Chapter 1: Introduction

After completing this chapter, you will understand what this class is, what topics are covered, the overall class objectives, and you will understand what is contained in the ModusToolbox™ any cloud solution.

Table of contents

Document conventions

Convention	Usage	Example
`Courier New`	Displays code and text commands	`CY_ISR_PROTO(MyISR);` `make build`
Italics	Displays file names and paths	*sourcefile.hex*
[**bracketed, bold**]	Displays keyboard commands in procedures	[**Enter**] or [**Ctrl**] [**C**]
Menu > Selection	Represents menu paths	**File > New Project > Clone**
Bold	Displays GUI commands, menu paths and selections, and icon names in procedures	Click the **Debugger** icon, and then click **Next**.

1.1 What is this class?

This is a class to teach how to use Wi-Fi in ModusToolbox™ applications. The descriptions and exercises use a PSoC™ 6 MCU as a host to a CYW43012 device.

After completing this class, you should be able to create and debug Wi-Fi applications using ModusToolbox™ tools. This class will introduce you to several concepts including: the network stack, sockets, TLS, HTTP, MQTT, and interacting with Amazon Web Services (AWS).

1.2 Prerequisites

This class assumes that you know the basics of using the ModusToolbox™ ecosystem, how to interact with PSoC™ 6 MCUs including using peripherals, and how to program a PSoC™ 6 device. If you are unfamiliar with these topics, the following classes should be reviewed first:

- ModusToolbox™ Software Training Level 1 – Getting Started
- ModusToolbox™ Software Training Level 2 – PSoC™ MCUs

1.3 Required software

- ModusToolbox™ tools

You should already have ModusToolbox™ tools installed on your system. If not, refer to the ModusToolbox™ Software Training Level 1 Getting Started class or visit the ModusToolbox™ Software website for instructions.

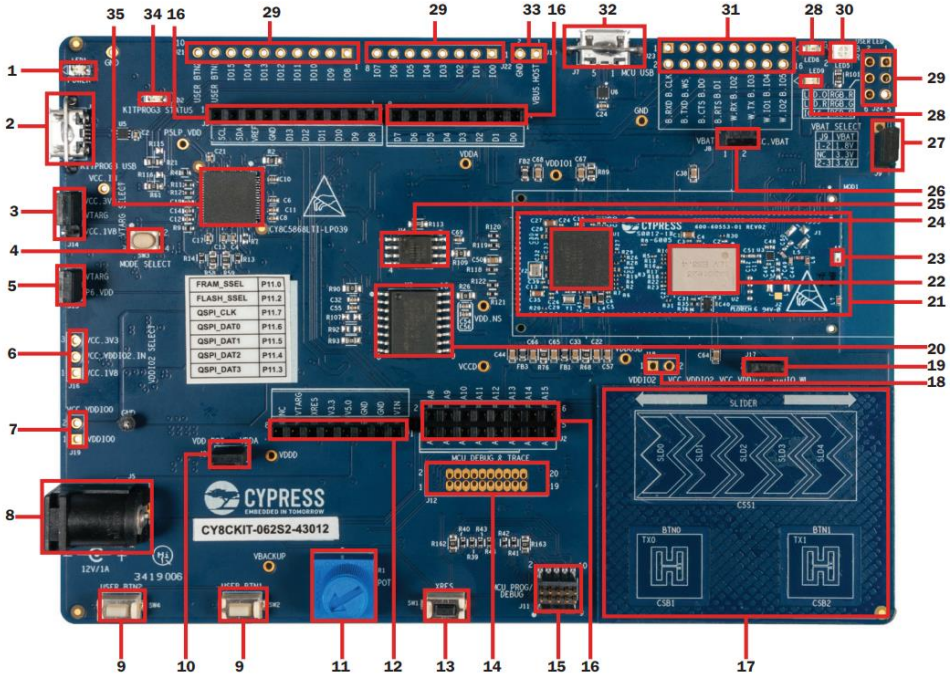
1.4 Development kits

For this class we will use the following development kits:

1.4.1 PSoC™ 6 kit

CY8CKIT-062S2-43012 – A PSoC™ 6-2M MCU and a CYW43012 Wi-Fi + Bluetooth® Combo

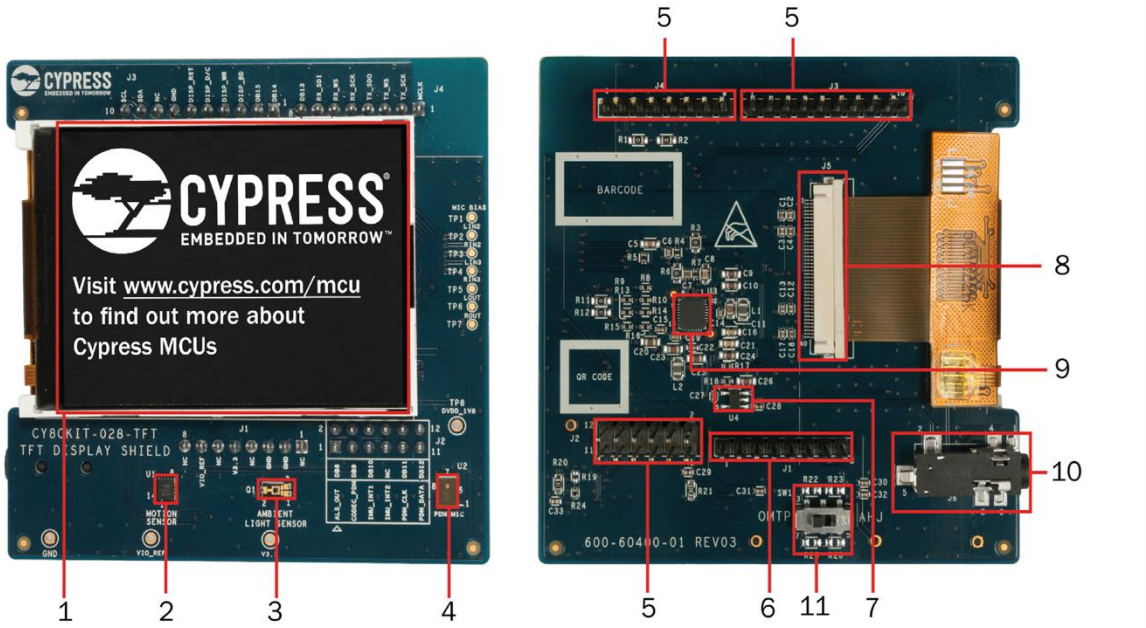
Note: *If you are taking this course on your own, you will need two of these kits to complete some of the exercises in later chapters. If you are taking this course in a classroom setting, you will only need one of these kits.*



1. Power LED (LED1)
2. KitProg3 USB connector (J6)
3. PSoC™ 6 MCU VDD power selection jumper (J14)
4. KitProg3 programming mode selection button (SW3)
5. PSoC™ 6 MCU VDD current measurement jumper (J15)
6. PSoC™ 6 MCU VDDIO2 and CYW43012 VDDIO power selection jumper (J16)
7. PSoC™ 6 MCU VDDIO0 current measurement jumper (J19)
8. External power supply VIN connector (J5)
9. PSoC™ 6 MCU user buttons (SW2 and SW4)
10. Potentiometer connection jumper (J25)
11. Potentiometer (R1)
12. Arduino-compatible power header (J1)
13. PSoC™ 6 MCU reset button (SW1)
14. PSoC™ 6 MCU debug and trace header (J12)
15. PSoC™ 6 MCU program and debug header (J11)
16. Arduino Uno R3-compatible I/O headers (J2, J3, and J4)
17. CAPSENSE™ slider (SLIDER) and buttons (N0 and BTN1)
18. PSoC™ 6 MCU VDDIO2 current measurement jumper (J18)
19. CYW43012 VDDIO current measurement jumper(J17)
20. Infineon serial NOR flash memory (S25FL512S, U3)
21. Infineon PSoC™ 6 (2M) MCU with CYW43012 Carrier Module (CY8CMOD-062S2-43012, MOD1)
22. CYW43012 based Murata Type 1LV module
23. Wi-Fi/Bluetooth® antenna
24. PSoC™ 6 MCU
25. Infineon serial Ferroelectric RAM (CY15B104QSN, U4)
26. CYW43012 VBAT current measurement jumper (J8)
27. CYW43012 VBAT power selection jumper (J9):
28. PSoC™ 6 MCU user LEDs (LED8 and LED9)
29. PSoC™ 6 I/O header (J21, J22, J24)
30. RGB LED (LED5)
31. Wi-Fi/Bluetooth® GPIO header (J23)
32. PSoC™ 6 USB device connector (J7)
33. Optional USB Host power supply header (J10)
34. KitProg3 status LED (LED2)
35. KitProg3 (PSoC™ 5LP MCU) programmer and debugger (CY8C5868LTI-LP039, U2)
36. MicroSD Card holder (J20) (on back of board)

1.4.2 Arduino compatible shield

CY8CKIT-028-TFT



1. 2.4-inch TFT display	7. TFT display power control load switch (U4)
2. Motion Sensor (U1)	8. TFT display connector (J5)
3. Ambient Light Sensor (Q1)	9. Audio CODEC (U3)
4. PDM microphone (U2)	10. Audio Jack (J6)
5. Arduino compatible I/O headers (J2, J3, J4)	11. Audio Jack Selection (OMTP/AHJ) Switch (SW1)
6. Arduino compatible power header (J1)	

1.5 Introduction to ModusToolbox™ any cloud

In addition to the PSoC™ 6, the development kit used in this course has a CYW43012 connectivity device that supports both Wi-Fi and Bluetooth®. Let's face it, most electronic devices these days have some sort of wireless connectivity. In this chapter we will focus on the Wi-Fi piece of the puzzle. The connection between the PSoC™ 6 and the connectivity device is done using SDIO for Wi-Fi (or SPI on some devices) and an HCI UART interface for Bluetooth®. Furthermore, wireless co-existence between Wi-Fi and Bluetooth® is supported so both functions can operate simultaneously.

Infineon has several cloud solutions depending on how you decide to manage your connected devices. The ModusToolbox™ any cloud solution was built for customers with their own cloud device management back end, whether hosted on AWS, Google, Azure, AliCloud, or any other cloud infrastructure.

If you are using Amazon Web Services IoT Core for your device management, then you may prefer to use our more customized AWS IoT Core solution. Likewise, if you are using Arm Pelion for your device management, you may prefer our customized Arm Pelion solution. Note that if you are using Amazon's cloud services for storage but not their IoT Core device management, ModusToolbox™ any cloud is a great solution for you, as you will see in some of the upcoming exercises.

ModusToolbox™ any cloud provides features such as the Wi-Fi Connection Manager, a Secure Socket layer, support for application layer cloud protocols, Bluetooth® Low Energy (Bluetooth® LE) functionality, and Low Power Assist (LPA).

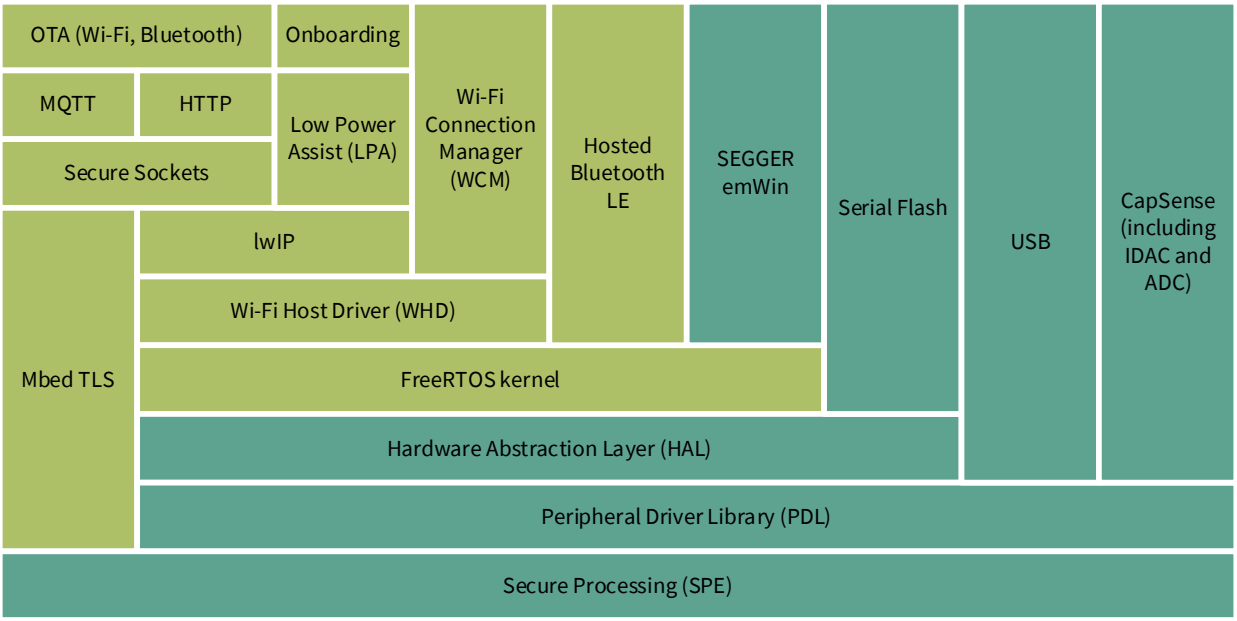
Currently, ModusToolbox™ any cloud supports the MQTT and HTTP application layer cloud protocols.

While the ModusToolbox™ any cloud solution provides core functionality, including connectivity, security, firmware upgrade support, and application layer protocols like MQTT, it is also flexible so you can modify or extend it to match your needs.

1.6 ModusToolbox™ any cloud library descriptions

The ModusToolbox™ any cloud solution is distributed as a collection of libraries that work together to help you easily get your IoT device up and running on the Cloud. Some of the libraries were written by Infineon, while others are industry standard open source libraries. As you have seen, these can be pulled into a ModusToolbox™ application easily by using the Library Manager.

The ModusToolbox™ any cloud solution libraries fit together with the core PSoC™ 6 libraries like this:



Most libraries are available as GitHub repositories. These “repos” contain source files, readme files, and documentation such as an API reference.

The following subsections describe the ModusToolbox™ any cloud libraries.

1.6.1 Wi-Fi Middleware Core

This library is a collection of the core libraries that any Wi-Fi application will need plus a little bit of glue logic. By adding the Wi-Fi Middleware Core library to an application, you get:

- Wi-Fi Host Driver (WHD) - Embedded Wi-Fi Host Driver that provides a set of APIs to interact with Cypress WLAN chips.
- FreeRTOS - FreeRTOS kernel, distributed as standard C source files with configuration header file, for use with the PSoC™ 6 MCU.
- CLib Support - This is a support library that provides the necessary hooks to make C library functions such as malloc and free thread safe. This implementation is specific to FreeRTOS.
- lwIP - A Lightweight open-source TCP/IP stack. lwIP stands for "lightweight IP".
- MbedTLS - An open source, portable, easy to use, readable and flexible SSL library, that has cryptographic capabilities.
- RTOS Abstraction Layer - Minimalistic RTOS-agnostic kernel interface allowing middleware to be used in multiple ecosystems, such as Arm’s Mbed OS and Amazon’s FreeRTOS. It is not recommended for use by the end application - the user should write code for their RTOS of choice such as FreeRTOS.

- Secure Sockets - Network abstraction APIs for underlying lwIP network stack and MbedTLS security library. The secure sockets library eases application development by exposing a socket like interface for both secure and non-secure socket communication.
- Predefined configuration files for FreeRTOS, lwIP and MbedTLS for typical embedded IoT use-cases.
- Associated glue layer between lwIP and WHD.

The Library Manager name for the Wi-Fi Middleware Core library is *wifi-mw-core*. It includes the other libraries listed above automatically.

1.6.2 Wi-Fi Connection Manager (WCM)

The WCM makes Wi-Fi connections easier and more reliable. Firstly, it implements Wi-Fi Protected Setup (WPS) to simplify the secure connection of a device to a Wi-Fi access point (AP). This enables applications to store the credentials in non-volatile memory so that future connections are just automatic whenever the AP is available. Secondly, it provides a monitoring service to detect problems and keep connections alive, improving reliability.

The Wi-Fi Connection Manager library includes the Wi-Fi Middleware Core library as a dependency.

The Library Manager name for this library is *wifi-connection-manager*.

1.6.3 HTTP

This is actually two separate libraries – one for HTTP servers and one for HTTP clients. The libraries enable both secure (https) and non-secure (http) modes of connection. They support RESTful HTTP methods: HEAD, GET, PUT, and POST.

The Library Manager names for these libraries are *http-client* and *http-server*.

1.6.4 MQTT

This library includes the open source AWS IoT device SDK embedded C library plus some glue to get it to work seamlessly in ModusToolbox™ any cloud. It is based on MQTT client v3.1.1 and supports QoS levels 0 and 1. Both secure and non-secure TCP connections can be used.

The Library Manager name for this library is *mqtt*.

1.6.5 Over-The-Air (OTA) Bootloading

The OTA toolkit library is an extensible solution based on MCUBoot that can be modified to work with any third-party or custom IoT device management software. With it you can rapidly create efficient and reliable OTA schemes. It currently supports OTA over MQTT and HTTP.

The Library Manager name for this library is *OTA*.

1.6.6 Bluetooth®/Bluetooth® LE Hosted Stack (BTSTACK and FreeRTOS Wrapper)

In addition to the great Wi-Fi support in ModusToolbox™ any cloud, you can use the Bluetooth® LE functionality in the 43xxx combo device to enable Bluetooth® LE for your device. For example, it can easily be used to enable Wi-Fi onboarding so that you can safely and quickly connect your device to a Wi-Fi network

using Bluetooth® LE to select the network and enter the password. One of the ModusToolbox™ any cloud examples will show you that exact thing.

In the Library Manager, you can include the *bluetooth-freertos* library which will include the BTSTACK library automatically.

1.6.7 Low Power Assistant (LPA)

The LPA is a library and associated settings in the Device Configurator that allow you to configure a PSoC™ 6 Host and WLAN (Wi-Fi / Bluetooth® Radio) device for optimized low-power operation. With LPA you can achieve the most aggressive power budgets by placing the host device into sleep or deep sleep modes while networks are quiet or when there is traffic that can be handled by the connectivity device.

The Library Manager name for this library is *LPA*.

1.7 Path Lengths

Windows has a path length limitation of 260 characters. This isn't usually a problem, but some of the libraries used for IoT functionality have very long paths. Therefore, you may run into a build error if the path to your application is too long. For example, the HTTP client and MQTT libraries include the aws-iot-device-sdk-embedded-C library as a dependency. When that library is included in an application, the build directory will have a file in this location:

<path_to_application>\build\CY8CKIT-062S2-43012\Debug\ext\mtb_shared\aws-iot-device-sdk-embedded-C\202103.00\libraries\aws\ota-for-aws-iot-embedded-sdk\source\dependency\3rdparty\tinycbor\src\cborencoder_close_container_checked.d

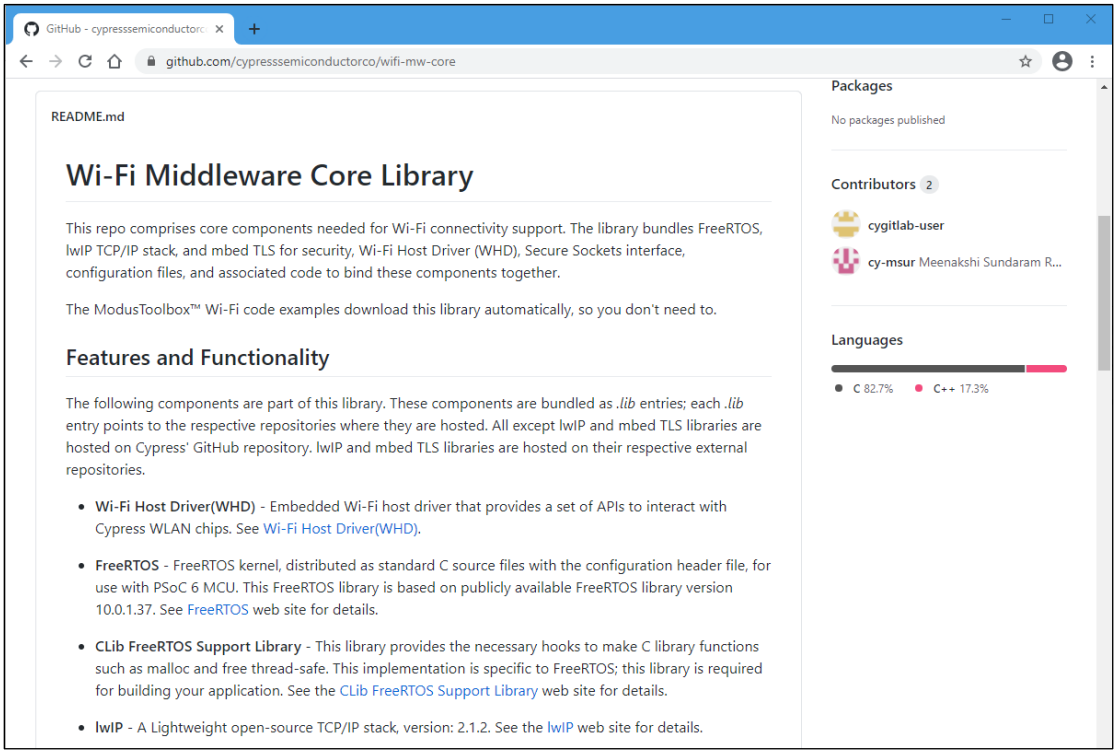
That's 209 characters, so only 51 characters are left for the path to the application (which by default is *C:\Users\<user_name>\mtw\<application_name>*.

If you see build errors indicating that file cannot be found in the build directory, it is likely due to the path length. You can shorten the name of the application or move your workspace to a location with fewer characters in the path (e.g. *C:\mtw*).

1.8 Documentation

The best place to find documentation for ModusToolbox™ any cloud is inside the individual libraries themselves. You can look on GitHub directly, you can open the documentation from a library that you have downloaded, or you can open the documentation for a downloaded library from inside the Eclipse IDE for ModusToolbox.

For example, here is the *README.md* for the Wi-Fi Middleware Core as seen on GitHub:



Once you have downloaded a library, you can go to the *docs* directory or you can open it from the Quick Panel inside Eclipse. In either case, you will typically find an *api_reference_manual.html* file with full descriptions of the API:

1.9 Code examples

It is always easier to start with an existing application rather than starting with a blank slate. Infineon provides a wealth of ModusToolbox™ any cloud code examples for just that reason.

All Infineon code examples are hosted on GitHub so ultimately that's where you will find them, but there are a few ways to get there:

1. Inside the project creator tool:

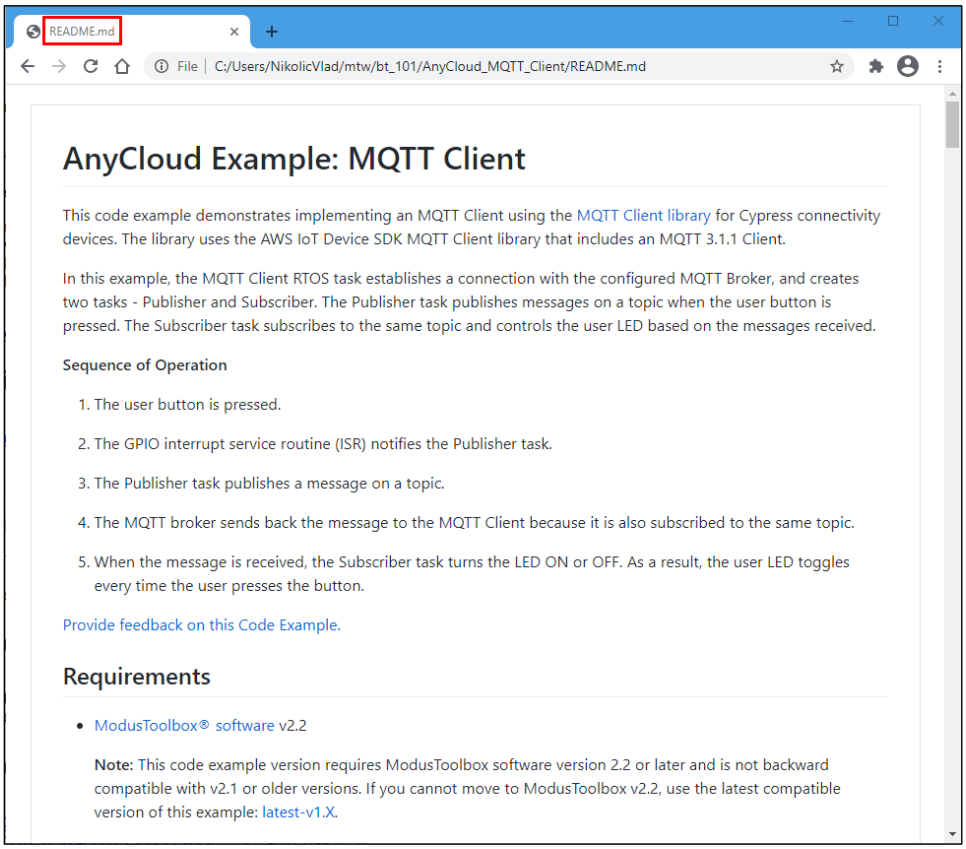
   Look for starter applications whose names start with "AnyCloud"

2. Directly from the Infineon GitHub website
3. From the Infineon website

The first method is recommended if you want to create an application from the code example since it handles all of the BSP and library downloads automatically.

Of course, you can learn a ton by looking at the source code and reading the comments from the examples. On top of that, every Infineon code example has a README.md file that explains the example, tells you which kits it will run on, shows how to set it up, explains how to use it, and explains the design. The README.md is always a good starting point.

You will need a markdown reader to get the most out of the README.md documents. There are plugins for most web browsers as well as an extension for VS Code. Markdown will also show inside the Eclipse IDE, but some things may be formatted strangely since Eclipse doesn't support everything that Markdown can do.



Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Published by

Infineon Technologies AG

81726 Munich, Germany

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.