPA/WPA2 - Security Option Parameter not required.
3. WEP - Network Name, Password, and Security Option (3) Parameter are required when connecting to a WEP network. e.g. 'wifi MCHP.IOT,microchip,3'

** - Debug Severity options are available to `debug_printer()` by using a severity number from zero to four:

0. Normal – Only SEVERITY_NONE messages are printed. In the application this is a print out of the payload received over the Subscribed Topic

1. Warning – SEVERITY_WARNING and under messages are printed*

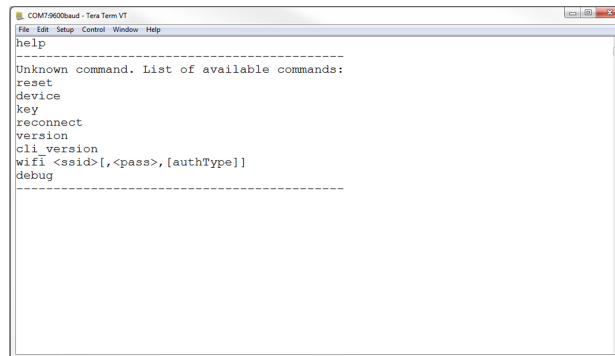
2. Notice – SEVERITY_NOTICE and under messages are printed*

3. Info - SEVERITY_INFO and under messages are printed*

4. Debug - SEVERITY_DEBUG and under messages are printed. There are multiple ERROR handling assist messages in place using this SEVERITY

Note: *There are NO messages of this type in the IoT WG development board application.

Figure 2-11. Serial Command Line Interface



2.5.3 On-board Programmer/Debugger Interface

For users familiar with the MPLAB X IDE, the PIC-IoT boards can also be programmed, and/or debugged directly via these IDEs standard operations. The PIC-IoT development boards are automatically detected by the MPLAB X, enabling full programming and debugging through the on-board PKOB nano interface. For code generation, see [Chapter 3](#) on how to generate a sample application code in MCC.

3. Chapter 3: Code Generation

3.1 Generating code from MCC

The source code of the PIC-IoT WG development boards is available using the MPLAB Code Configurator (MCC) in MPLAB X IDE.

3.1.1 Generating the demo

Once the board is connected to the host machine and MPLAB X is launched, follow these steps to generate microcontroller code for it.

3.1.1.1 Generating the demo


1. Create a new Standalone project (see the [Figure 3-1](#)) in MPLAB X 5.15 using the PIC24FJ128GA705 as device (see the [Figure 3-2](#)); the PKOB nano as programming tool (see [Figure 3-3](#)); and the XC16 as compiler (see [Figure 3-4](#)). Finally, name the MPLAB project and its location (see [Figure 3-5](#)). The Start page will then appear.
2. On the MPLAB X toolbar, look for and click the MCC Icon () or click Tools>Embedded>MPLAB X Code Configurator v3 Open/Close.
3. Under Device Resources, scroll down to the 'Internet of Things' header. Under Examples, double-click on **PIC-IoT WG Sensor Node** (see [Figure 3-6](#)).

Figure 3-1. New Project

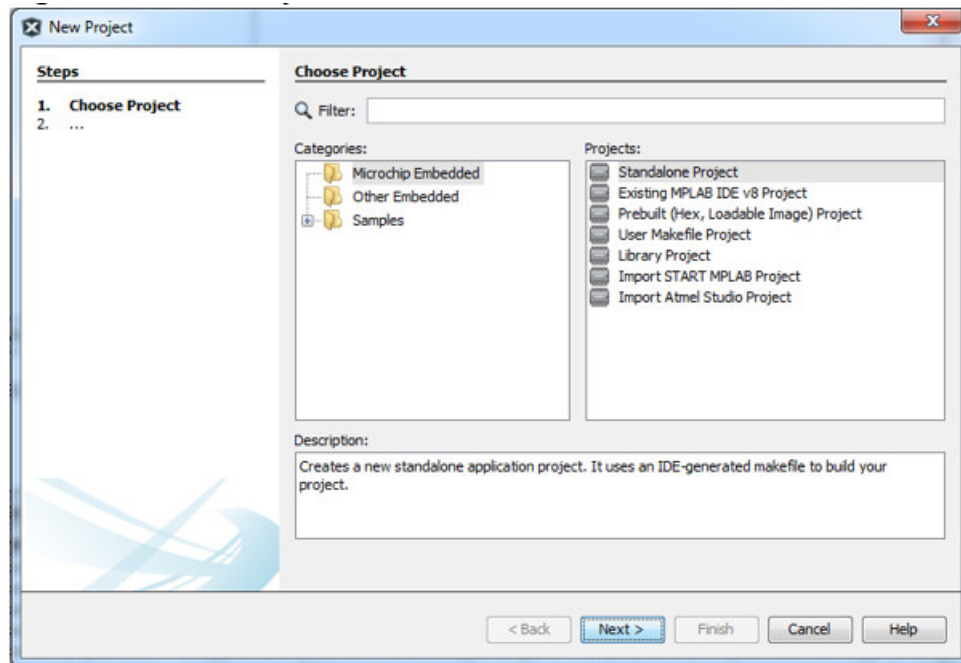


Figure 3-2. Selecting a Device

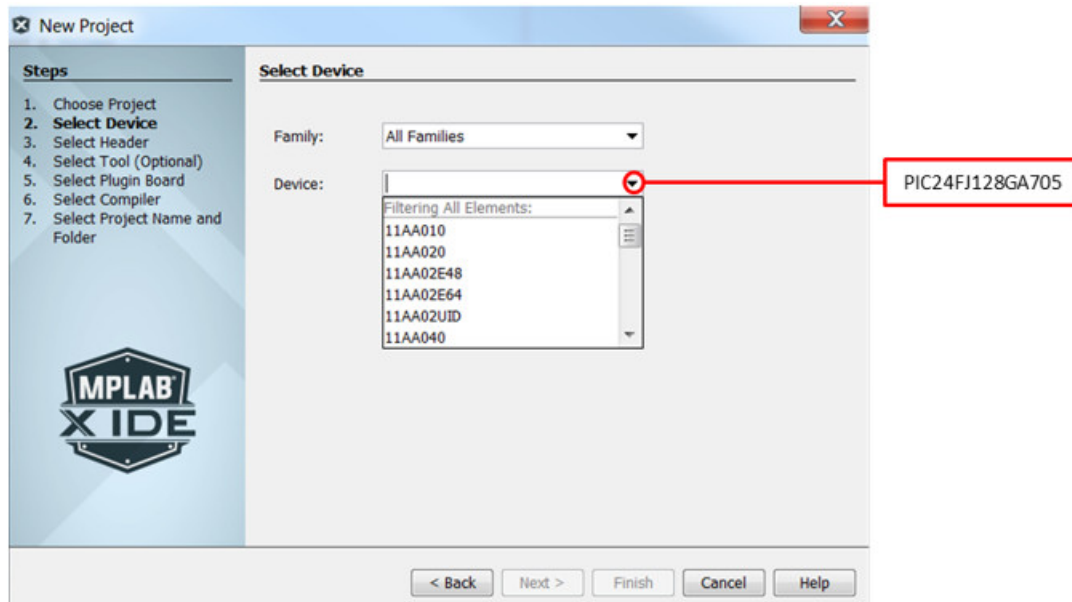


Figure 3-3. Selecting a Programmer

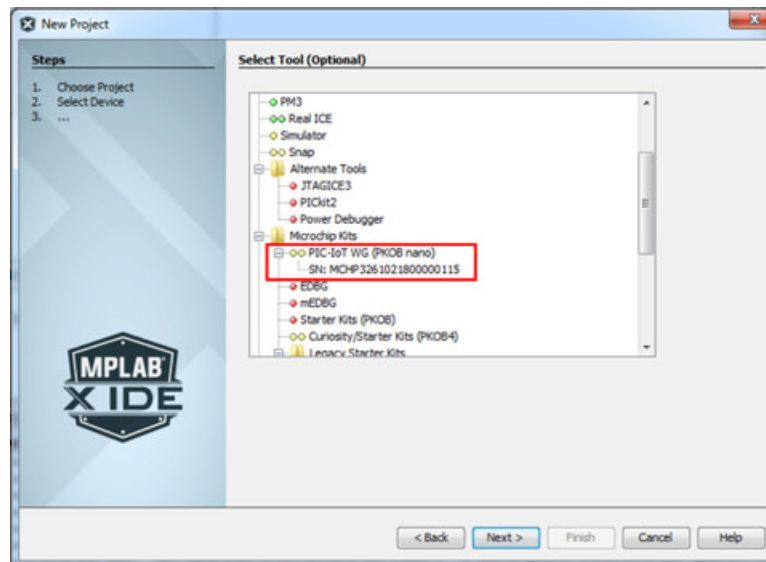


Figure 3-4. Selecting a Compiler

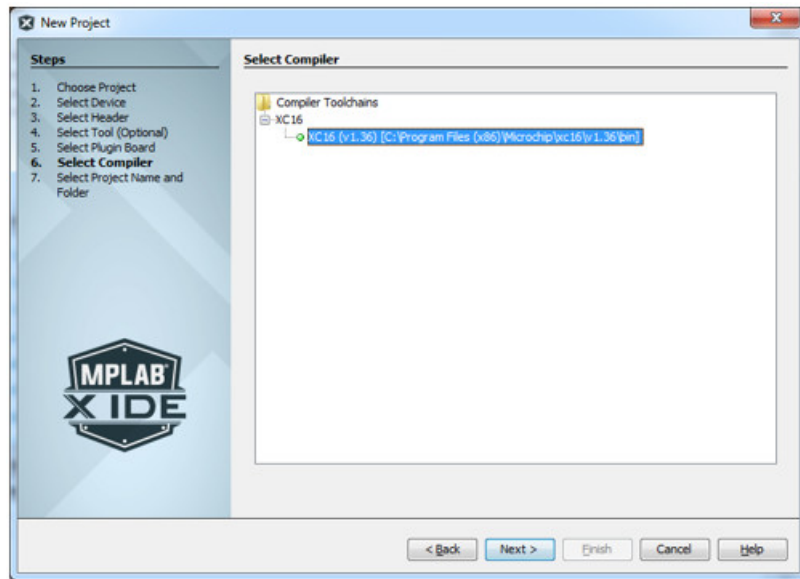


Figure 3-5. Naming a New Project

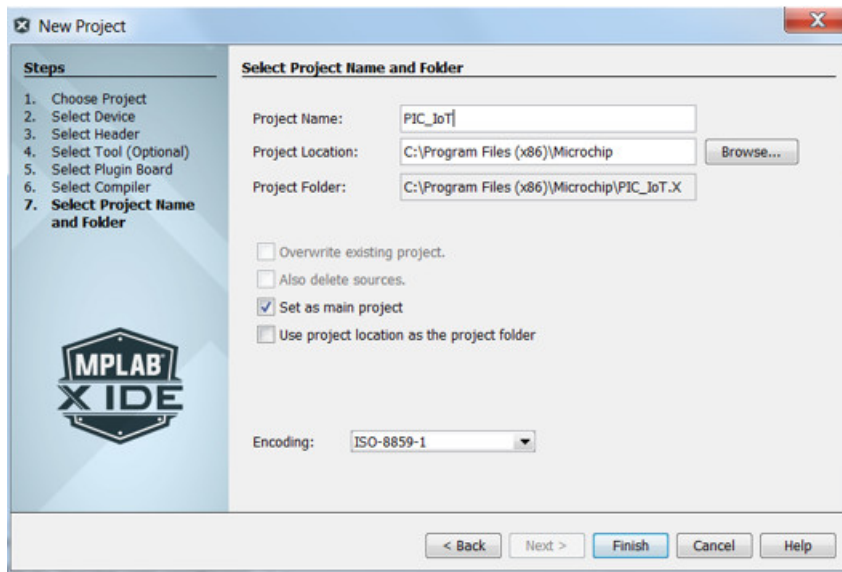
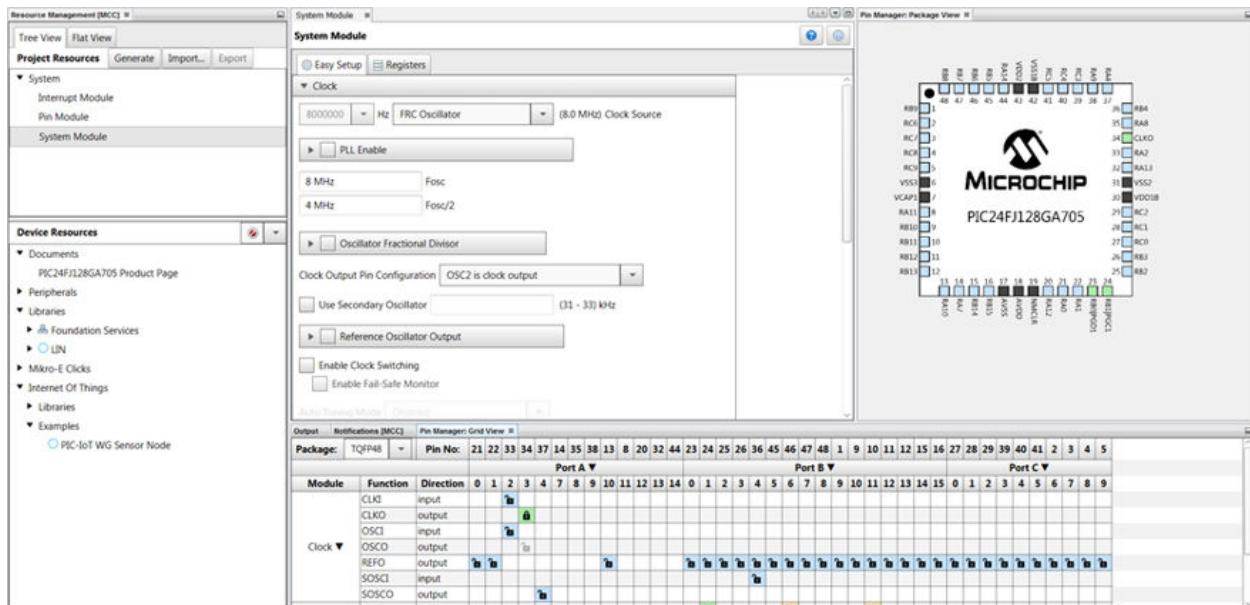


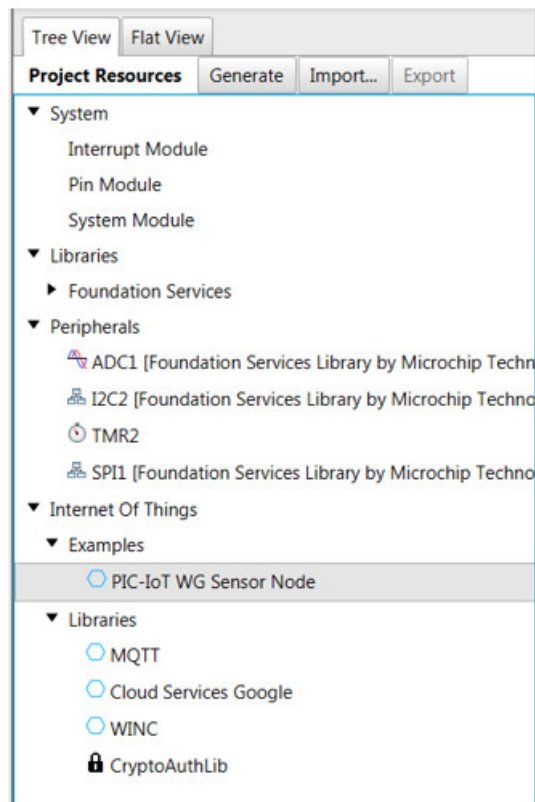
Figure 3-6. MCC Start Page



3.1.1.2 Configuring the Settings of the Project

The example module makes use of multiple libraries and peripherals. To configure the libraries, double-click on each library in the Device Resources window (see the [Figure 3-7](#)) to view their setup windows.

Figure 3-7. MCHP-IoT Peripheral Libraries



3.1.1.3 Component Library and Peripherals

- **CryptoAuthLib**

- The Crypto Authentication Library (CryptoAuthLib) is not available for user modification but it shows the macros that need to be enabled for the Crypto Authentication functionalities for the PIC-IoT board to work. It also indicates the communication settings between the ECC608 chip and the embedded microcontroller on board (see the [Figure 3-8](#)).

- **WINC**

- Under the WINC library, the user can configure the default SSID, password, and the authentication type used for the network to which the board will be connected (see the [Figure 3-9](#)).

- **Cloud Services - Google**

- The Cloud Services Google library contains settings for developers to use customer Google Cloud Project credentials by modifying the Project ID, Project Region, and Registry ID. These credentials configure which Google Cloud Server the Socket connection is established over. The default credentials used are those from the public Microchip sandbox project (see the [Figure 3-10](#)) used for the PIC-IoT demonstration application.

- **Message Queuing Telemetry Transport (MQTT)**

- MQTT is used as a messaging protocol that operates on top of a TCP/UDP connection to transporting data between client and broker over the Cloud. In MCC, the user can change their MQTT host and connection time-out duration (see the [Figure 3-11](#)).

Figure 3-8. CryptoAuthLib MCC

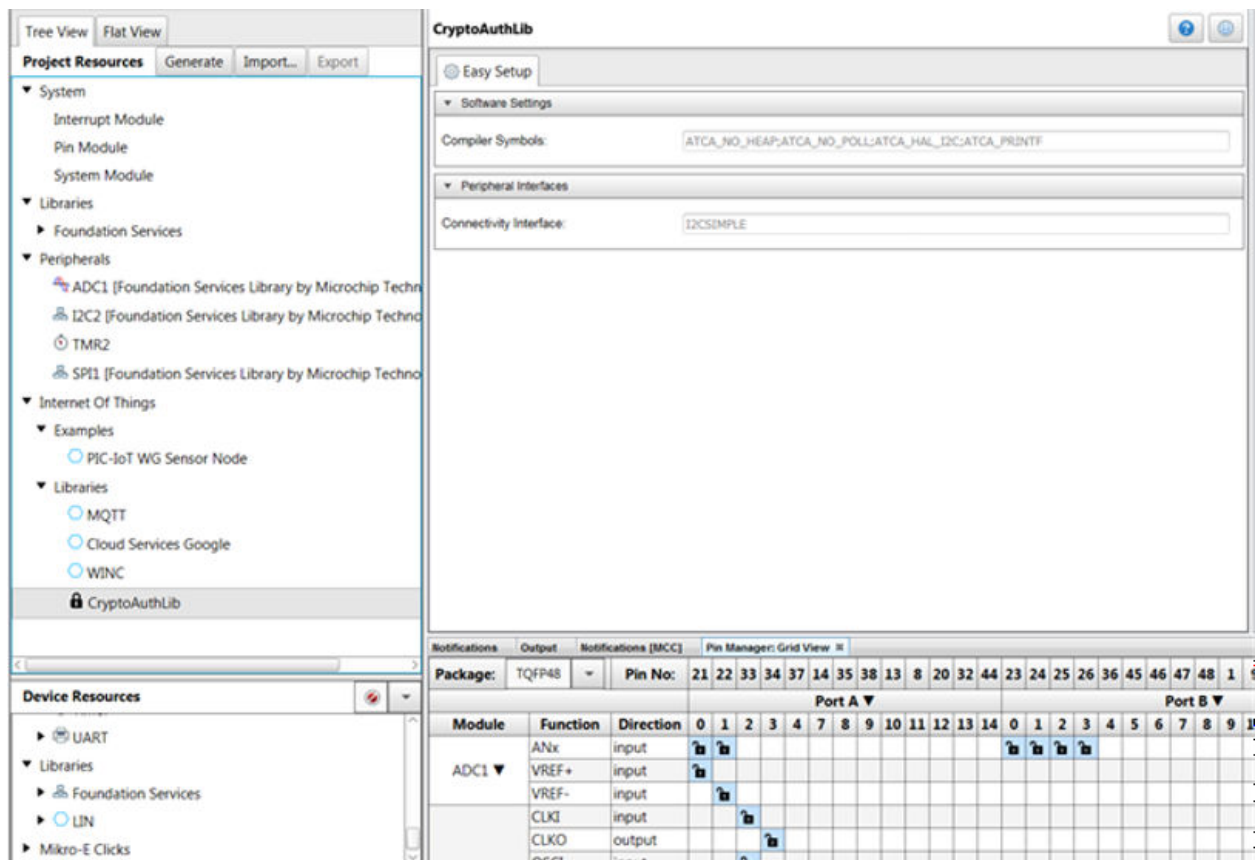


Figure 3-9. WINC MCC

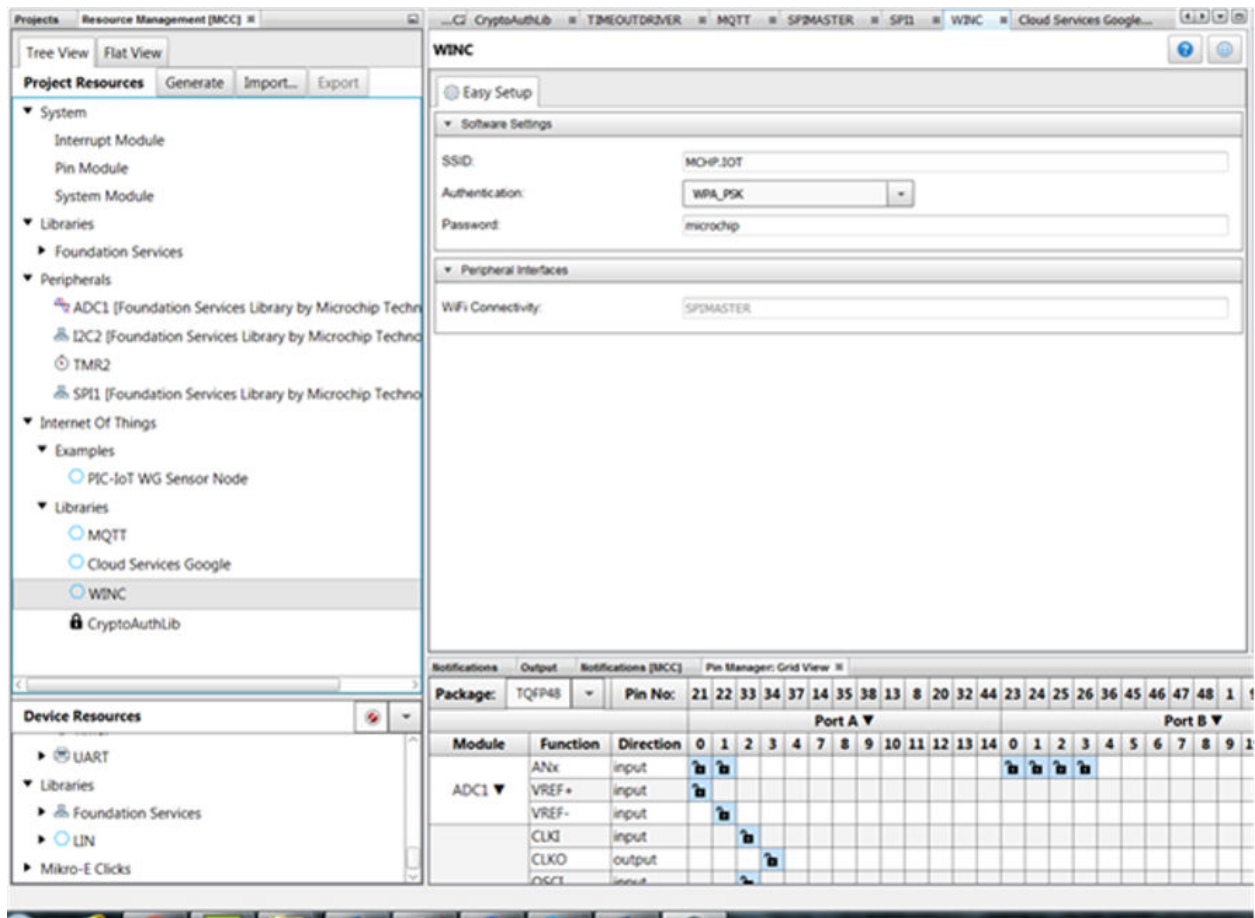


Figure 3-10. Cloud Services Google MCC

The screenshot displays the Microchip Configuration (MCC) tool interface. On the left, the 'Project Resources' tree shows the project structure, including 'System', 'Libraries', and 'Peripherals'. The 'Cloud Services Google' library is selected under 'Internet Of Things' > 'Libraries'. The main panel on the right shows the 'Easy Setup' configuration for 'Cloud Services Google'. The configuration includes fields for 'Project Id' (pic-iot), 'Project Region' (us-central1), and 'Registry Id' (PIC-IOT). Below these, the 'Library Dependencies' section shows 'Crypto Authentication' (CryptoAuthLib), 'MQTT Service' (MQTT), and 'WiFi Connectivity' (WINC). At the bottom, the 'Pin Manager' grid shows the pin configuration for the TQFP48 package. The grid lists modules and functions with their directions and pin assignments.

Package:	TQFP48	Pin No:	21	22	33	34	37	14	35	38	13	8	20	32	44	23	24	25	26	36	45	46	47	48	1	
Module		Function	Direction	0	1	2	3	4	7	8	9	10	11	12	13	14	0	1	2	3	4	5	6	7	8	9
ADC1		ANx	input																							
		VREF+	input																							
		VREF-	input																							
		CLKI	input																							
		CLKO	output																							
		OSCT	input																							

The screenshot displays the Arduino IDE environment. On the left, the 'Project Resources' panel lists various libraries, with 'MQTT' highlighted under the 'Internet Of Things' category. The main workspace shows the 'MQTT' configuration window. The 'Easy Setup' tab is active, and the 'Software Settings' section is expanded, showing the following configuration: MQTT Host: mqtt.googleapis.com, MQTT Port: 443, and Connection Timeout: 10. The 'Peripheral Interfaces' section shows the Scheduler Service set to TIMEOUTDRIVER. At the bottom, the 'Pin Manager' is visible, showing a grid view of pins for the TQFP48 package, with columns for Port A and Port B, and rows for various pin functions like ANx, VREF+, VREF-, CLKI, and CLKO.

- Once the changes are made, click the **Generate** button on the left-hand corner of the window (see the [Figure 3-12](#)) and wait for the generation to complete.
- For the code to work at optimal level, the user needs to change the optimization settings for the compilers. Right-click on the project name and select **Properties**. Click XC16 in the left-hand sidebar (see the [Figure 3-13](#)).
- Select **Most Useful Options** in the Categories drop-down menu (see the [Figure 3-14](#)). Select Optimization Level "s" from the drop-down menu. Click the **Apply** button and then **OK**.
- Click the **Make and Program Device** button near the middle of the toolbar. Make sure the board is connected to the system during programming.

Figure 3-12. Generating MCC Code

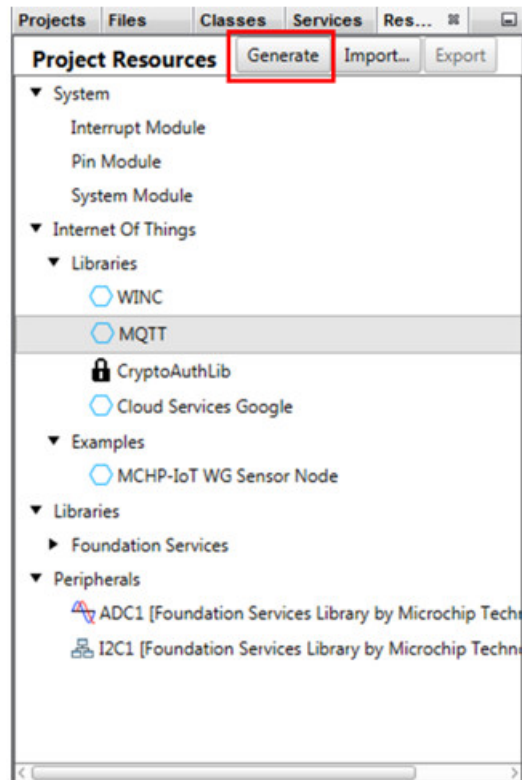


Figure 3-13. Project Properties

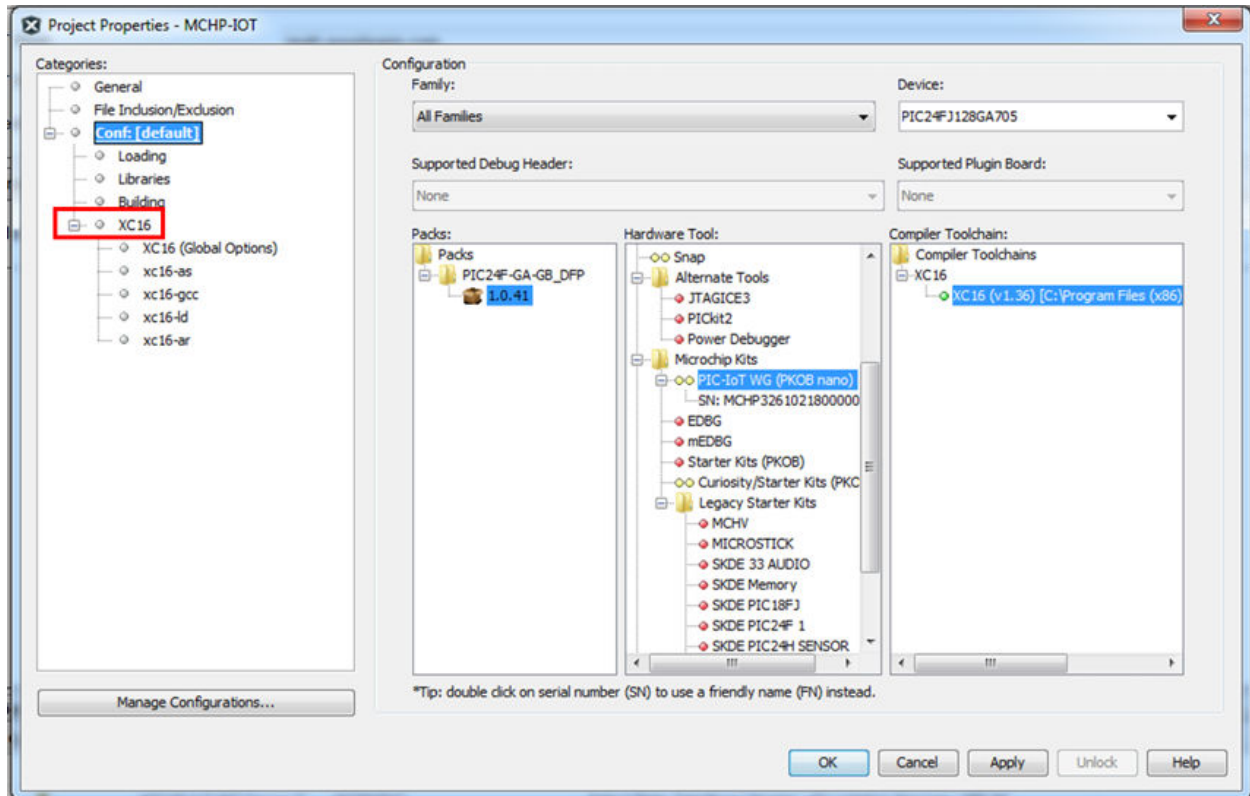
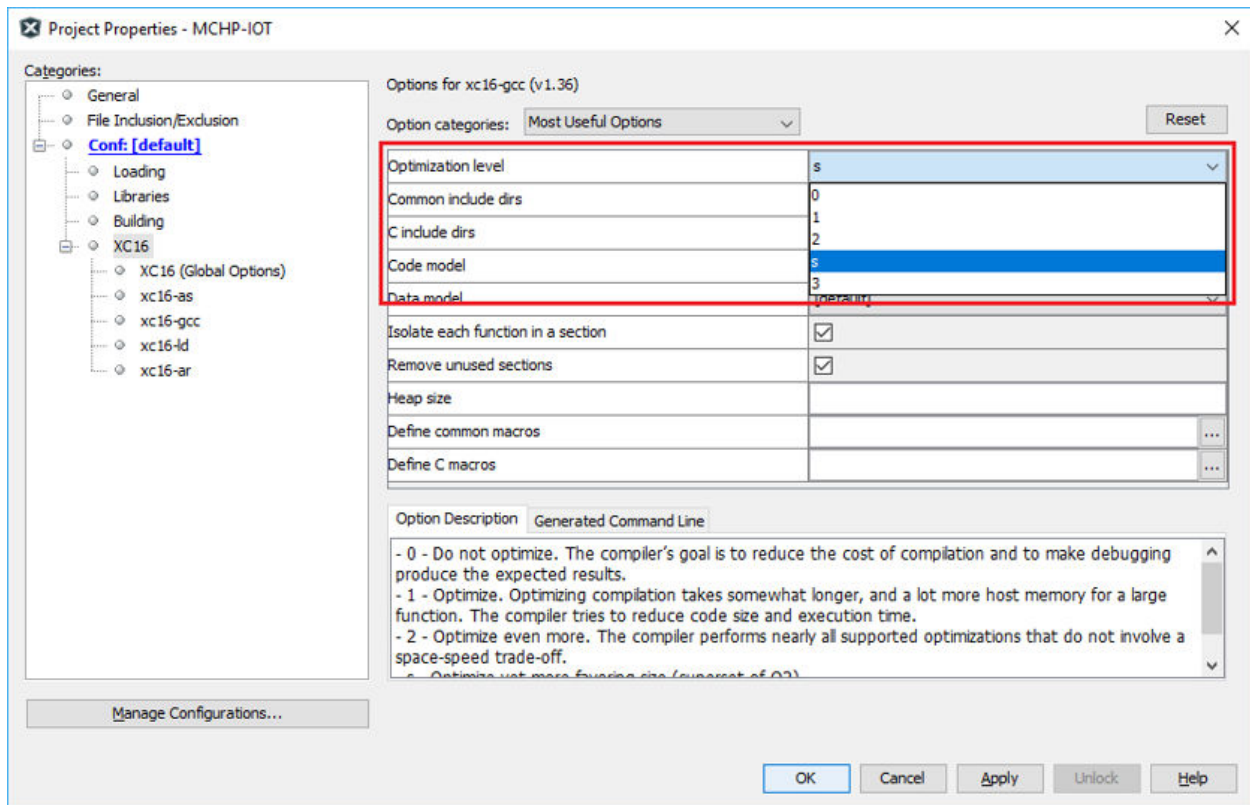


Figure 3-14. Optimization Settings



4. Chapter 4: Hardware Guide

4.1 Technical Features

- Embedded Microcontrollers:
 - PIC24FJ128GA705
- Four Application LEDs (Blue, Green, Yellow, Red)
- Two Mechanical Buttons
- WINC1510 Wi-Fi Module
- TEMT6000 Light Sensor
- MCP9808 Temperature Sensor
- ATECC608A CryptoAuthentication™ Device
- mikroBUS™ Click® Header Footprint
- PKOB nano
 - Board identification in Microchip MPLAB X
 - One green board power and status LED
 - Programming and debugging
 - Virtual COM port (CDC)
 - Two logic analyzer channels (DGI GPIO)
- USB and Battery Powered
- Li-Ion/LiPo Battery Charger
- Fixed 3.3V

4.2 Hardware Description

4.2.1 On-Board Debugger

The PIC-IoT WG boards are paired with an Embedded Debugger for on-board programming and debugging called PKOB nano. The PKOB nano 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 MPLAB X, the PKOB nano debugger interface can program and debug the central microcontroller.

4.2.2 Power

The PIC-IoT development boards can be powered through the USB port, or by a Li-Ion/LiPo battery.

The charger is configured to limit the charge current to 100 mA to prevent overcharging of small capacity batteries. The minimum recommended battery capacity is 400 mAh.

Note: The MCP73871 has a battery charge voltage of 4.2V; ensure any battery used is also of this charge voltage.

Table 4-1. Charger Status LEDs

LEDs	Function
Red (charging)	The battery is being charged by USB.

.....continued	
LEDs	Function
Red (discharging)	The battery voltage is low. Triggers if the voltage is under 3.1V.
Green	Charge complete.
Red and Green	Timer Fault. The 6-hour charge cycle has timed out before complete charge.

4.2.3 mikroBUS Socket

The PIC-IoT WG Sensor Node boards feature a mikroBUS Socket footprint for expanding functionality of the development kit using MikroElektronika Click Boards, and other mikroBUS add-on boards.

4.2.4 WINC1510

Microchip's WINC is a low-power consumption 802.11 b/g/n IoT module, specifically optimized for low-power IoT applications. The WINC is available in a QFN package, or as a certified module.

4.2.5 ATECC608A

The ATECC608A is a secure element from the Microchip CryptoAuthentication portfolio with advanced Elliptic Curve Cryptography (ECC) capabilities. The ATECC608A CryptoAuthentication device on the PIC-IoT development board is used for managing the private and public keys used for the secure IoT communication.

4.2.6 Temperature Sensor

The MCP9808 digital temperature sensor converts temperatures between -20°C and +100°C to a digital word value with $\pm 0.25^\circ\text{C}/\pm 0.5^\circ\text{C}$ (typical/maximum) accuracy. The 7-bit I2C address is (0x18) used by the embedded device acting as a master to communicate with the MCP9808 slave device.

4.2.7 Light Sensor

The TEMT6000X01 light sensor is mounted on the PIC-IoT development board for measuring the light intensity as a 10-bit value (0 – 1023). The sensor is a current source that will induce a voltage across the series resistor, which in turn can be measured by the embedded device's ADC.

4.2.8 LEDs

There are four LEDs available on the PIC-IoT development board that are controllable with PWM, or GPIO.

4.2.9 Mechanical Buttons

The boards have two mechanical buttons connect to Interrupt-on-Change (IOC)-capable GPIO pins.

4.3 Hardware Revision

4.3.1 Identifying Product ID and Revision

The revision and product identifier of PIC-IoT development board can be found in two ways: either through MPLAB X, or by looking at the sticker on the bottom side of the PCB. By connecting a PIC-IoT development board to a computer with 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. Boards with limited space have a sticker with only a QR-code, which contains a serial number string.

The serial number string has the following format: 'nnnnrrsssssssss', where 'n' is the product identifier, 'r' is the revision letter and 's' is the serial number.

The product identifier of the board is A09-3203.

5. FAQs, Tips and Troubleshooting

5.1 FAQs and Tips

1. How can I change the Wi-Fi configuration?

There are four ways to do it:

1. Connect to the USB and click the "[click-me](#)" file to reach the <https://www.pic-iot.com/> page (with the /device/{deviceId} suffix). Then enter the new credentials in the web form. Download the resulting file to the CURIOSITY drive. Read more in the [2.3.1 Via PIC-IoT Webpage](#) section.
2. Connect to the USB and open a serial port terminal (Windows users will need to install serial port drivers). From the command line, use the `Wi-fi` command. Read more in the [2.3.1 Via PIC-IoT Webpage](#) section.
3. Press the SW0 button while powering-up the board. The WINC will turn to Access Point mode. Connect the laptop or phone to it and fill in the online form. See the [2.3 Connecting the Board to Wi-Fi Networks](#) section for details.
4. Use MCC to re-build the project after changing the default Wi-Fi configuration in the WINC module. Re-program the board using MPLAB X or drag and drop the new image to the CURIOSITY drive. Further details can be found in the [3.1 Generating code from MCC](#) section.

2. How can I change the Wi-Fi credentials using the online form without exposing the details to security threats?

Although it appears in your browser, the Wi-Fi credential setup form does not transfer any information to third parties. A small text file (WIFI.CFG) is created (this can also be done manually using any text editor) and it is recommended that you save it directly to the CURIOSITY drive. (Since the browser settings vary according to platform and personal preferences, you might have to change them or perform a drag and drop from your default download folder). Even though it looks like the WIFI.CFG file is now stored on the CURIOSITY drive, this is just an artifact of your operating system (caching). No file is permanently recorded and the information contained is immediately used to update the Wi-Fi module settings. These settings will be maintained after subsequent power cycles of the PIC-IoT board, but the file will disappear.

3. Can I use my phone/tablet alone to perform the demo?

Assuming you have a way to provide power to the board (a USB back-up battery, a USB charger, a Li-Ion battery, or other 3.3V-5V power supply), you can scan the QR code (on a sticker under the board, next to the Microchip and Google color logos) using any smartphone camera (old operating system versions might still require a separate app) and open the resulting link in the smartphone browser.

4. I scanned the bar code with my phone/tablet but nothing happened?

Make sure you are scanning the QR code present on the sticker under the PIC-IoT WG board. You can recognize it by the distinguishing squares on the three of its corners and its proximity (same sticker) to the MCHP and Google logos (in color). Although there are also other bar codes present on the Wi-Fi module and/or the anti-static bag the board came with, those are not QR codes

5. Which battery is recommended to be used with the IoT WG Sensor Node board?

We recommend Li-Ion or Li-Poly batteries with 500 mAh capacity and 3.7 V nominal. Coin cell batteries are not applicable as they cannot provide enough current for the Wi-Fi module when in transmission (short bursts due to excess of 200 mA).



Pro Tips:

The following steps are not required for operating the IoT WG Sensor Node board but will increase significantly the possibility of positive results.

1. **Get a USB cable with all the four wires connected.** There are a lot of non-compliant USB cables out there that are providing only 5V power (two wires). How can you verify it? Plug the board into your laptop and check in the File Manager (Finder) for the presence of a new hard drive (named CURIOSITY). If you do not see it, popping up after a second or two, the cable is not the appropriate one.
2. **Prepare your Wi-Fi router for the demo.** The easiest way to go is to setup your phone as a hotspot. You will want to assign to it the following credentials, name (SSID): **MCHP.IOT** and password: **microchip** (WPA 2 is assumed, do not use WEP nor OPEN.) This Wi-Fi configuration is the factory default for all boards, so it will minimize the effort for first users. If preparing for a (medium/large) classroom demo, you will want to setup a proper Wi-Fi router (2.4 GHz) instead. This will give you range and capacity while using the same Wi-Fi credentials, if possible.
3. Make Google Chrome or Firefox your default web browser (if only for the day of the demo/visit). Safari works well on MACs. Even if the latest versions of Internet Explorer are known to work OK on Windows 10, the chances decrease significantly as the Windows user base is less uniform and you might find some (customer) corporate policies to maintain antique versions for legacy application compatibility reasons.
4. Ensure no pop-up blockers or other anti-virus extensions to the browsers are active. These can and will interfere with the script that is at the heart of the <https://www.pic-iot.com/> microsite. Often, these can be selectively disabled for that specific webpage.
5. Count the number of cellphones in the room. If it is more than 50, do not try a live demo. The board will eventually establish a connection but it might not happen as promptly and cleanly as you would like to demonstrate. The Wi-Fi pollution is a serious problem at large events, fairs and in general in any large public space nowadays.
6. Check your setup ahead of time and verify you have 4G (or LTE if using the phone hotspot), a full charge and no firewall (if using a router) blocking access to the Google Cloud server (try opening any secure webpage such as: <https://microchip.com>. If you get the homepage, the way is clear).

5.2 LED Status Troubleshooting

Table 5-1. Troubleshooting and Diagnostics

LED Sequence	Description	Diagnosis	Action
	No LEDs are On	Board is not programmed	Download image .hex file from the bottom of the microsite page.
	Only Red LED is ON	Board is not connected to the Wi-Fi Access Point	<ul style="list-style-type: none">• Verify Wi-Fi credentials• Confirm Access Point is available

PIC-IoT WG User Guide

FAQs, Tips and Troubleshooting

.....continued










LED Sequence	Description	Diagnosis	Action
	Blue LED BLINKS with all other LEDs OFF	Board is in Soft AP mode	<ul style="list-style-type: none"> • Must connect to board using phone, or a network capable device • Send updated credentials via Soft AP
	Green LED is BLINKING; Other LEDs are OFF	Board is using DEFAULT CREDENTIALS	<ul style="list-style-type: none"> • Allow board to connect to Access Point • Update CREDENTIALS through CLI if DEFAULTS selection was invalid
	Blue and Red LEDs are ON	Board is not connected to the Google IoT Cloud Servers	<ul style="list-style-type: none"> • Verify MQTT required ports • Verify project credentials • Check local network firewall settings • Use tethered cell phone or laptop connection for internet
	Blue, Green and Red LEDs are ON	Sensor data is not being published to Cloud	<ul style="list-style-type: none"> • Verify device registration to the project • Check Google account settings
	Blue and Green LEDs are ON. Yellow LED is BLINKING	Everything is working	<ul style="list-style-type: none"> • Nothing to be done
 OR 	Blue and Green LEDs are ON. Yellow LED held HIGH/LOW	Subscribe topic toggle value received	<ul style="list-style-type: none"> • Nothing to do. • LED will reflect 'Toggle' value LED behavior returns to normal after HOLD PERIOD

Table 5-2. PKOB nano LED Troubleshooting

LED Sequence	Description	Diagnosis	Action
	PKOB nano LED is OFF	Board is not powered	<ul style="list-style-type: none">• Check the USB connection• Replace the board
	PKOB nano LED is ON but CURIOSITY driver not found	Faulty USB connection	<ul style="list-style-type: none">• Check the PC device manager• Replace the USB cable• Replace the USB connector

6. Appendix

6.1 Relevant Links

The following list contains links to the most relevant documents and software for the PIC-IoT WG board. For those accessing the electronic version of this document, the underlined labels are clickable and will redirect to the appropriate website..

- [MPLAB® X IDE](#) - Free IDE to develop applications for Microchip microcontrollers and digital signal controllers.
- [MPLAB® Code Configurator \(MCC\)](#) - a free, graphical programming environment that generates seamless, easy-to-understand C code to be inserted into the project. Using an intuitive interface, it enables and configures a rich set of peripherals and functions specific to the application.
- [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 Embedded Debugger Data Gateway Interface found on Xplained Pro boards and COM ports.

6.2 Board Layouts

The PIC-IoT board layouts can be seen below.

Figure 6-1. PIC-IoT WG Development Board Schematic

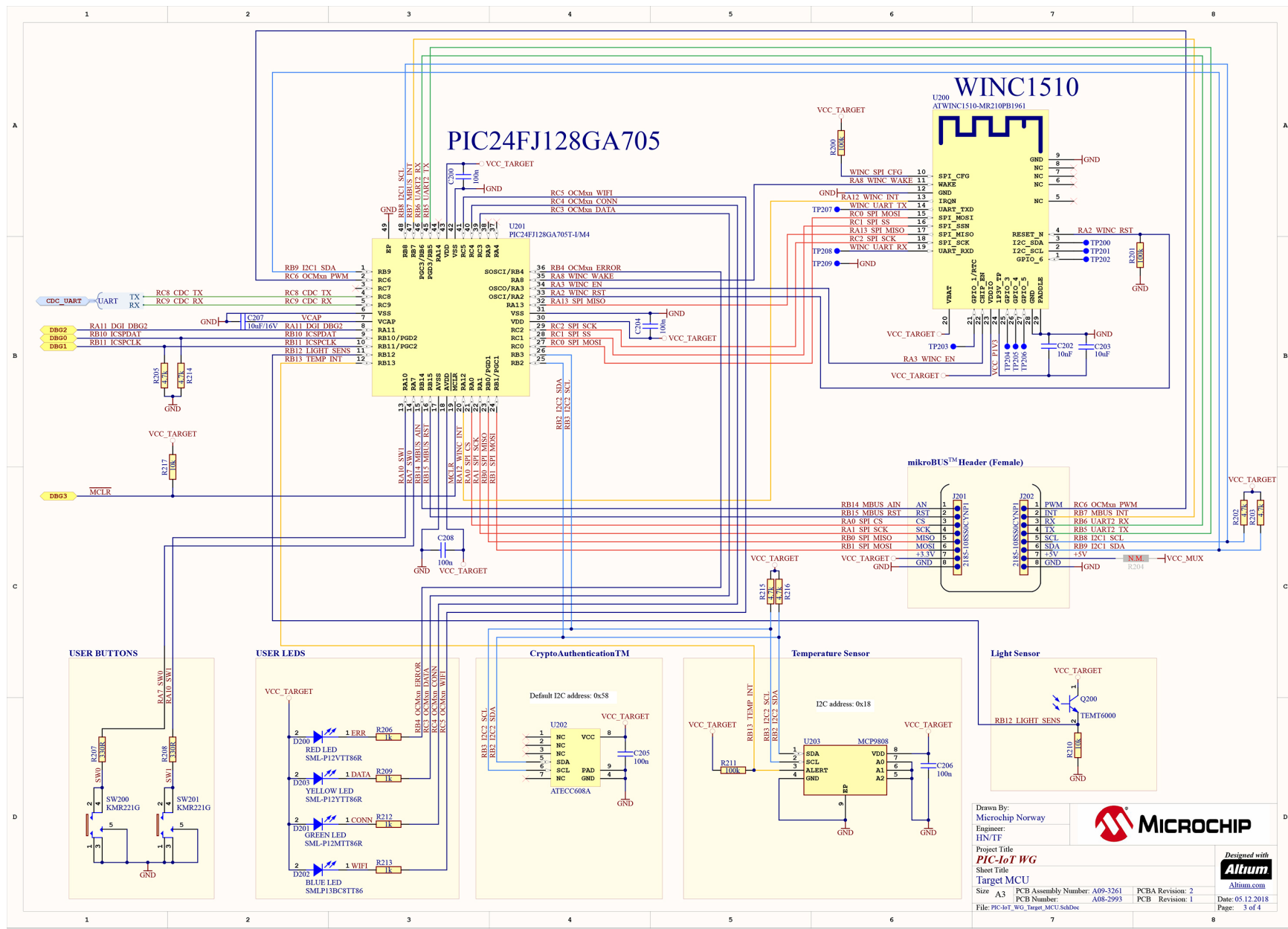
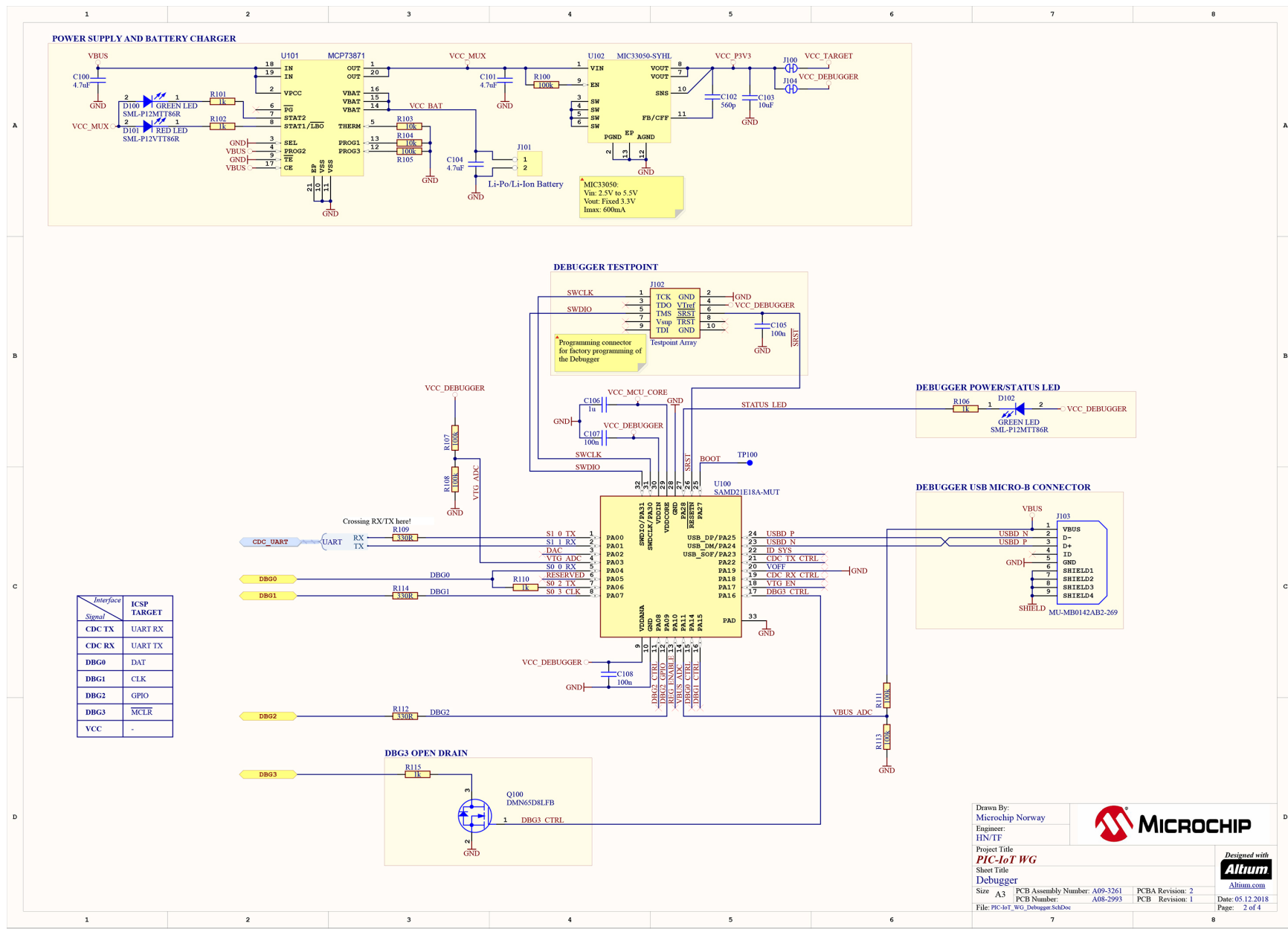


Figure 6-2. PIC-IoT WG Development Board Schematic



7. Document Revision History

Revision	Date	Comment
A	2/2019	Initial release of the document.

The Microchip Web Site

Microchip provides online support via our web site at <http://www.microchip.com/>. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **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 (FAQ), technical support requests, online discussion groups, Microchip consultant 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

Customer Change Notification Service

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

To register, access the Microchip web site at <http://www.microchip.com/>. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of 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.”

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

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, Klear, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire 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, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

Silicon Storage Technology is a registered trademark 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.

© 2019, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-4186-1

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

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: 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: 45-4450-2828 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 Germany - Garching Tel: 49-8931-9700 Germany - Haan Tel: 49-2129-3766400 Germany - Heilbronn Tel: 49-7131-67-3636 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:](#)

[AC164164](#)