Chapter 1: Introduction

After completing this chapter, you will understand what this class is, what topics are covered, and the overall class objectives. You will also understand the programmer provided on the development kit and how to update its firmware.

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 class is an in-depth look into the AIROC™ Bluetooth® SDK (BTSDK) MCU families. We introduce you to the basic operation of the MCU devices and familiarize you with their development flow within the ModusToolbox™ ecosystem. This enables you to create your own applications for devices in the AIROC™ Bluetooth® SDK (BTSDK) MCU families.

This is a "Level 2" class, meaning that it is intended as a detailed look into a specific product. This class is part of a series of classes that also include these other levels:

- Level 1 classes are an entry point into a particular topic and cover a broad range of topics at a shallow depth.
- Level 3 classes dig even deeper by looking at complete solutions such as Bluetooth®, Wi-Fi, Motor Control, or Machine Learning.

This class will cover application structure, using MCU peripherals, using an RTOS, and general low power operation. Bluetooth® operation is NOT covered in this class but once you understand the basics of using AIROC™ Bluetooth® SDK (BTSDK) MCUs, you will be able to follow along with a Level 3 class that covers Bluetooth® in detail.

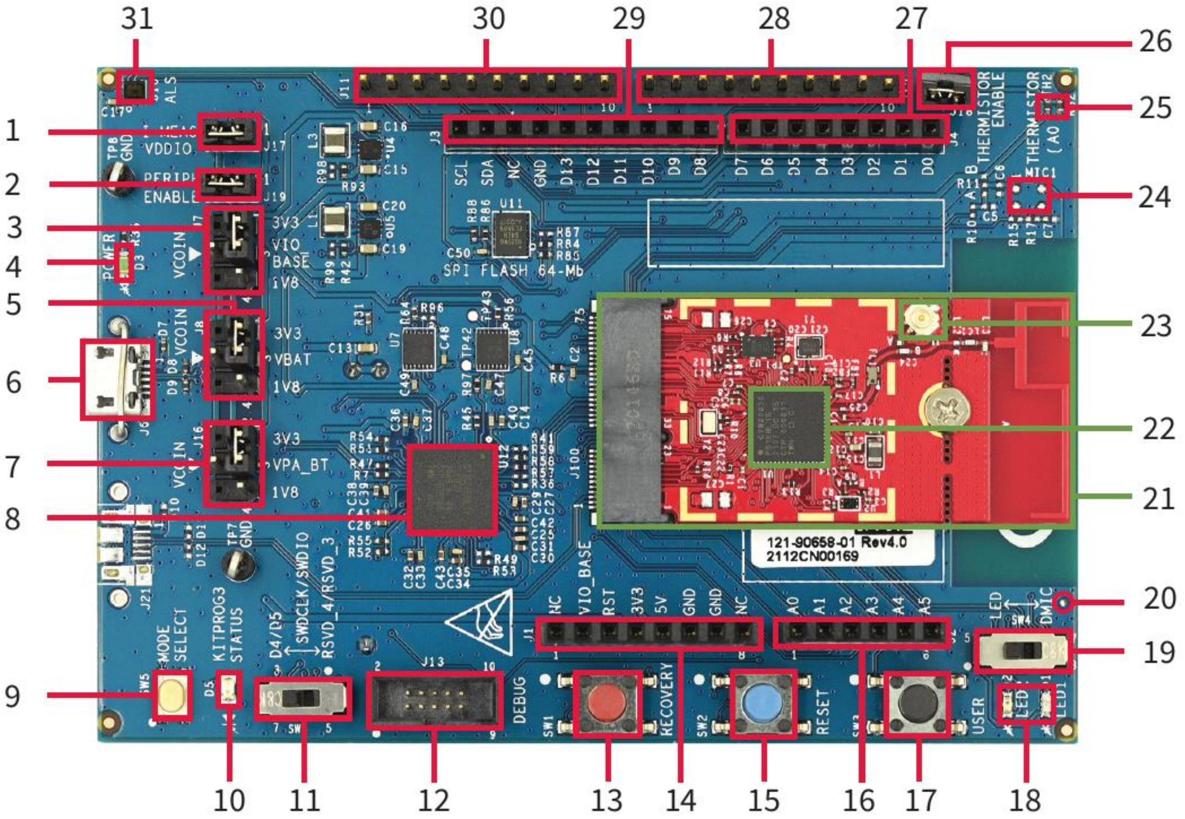
1.2 Prerequisites

- ModusToolbox™ Software Training Level 1 - Getting Started

This class will not cover what ModusToolbox™ software is, what tools it includes, or how to use any of its features. If you are unfamiliar with ModusToolbox™ software, you should take the "ModusToolbox™ Software Training Level 1 - Getting Started" class before embarking on this class.

1.3 Development kit

For this class we will use the CYW920835M2EVB-01 development kit.



1. VDDIO current measurement jumper (J17)
2. Peripheral enable jumper (J19)
3. VDDIO select jumper (J7)
4. Baseboard power status LED (D3)
5. VBAT select jumper (J8)
6. USB connector for programming/ USB-UART (J6)
7. VPA select jumper (J16)
8. KitProg3 based on PSoC™ 5LP MCU (U12)
9. KitProg3 mode select (SW5)
10. KitProg3 status LED (D5)
11. Debug interface select jumper (SW8)
12. Debug header (J13)
13. Recovery button (SW1)
14. Header compatible with Arduino (J1)
15. Reset button (SW2)
16. Header compatible with Arduino (J2)
17. User button (SW3)
18. User LEDs (D1, D2)
19. User LED/DMIC switch (SW4)
20. Digital mic sound port (J16)
21. CYW920835M2IPA1 Bluetooth® M.2 radio card
22. AIROC™ CYW20835 Bluetooth® LE system-on-chip (CYW920835M2IPA1.U1A)
23. External antenna connector (CYW920835M2IPA1.J1)
24. Analog mic footprint (MIC1)
25. Thermistor (TH2)
26. Thermistor enable jumper (J18)
27. Header compatible with Arduino (J4)
28. Bluetooth® I/O header (J12)
29. Header compatible with Arduino (J3)
30. Bluetooth® I/O header (J11)
31. Ambient light sensor (U10)

1.4 Introduction to AIROC™ Bluetooth® SDK (BTSDK) MCUs

AIROC™ Bluetooth® SDK (BTSDK) MCUs provide a single chip solution for all of your Bluetooth® application needs. These devices support Bluetooth® Low Energy, and in some cases Bluetooth® classic and Bluetooth® mesh.

The complete family of devices can be found at:

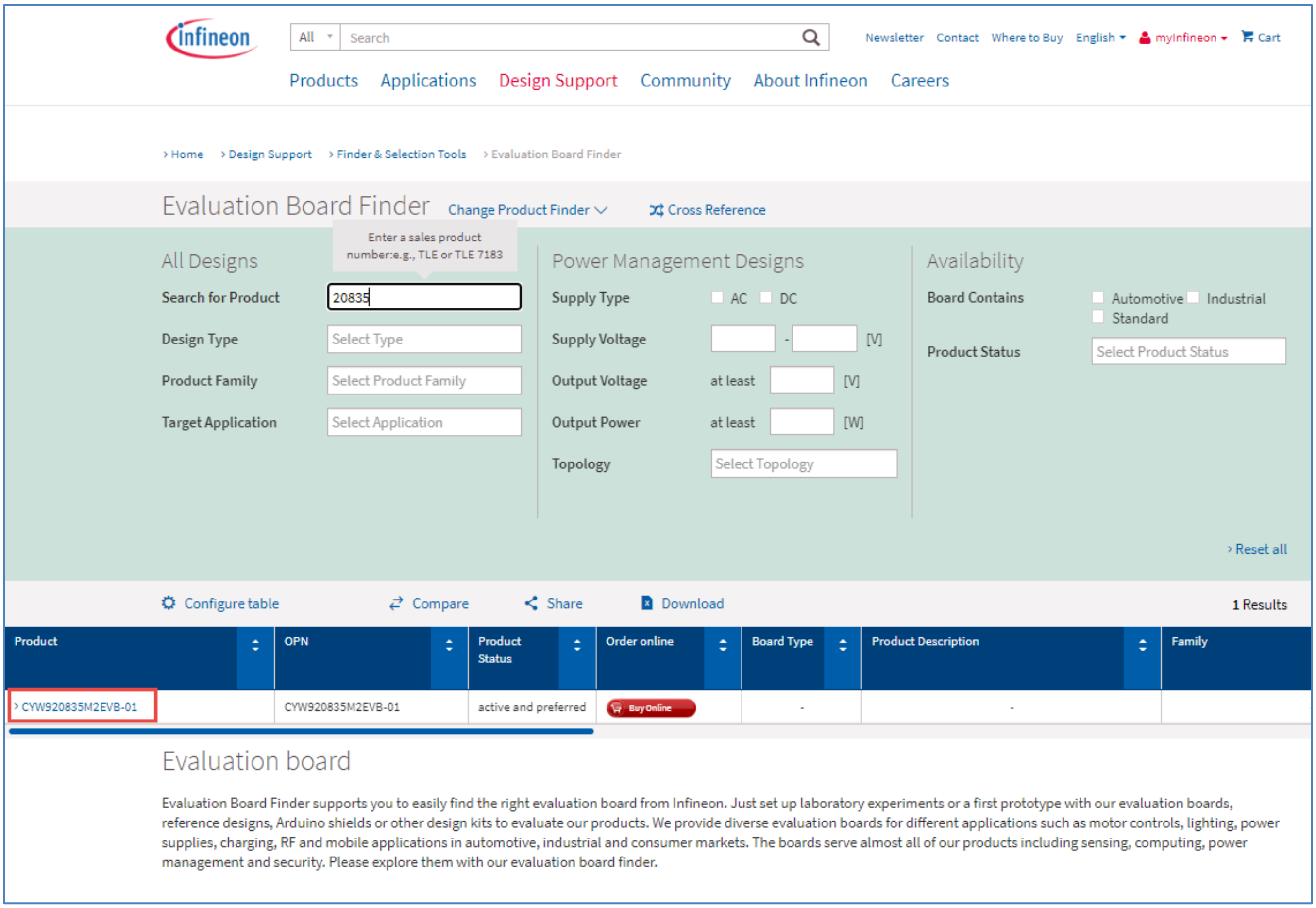
https://www.infineon.com/cms/en/product/wireless-connectivity/airoc-bluetooth-le-bluetooth-multiprotocol/

1.4.1 Development kits and Evaluation boards

Many development kits are available for these devices. To find the list of kits, visit:

https://www.infineon.com/cms/en/design-support/finder-selection-tools/product-finder/evaluation-board/

Enter the device part number (or a subset of it) in the "Search for Product" field to see a list of available kits. For example, you can enter "20835":



Search for Product: 20835

1 Results

Product	OPN	Product Status	Order online	Board Type	Product Description	Family
› CYW920835M2EVB-01	CYW920835M2EVB-01	active and preferred	Buy Online	-	-	

Evaluation board

Evaluation Board Finder supports you to easily find the right evaluation board from Infineon. Just set up laboratory experiments or a first prototype with our evaluation boards, reference designs, Arduino shields or other design kits to evaluate our products. We provide diverse evaluation boards for different applications such as motor controls, lighting, power supplies, charging, RF and mobile applications in automotive, industrial and consumer markets. The boards serve almost all of our products including sensing, computing, power management and security. Please explore them with our evaluation board finder.

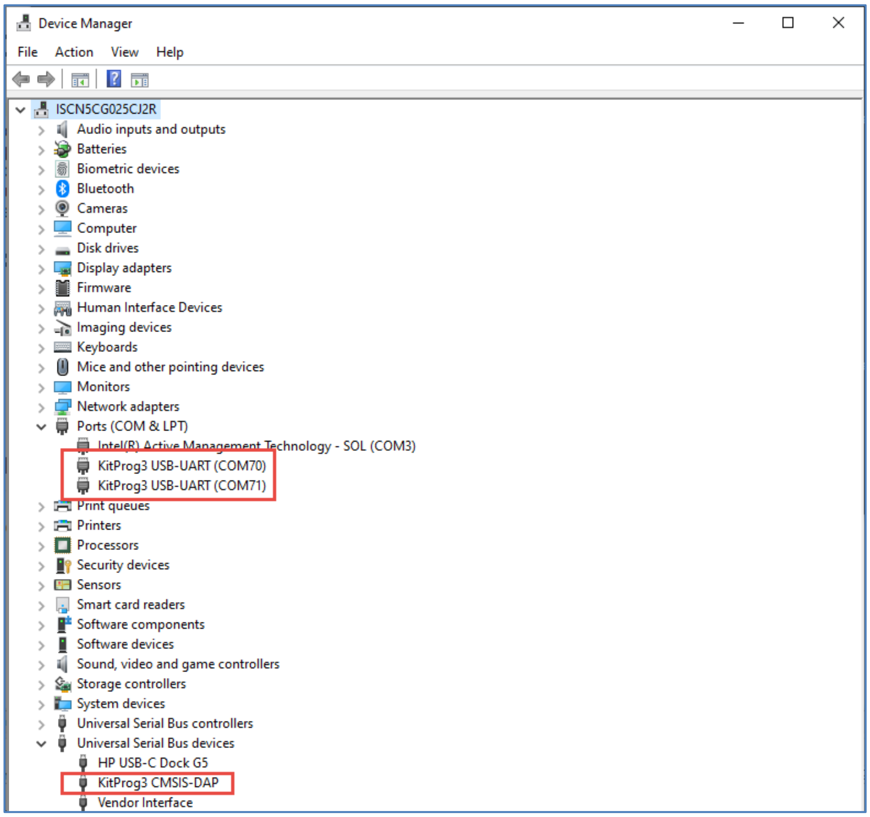
1.5 Infineon Kit Programmer/USB Bridge

The kit we are using has a PSoC™ 5LP chip on it that is used as a programmer and USB bridge. It provides the capability for your computer's USB to connect to the target device using multiple protocols such as Serial Wire Debug (SWD), UART, and I^{2}C. The host application on your computer needs to talk to the programmer to download firmware into the flash and to debug the MCU.

There are a bunch of different protocols out there for debugging MCUs. However, a few years ago Arm developed a standard called CMSIS-DAP, which has two variants (Bulk and DAPLink).

The programmer will appear to your computer to be multiple USB endpoints that implement each of the functions described in the previous paragraphs. In the case of the kit we are using, the programmer will appear as three USB endpoints: one CMSIS-DAP Bulk endpoint, and two UART endpoints. One of the UART endpoints is used mainly for programming (called the HCI UART) while the other is used for printing messages to a serial terminal emulator window (called the Peripheral UART or PUART).

On a Windows PC, you can see these in the Device Manager:



```
ISCN5CG025CJ2R
  Audio inputs and outputs
  Batteries
  Biometric devices
  Bluetooth
  Cameras
  Computer
  Disk drives
  Display adapters
  Firmware
  Human Interface Devices
  Imaging devices
  Keyboards
  Mice and other pointing devices
  Monitors
  Network adapters
  Ports (COM & LPT)
    Intel(R) Active Management Technology - SOL (COM3)
    KitProg3 USB-UART (COM70)
    KitProg3 USB-UART (COM71)
  Print queues
  Printers
  Processors
  Security devices
  Sensors
  Smart card readers
  Software components
  Software devices
  Sound, video and game controllers
  Storage controllers
  System devices
  Universal Serial Bus controllers
  Universal Serial Bus devices
    HP USB-C Dock G5
    KitProg3 CMSIS-DAP
    Vendor Interface
```

1.5.1 Recovery Mode

Sometimes when attempting to program the device, the programmer will be unable to put the MCU into the correct mode to transfer the new target device firmware over the programming UART. In order to force the device into programming mode, use the following steps:

1. Press and hold the **RECOVERY** button on the board.
2. Press and then release the **RESET** button on the board.
3. Release the **RECOVERY** button.

1.6 Firmware Loader

Firmware loader (fw-loader) is a tool that we deliver as part of ModusToolbox™ software. It is a command-line tool that allows you to install new KitProg firmware onto a PSoC™ kit, and switch modes programmatically.

You can find it in the following directory:

<install_dir>/ModusToolbox/tools_<version>/fw-loader/bin

It is also available independently on GitHub at http://github.com/infineon/firmware-loader.

The following table shows some of the basic fw-loader commands:

Command	**Description**
`fw-loader --help` (or no argument)	Print out help information.
`fw-loader --device-list`	List all the KitProg devices attached to your computer.
`fw-loader --update-kp3`	Install the latest firmware onto your KitProg.
`fw-loader --mode kp3-daplink`	Put the KitProg into DAPLink CMSIS-DAP mode.
`fw-loader --mode kp3-bulk`	Put the KitProg into CMSIS-DAP Bulk mode.
`fw-loader --mode kp3-hid`	Put the KitProg into CMSIS-DAP HID mode (not typically used).
`fw-loader --mode kp3-bootloader`	Put the KitProg into bootloader mode (not typically used).

You will practice using this tool in exercise 2.

1.7 Exercises

Exercise 1: Look at kit documentation

□ 1. Visit the websites for the kit used in this class and familiarize yourself with the kit documentation.

Exercise 2: Update Programmer firmware

Your kit may have an older version of the programmer firmware. In this exercise we will determine what version your kit has and update it if necessary.

□ 1. Connect your kit to your computer using the provided USB cable. If your kit has multiple USB connectors, be sure to connect to the one labeled KITPROG.

□ 2. Run a command line terminal.

Windows: run `modus-shell` from the **Start** menu or enter `modus-shell` in the Windows search box.
MacOS or Linux: start a standard command terminal

□ 3. Navigate to the directory *<install_dir>/ModusToolbox/tools_<version>/fw-loader/bin*

Note: On Windows, steps 2 and 3 can be done together by either running `fw-loader` from the Start menu list or by entering `fw-loader` in the Windows search box.

□ 4. Check the firmware version on the kit:

```
./fw-loader --device-list
```

□ 5. Does a device show up? Is it in KitProg3 CMSIS-DAP Bulk mode? What version of the firmware does it have installed?

□ 6. If you see a message that the firmware is outdated, update it to the latest version of KitProg3:

```
./fw-loader --update-kp3
```

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.