Chapter 1: Introduction

After completing this chapter, you will understand what this class is, what topics are covered, and the overall class objectives.

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 Why use machine learning?

This class is an introduction to the ModusToolbox™ Machine Learning (ML) solution. Three main reasons for using ML over traditional algorithmic programming are:

1. It reduces the time you typically spend programming. You can create a model, show it lots of examples and let the model complete the rest.
2. ML allows customization. For example, if you have a model that identifies pictures with stop signs, you might be able to retrain the model to identify traffic lights.
3. ML solves problems that can't be addressed with traditional programming. For example, while face recognition is easy for humans, writing an algorithm to do it would be nearly impossible.

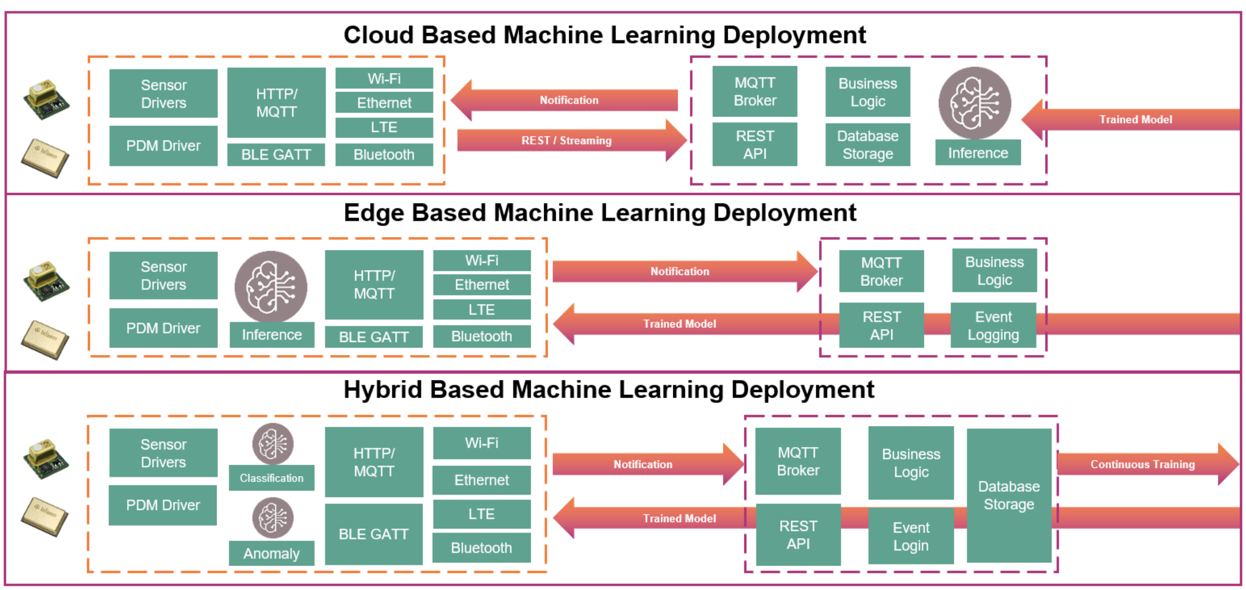
Note: *Not all problems are suited to machine learning solutions. In some cases, a traditional embedded program is much easier to create and will yield better results than a machine learning solution. The first step of any engineering problem is to understand it well enough to know what the right approach is.*

1.2 Machine learning on the edge

Historically, machine learning solutions have been accomplished using cloud computing. With this method, the IoT device interfaces with the sensor to collect data, but the data is streamed to the cloud for the machine learning solution to run (i.e. the inference engine).

With edge-based machine learning, the machine learning solution is run on the edge device, and only provides notifications to the cloud when significant events occur. The trained model can be hard-coded into the device or it can be updated via the cloud if necessary (i.e. over-the-air update).

An even more advanced hybrid option is able to run the machine learning solution on the edge device and also detect anomalies in the acquired data, passing them to the cloud for continuous training of the model.

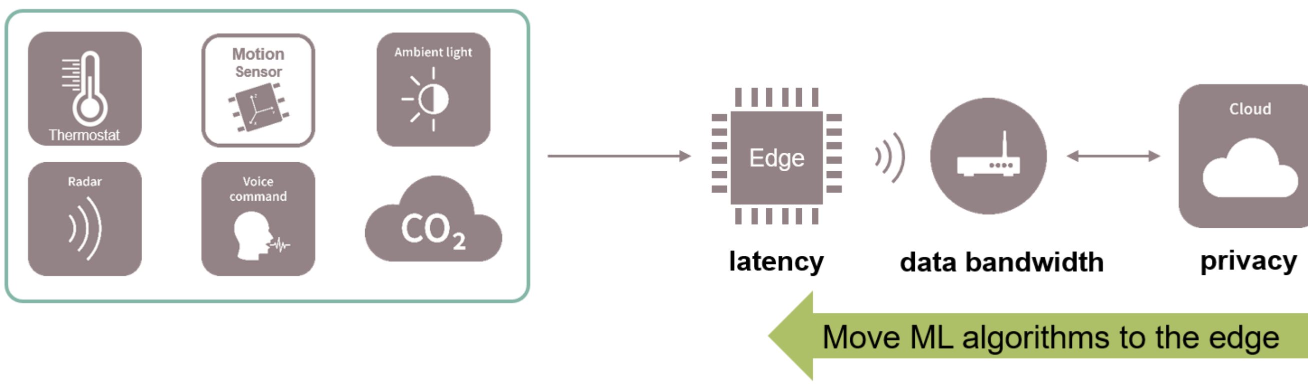


Note: *In all three methods described above, creation and training of the model is currently done on the cloud. The Edge device can run the ML solution to come up with classification results and detect anomalies, but (at least not today) it does not complete the model training steps.*

1.2.1 Why use machine learning on the edge?

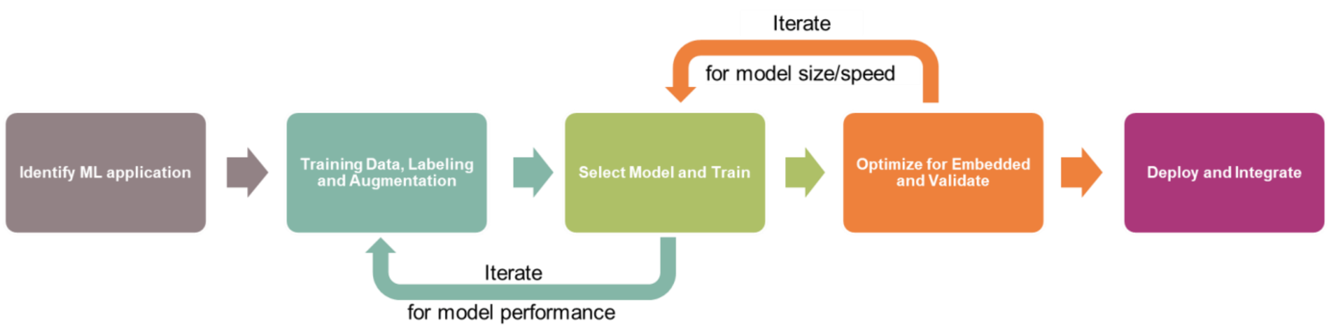
The ModusToolbox™ ML solution enables you to create, optimize, and deploy ML solutions on the "edge" using Infineon MCU devices. Being able to function independently from the cloud enables:

- Higher data bandwidth
- Less latency
- Better privacy
- Offline functionality
- Lower power
- Lower cost because of a reduced need for cloud computing and data storage



1.3 The ModusToolbox™ machine learning flow

The complete ModusToolbox™ ML flow is shown below:



The basic steps as shown in the figure are:

1. Identify the application that will use ML.
2. Collect training data required for the model.
3. Select or develop the model and train it using the training data.
4. Optimize (for size and performance) and validate the model for the chosen embedded MCU.
5. Deploy the model to the chosen embedded MCU and integrate it with the final user application.

ModusToolbox™ ML tools and software are used for optimization, validation, deployment, and integration (i.e. shown in steps 4 and 5 above). Steps 1-3 can be done manually using ML frameworks such as Keras with TensorFlow. You can also use Infineon ML partners such as EdgeImpulse, SensiML or Cyberon to make the process easier. In this class we will cover each step with descriptions and examples of how it can be accomplished for different applications.

1.4 Prerequisites

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

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" course before embarking on this course.

If you are unfamiliar with basic use of PSoC™ 6 MCUs including the use of HAL and PDL functions to control peripherals, you should at least review the first three chapters of the ModusToolbox™ Software Training Level 2 – PSoC™ MCUs class. You can focus on the PSoC™ 6 information and exercises.

1.5 References

For the latest information about Infineon's machine learning solution, visit:

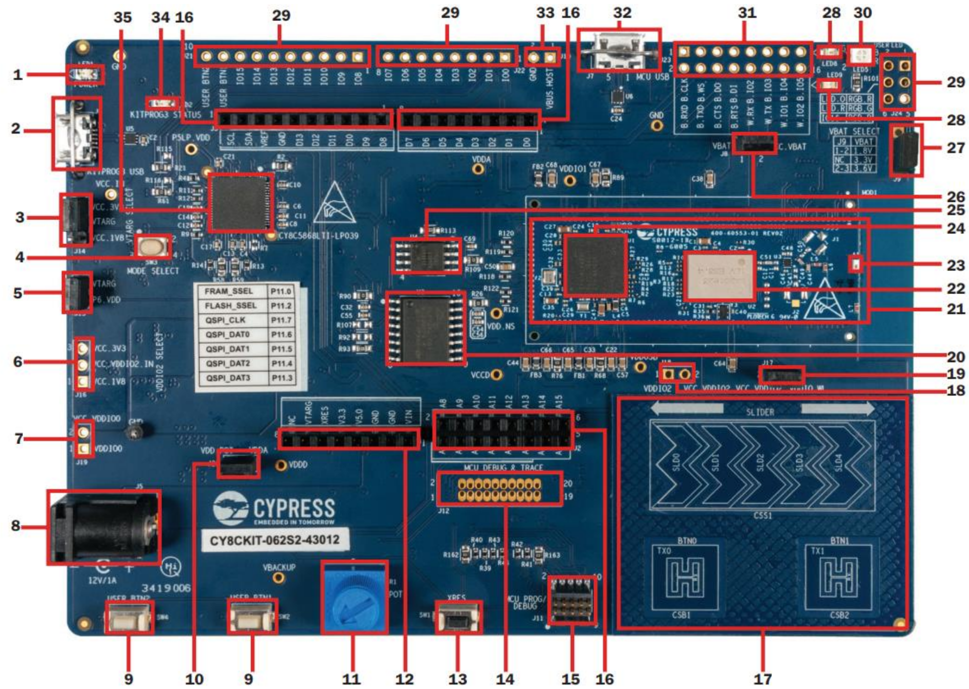
https://www.infineon.com/cms/en/design-support/tools/sdk/modustoolbox-software/modustoolbox-machine-learning/

1.6 Development Kits

For this class we will use the following development kits:

1.6.1 PSoC™ 6 kit

CY8CKIT-062S2-43012 – A PSoC™ 62 MCU and a CYW43012 Wi-Fi + Bluetooth® Combo.

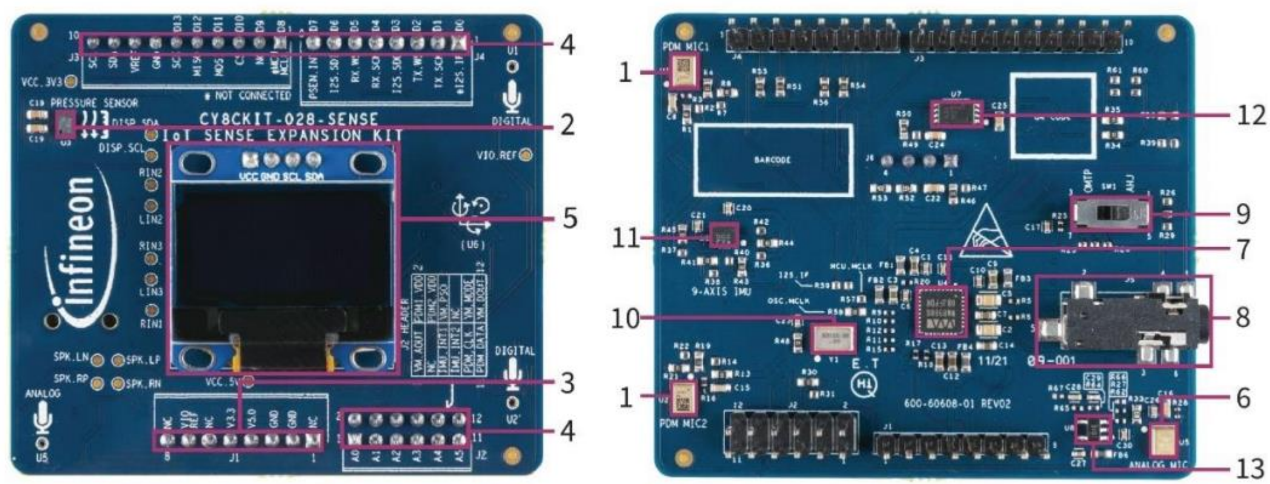


1. Power LED (LED1)
2. KitProg3 USB connector (J6)
3. PSoC™ 6 MCU VDD power selection jumper (J14)
4. KitProg3 programming mode selection button (SW3)
5. PSoC™ 6 MCU VDD current measurement jumper (J15)
6. PSoC™ 6 MCU VDDIO2 and CYW43012 VDDIO power selection jumper (J16)
7. PSoC™ 6 MCU VDDIO0 current 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)

Note: *Familiarize yourself with the locations of the white user buttons (9) and the black reset button (13). The reset button can be used to reset the kit. After reset, the programmed firmware will execute starting from the beginning.*

1.6.2 Arduino compatible shield

CY8CKIT-028-SENSE



1. XENSIV™ digital MEMS microphones (U1, U2)
2. XENSIV™ digital barometric air pressure sensor (U3)
3. Arduino™ UNO R3 compatible power header (J1)
4. Arduino™ UNO R3 compatible I/O headers (J2, J3, J4)
5. OLED module (ACC6)
6. Vesper piezoelectric MEMS analog microphone (U5)
7. Audio codec (U4)
8. 3.5-mm stereo audio jack socket (J5)
9. Audio jack socket type selection switch (SW1)
10. Crystal oscillator (Y1)
11. Bosch 9-axis absolute orientation sensor (U6)
12. I2C level translator (U7)
13. Preamplifier (U8)

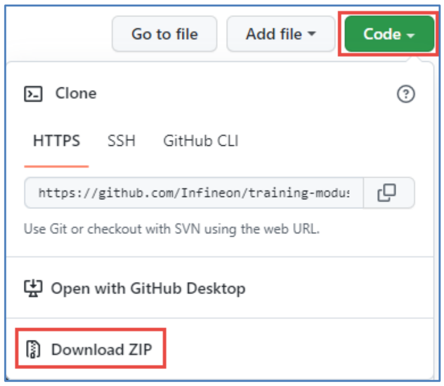
1.7 Exercises

Exercise 1: Download Class Material

In this exercise, you will download the class material from GitHub. This will give you local access to the manuals and projects.

☐ 1. Use a Web browser to go to the class GitHub site at: https://github.com/Infineon/training-modustoolbox-level3-machine-learning

☐ 2. Click the Code button.



☐ 3. Click the Download ZIP button to download the repo to your local disk to a convenient location and then unzip it.

Note: If you are familiar with Git operations, you have the option to clone the repository to your local disk using the URL instead of downloading a ZIP file.

Note: The remaining exercises are meant for installing the various tools that are used in the rest of this class. You can install everything now or wait and install them when needed. The instructions are repeated in the chapter where each tool is first used. The first two tools are used by the ModusToolbox™ Machine Learning Configurator while the others are used with Infineon's Machine Learning partners.

Exercise 2: Install ModusToolbox™ Machine Learning Technology Pack

☐ 1. You should already have ModusToolbox™ installed from either the Level 1 Getting Started or Level 2 PSoC™ MCUs class. If not, install it first.

You must have ModusToolbox™ version 3.0 or later.

☐ 2. Go to https://softwaretools.infineon.com/tools/com.ifx.tb.tool.modustoolboxpackmachinelearning to download and install the ModusToolbox™ Machine Learning pack.

Note: Install the pack in the same location that you installed ModusToolbox™. If you just installed ModusToolbox™ for the current user, repeat the same for the pack.

Exercise 3: Install QEMU

☐ 1. The ModusToolbox™ Machine Learning Configurator requires a third party tool called QEMU. Download and install it now.

Note: Instructions for downloading, installing, and configuring QEMU can be found in the Software requirements section of the ModusToolbox™ Machine Learning user guide. The default location of the user guide is ~/ModusToolbox/packs/ModusToolbox-Machine-Learning-Pack/help/ml-user-guide.pdf.

Note: You must install a specific version of QEMU. Be sure to follow the instructions in the section specific to your operating system rather than following the link to the top level QEMU site as that will point you to a different version.

Note: When creating the `IFX_MTBML_QEMU_PATH` variable, be sure there are no trailing spaces at the end of the variable name or path. You must use backslashes for the path (e.g. C:\Program Files\qemu).

Exercise 4: Install Cyberon tools

Cyberon is a machine learning partner that provides speech recognition models with no data collection necessary. You can skip this step if you don't plan to evaluate Cyberon.

- ☐ 1. Follow this link to the Cyberon DSpotter Modelling Tool (DSMT) page: https://tool.cyberon.com.tw/DSMT_V2/index.php?lang=en
- ☐ 2. The web page has some description on how to use the tool. Near the bottom of the page, there is a list of links to download it. Download the latest version, located at the top of the download list, and install it.

Note: Once the main tool finishes installing, you will have the option to install the online and offline test tools. The online tool allows you to test your models in real time using the microphone on your computer before downloading to the kit, and the offline tool allows you to test the model on your computer with existing audio files. Here, we will use the online tool, but it is safe to install both. If you don't install one or both of the test tools, there are links to each from the same web page to get at your convenience.

Note: DSMT is only available for Windows.

Exercise 5: Install SensiML tools

SensiML is a machine learning partner that provides tools for data collection and machine learning model creation. You can skip this step if you don't plan to evaluate SensiML.

- ☐ 1. Go to the following page: https://sensiml.com/download
- ☐ 2. Download and install SensiML Data Capture Lab.
- ☐ 3. Download and install SensiML Open Gateway.
- ☐ 4. Download and install SensiML Python SDK.

Note: SensiML tools are only available for Windows.

Note: When you install the Python SDK, run the pip command from modus-shell.

Note: SensiML Analytics Studio runs in the cloud, so no installation is required for that tool.

Exercise 6: Install Edge Impulse tools

Edge Impulse is a machine learning partner that provides tools for data collection and machine learning model creation. Most of Edge Impulse's tools are web-based, but Infineon's Cypress Programmer is used for programming their test firmware onto the PSoC™. You can skip this step if you don't plan to evaluate Edge Impulse.

- ☐ 1. Go to the following page: https://softwaretools.infineon.com/tools/com.ifx.tb.tool.cypressprogrammer
- ☐ 2. Download and install Cypress Programmer.

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

Published by
Infineon Technologies AG
81726 Munich, Germany