



MicroPython – Python for Embedded Controllers

Workshop Edge-AI with Infineon TUW

Infineon DES DOS SW ESW EPE

February 2026



Course Key Dates

Appointment	Date	Time	Location
MicroPython Hands-on Introduction	09.02.2026	10:00	On-site
DEEPCRAFT™ Studio Hands-on Introduction	10.02.2026	09:00	On-site
Presentation of Project Ideas	10.02.2026	09:00	On-site
Mandatory Check-in & Q&A Session	13.02.2026	16:00	Online
Mandatory Check-in & Q&A Session	17.02.2026	16:00	Online
Mandatory Check-in & Q&A Session	19.02.2026	16:00	Online
Final Submission	22.02.2026	23:59	On-site

Agenda

10:00 – 12:00

10:00 – 10:30

Morning session

Hello and Welcome!

10:30 – 11:00

Hardware Overview

11:00 – 11:30

MicroPython Introduction

11:30 – 12:00

MicroPython Docs and Resources

12:00 – 13:00

Lunch break

13:00 – 17:00

Afternoon session

13:00 – 14:00 Getting Started with MicroPython

14:00 – 15:45

MicroPython Features Overview

15:45 – 16:45

DEEPCRAFT™ AI Models in MicroPython

16:45 – 17:00

Wrap-up and Goodbye!

Workshop Goals

After completing this workshop, you will be able to:

- 1. Be familiar with the Infineon PSOC6™ CY8CKIT-062S2-AI board.**

Understand its features, capabilities, and where to find relevant documentation and resources.

- 2. Understand MicroPython basics and its available resources.**

Learn what MicroPython is, why you'd want to use it, and how to access the resources needed to continue working with it independently.

- 3. Work effectively with MicroPython tools and features.**

Use the basic toolchain, operate in different MicroPython modes, and leverage the available features for your projects.

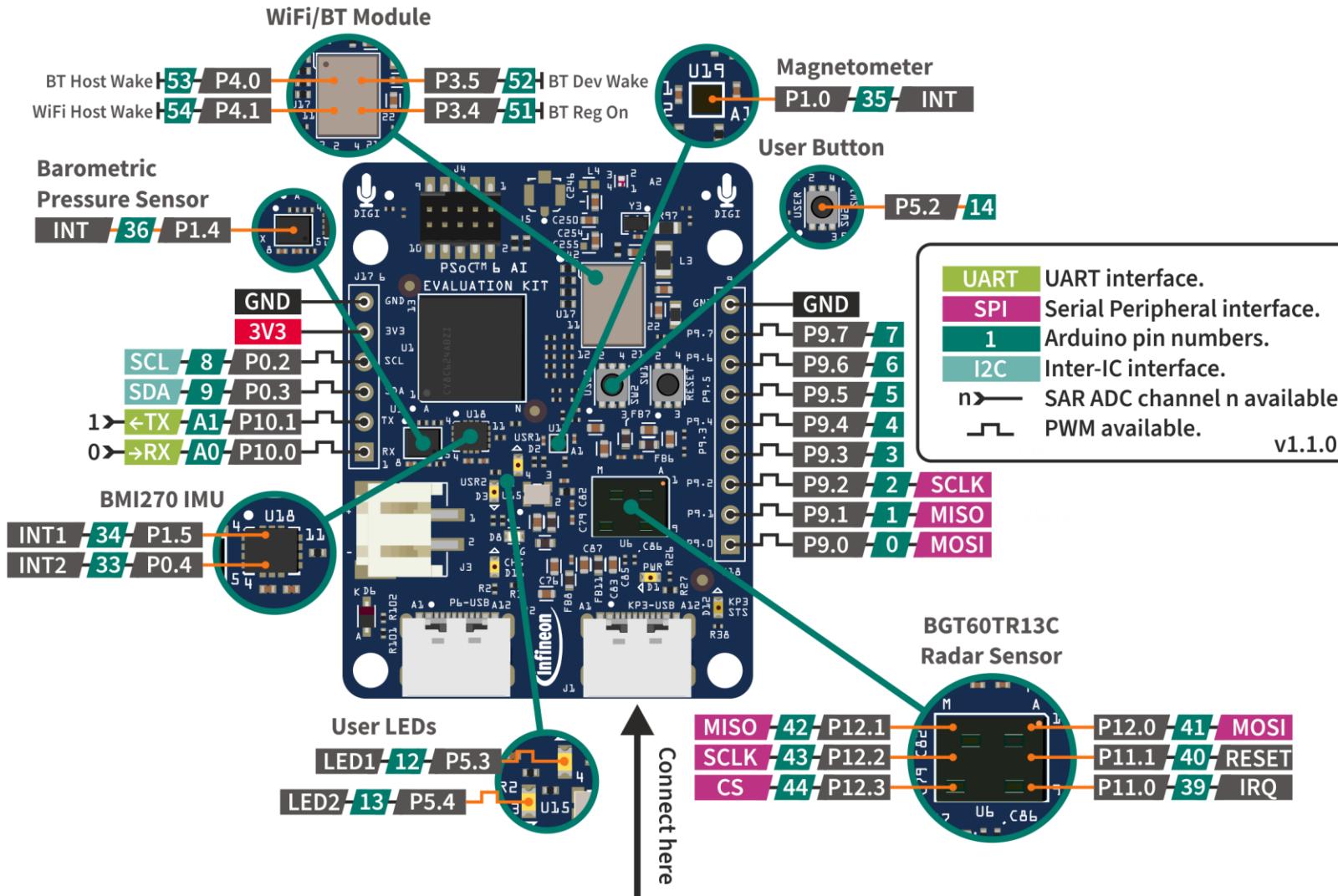
- 4. Develop your own MicroPython applications.**

Create basic applications using the CY8CKIT-062S2-AI board's hardware resources like sensors, actuators, and other components.

- 5. Integrate DEEPCRAFT™ AI into MicroPython projects.**

Understand what DEEPCRAFT™ Studio is and learn how to include and use DEEPCRAFT™ AI models in your MicroPython applications.

The hardware – the Infineon CY8CKIT-062S2-AI board



Features overview

- Dual core: CM4F + CM0+
- Memory: 2 MB Flash + 1 MB SRAM + 512 Mbit External Flash
- Connectivity: Wi-Fi 11 + BLE 5.1
- 1.8V and 3.3V Operation
- 2 User LEDs
- USB On-board KitProg3 Debugger
- USB Device Connector
- User and Reset Button
- Li-ion Battery Connector
- Sensors:**
 - 6-axis motion (BMI270)
 - Magnetometer (BMM350)
 - Barometric Pressure (DPS368)
 - Radar (BTG60TR13C)
 - Microphones (IM69D) x 2

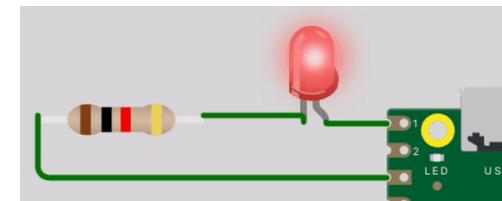
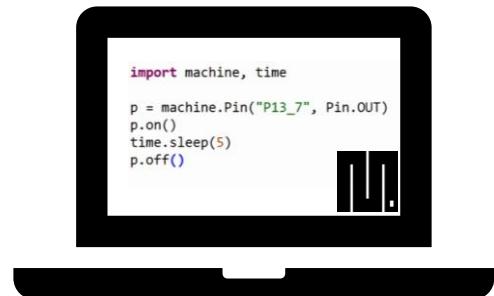
What is MicroPython?



Lean implementation
of Python3...

...optimized for small
footprint MCUs...

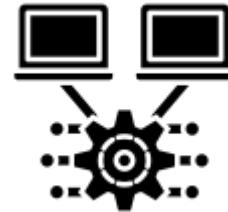
...interactive and
extendable.



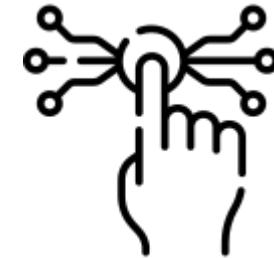
Why MicroPython?



Beginner friendly



Platform Compatibility



Interactivity



Extensive Library support



Rapid Prototyping



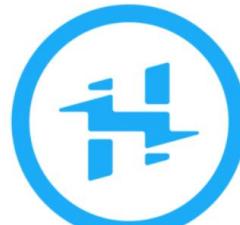
Integration with existing
Python Ecosystem

Where do I learn about MicroPython?



[MicroPython Official Docs](#)

[MicroPython Infineon PSOC6™ Docs](#)



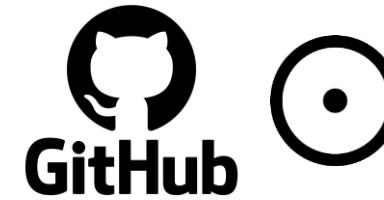
hackster.io

[Community Example Applications and Projects](#)

[Infineon Team Tutorial and Projects](#)



[MicroPython Official Website](#)



[MicroPython Github Official Issues](#)

[MicroPython Github Infineon Issues](#)



[MicroPython Github Official Discussions](#)

[MicroPython Github Infineon Discussions](#)



[MicroPython Community Forums](#)

How do I install MicroPython?

System requirements

- Python 3.x
- pip

From [Quick reference for the PSOC6™ > Installing Micropython](#)

1. In a terminal, create a folder and change directory:

```
mkdir mpy-installation  
cd mpy-installation
```

2. Run the following command:

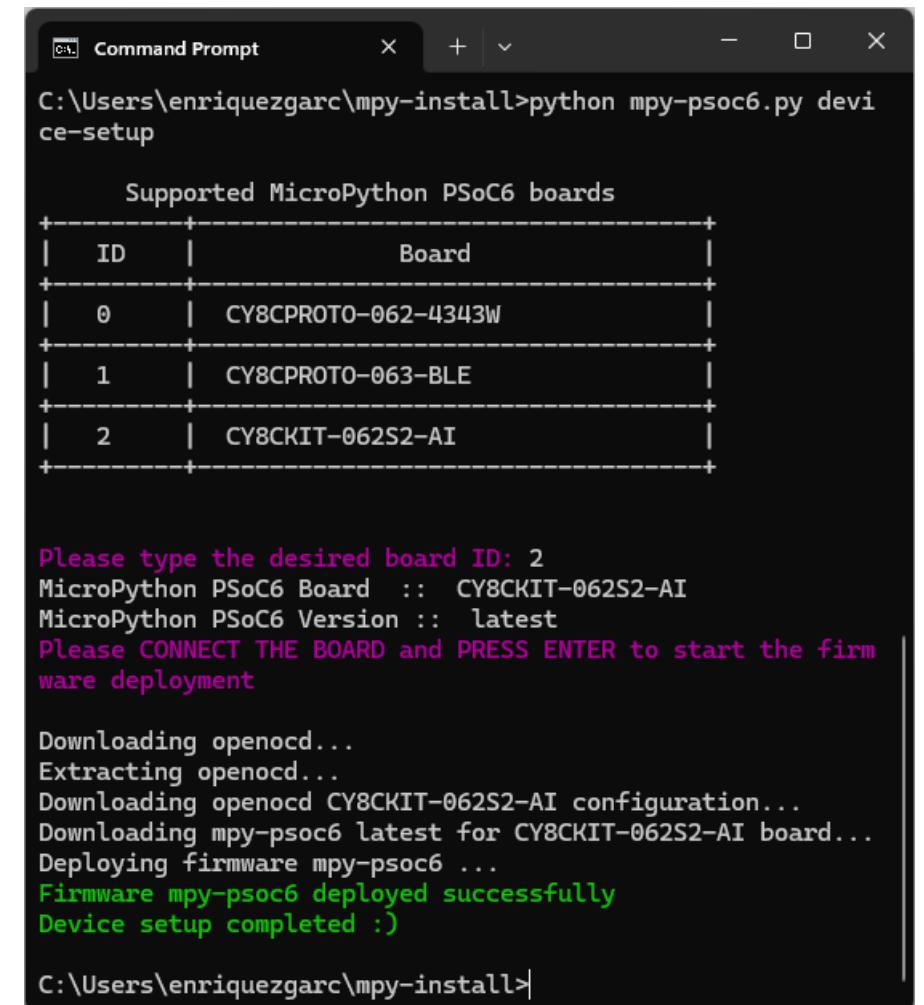
```
curl -s -L  
https://raw.githubusercontent.com/infineon/micropython/ports-psoc6-main/tools/psoc6/mpy-psoc6.py > mpy-psoc6.py
```

3. Install the requests library

```
pip install requests
```

4. Run the installation script:

```
python mpy-psoc6.py device-setup
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The path "C:\Users\enriquezgarc\mpy-install>" is visible at the top. The user runs the command "python mpy-psoc6.py device-setup". The script lists supported boards:

ID	Board
0	CY8CPROTO-062-4343W
1	CY8CPROTO-063-BLE
2	CY8CKIT-062S2-AI

The user selects board ID 2. The script then prompts for connection and deployment:

```
Please type the desired board ID: 2  
MicroPython PSoC6 Board :: CY8CKIT-062S2-AI  
MicroPython PSoC6 Version :: latest  
Please CONNECT THE BOARD and PRESS ENTER to start the firmware deployment
```

It then performs the deployment process:

```
Downloading openocd...  
Extracting openocd...  
Downloading openocd CY8CKIT-062S2-AI configuration...  
Downloading mpy-psoc6 latest for CY8CKIT-062S2-AI board...  
Deploying firmware mpy-psoc6 ...  
Firmware mpy-psoc6 deployed successfully  
Device setup completed :)
```

Finally, it shows the command prompt again:

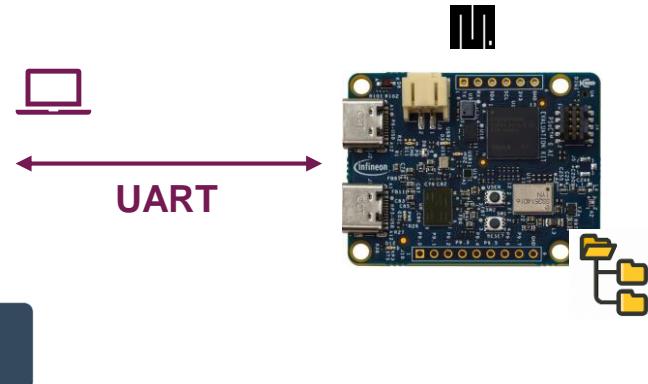
```
C:\Users\enriquezgarc\mpy-install>
```

How do I use MicroPython?

Serial Terminal (REPL)

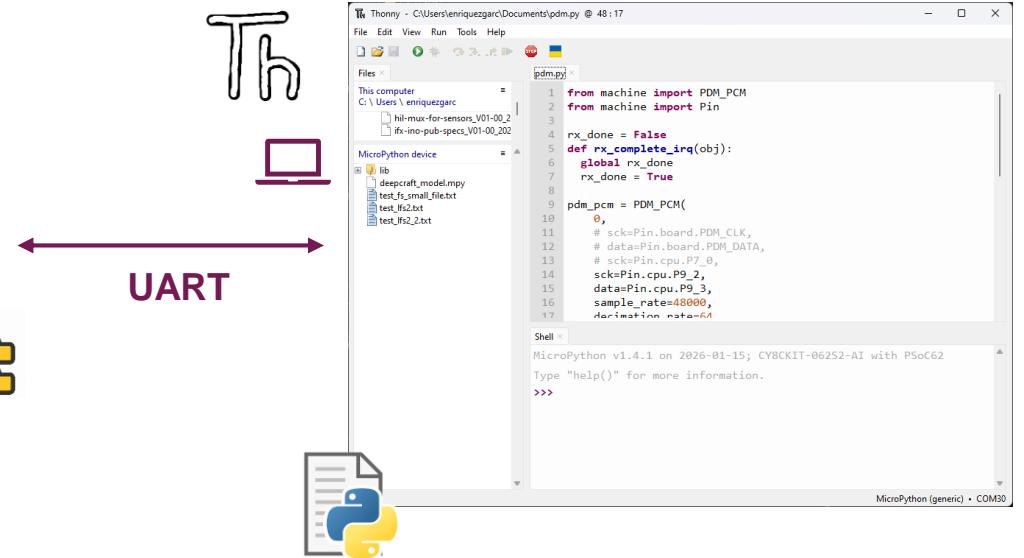
- Interacting with the Read-Eval-Print-Loop
- Using a Serial terminal client

```
MicroPython v0.3.0 on 2023-04-14; CY8CPROTO-062-4343W with PSoC62
Use Ctrl-D to exit, Ctrl-E for paste mode
LFS2 filesystem mounted at /
MicroPython v0.3.0 on 2023-04-14; CY8CPROTO-062-4343W with PSoC62
Type "help()" for more information.
>>> import machine
>>> 
```

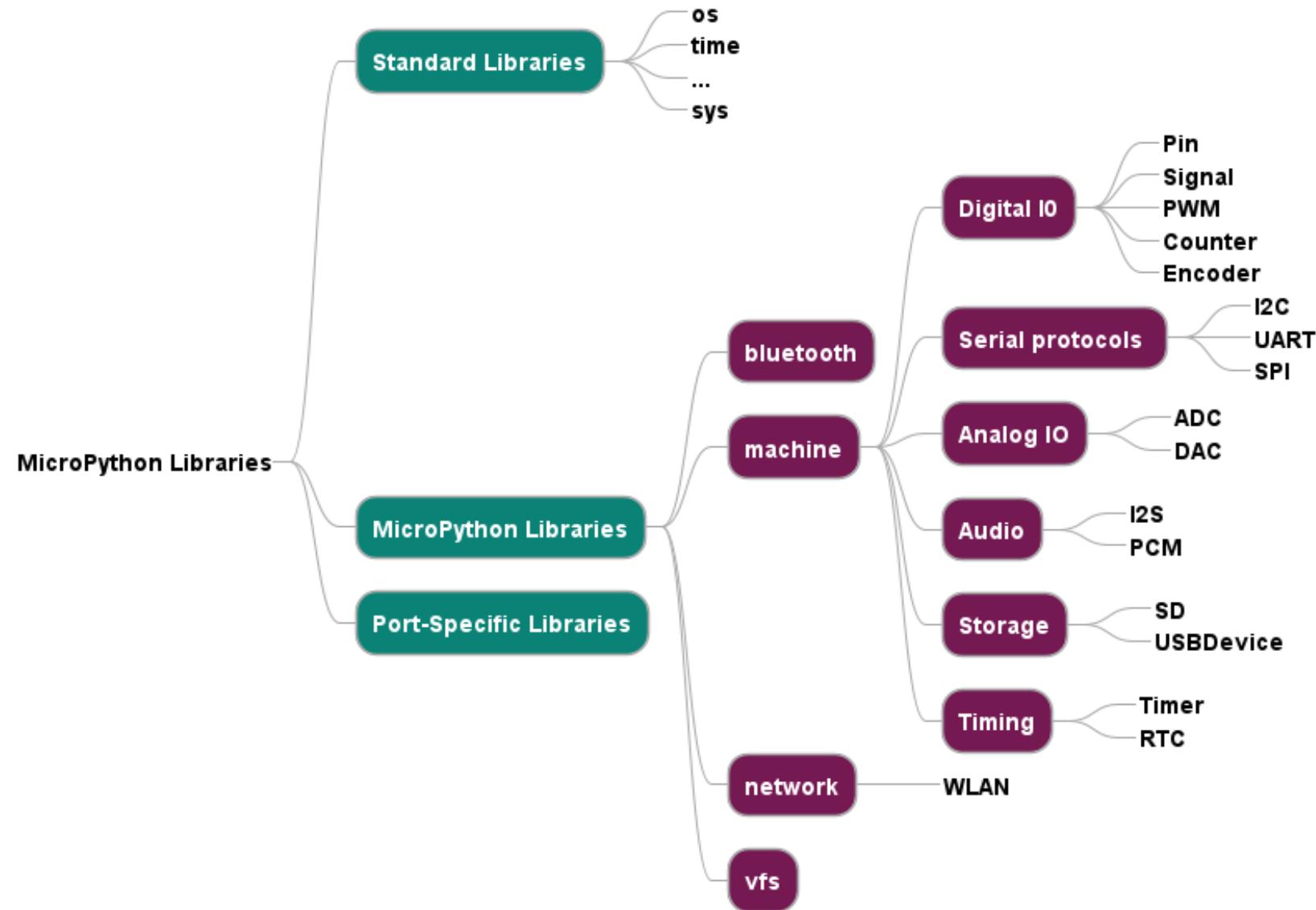


Script Modes (IDEs)

1. REPL raw mode
2. Upload scripts to the File System

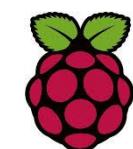


Which **libraries** are available in MicroPython?

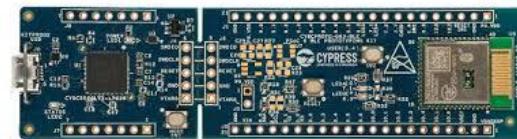


Which **devices** are supported in MicroPython?

- MicroPython is not vendor specific (cross-vendor)
- The most relevant ones:
 - STM32
 - RaspberryPi Pico
 - ESP32/ESP8266
 - Infineon PSOC6™
 - Infineon PSOC™ Edge (Coming soon ☺)
 - Zephyr RTOS



You can even run MicroPython in a Desktop **Unix** and **Windows** → But no hardware (machine, network, etc) features are available



How to use external libraries in MicroPython?

Installing external libraries options:

1. Copy file manually to the file system on

1. Using IDE Graphical Interface
2. mpremote (command line tool)

```
mpremote connect /dev/ttyUSB0 cp python-stdlib/base64/base64.py :/lib
```

2. mip package manager

1. From the PC

```
mpremote connect /dev/ttyUSB0 mip install package-name
```

2. From the script itself (on network-enabled devices)

```
import mip  
mip.install("package-name")
```

How to develop applications in MicroPython?

Let's build and run some examples together:

1. Blinky LED + Button
 - Intro to `machine` module and `Pin` class and `PWM`
 - Intro to Timer module
2. Digital Pressure Sensor Measurement
 - Intro to `machine` serial protocol `I2C`
 - Example of external library `mip` import
3. Audio Recording
 - Intro to `machine` `PCM` audio protocol
 - Example of `VFS` usage for file storage
4. Basic HTTP Server
 - Intro to `network` `WLAN` class
 - Usage of `boot.py` script

How do I **contribute** to MicroPython?

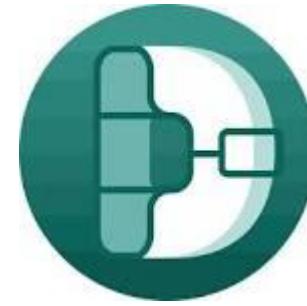
As MicroPython **user**:

- Engage in forum and discussions
- Report bugs or issues you encounter
- Request features
- Support others in the community

As MicroPython **developer**:

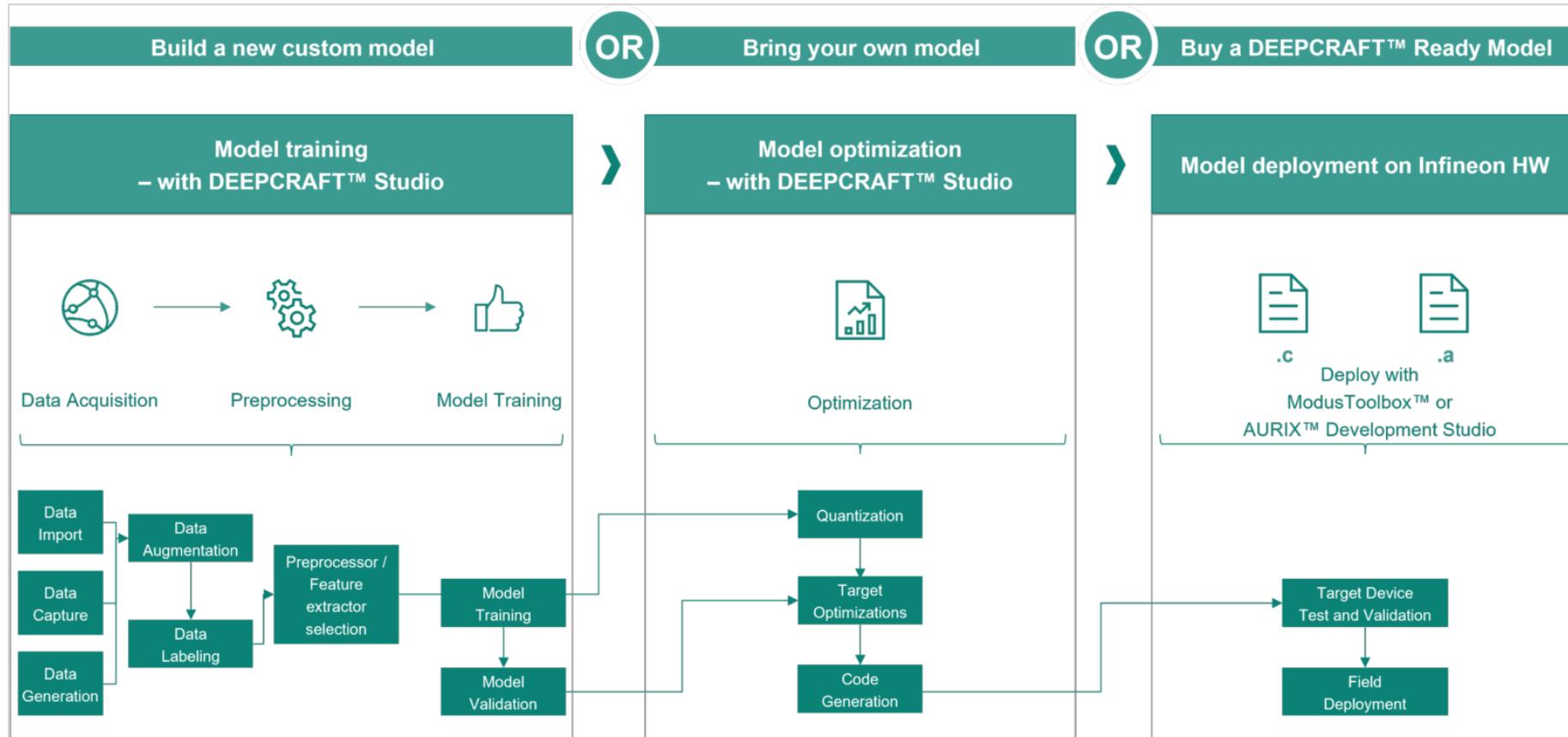
- Open source → Follow the [guidelines](#) to contribute
- Fix bugs in code and docs
- Extend existing functionalities
- Develop additional modules and tools for Micropython ecosystem
- Share your projects with the community

What is DEEPCRAFT™ ?

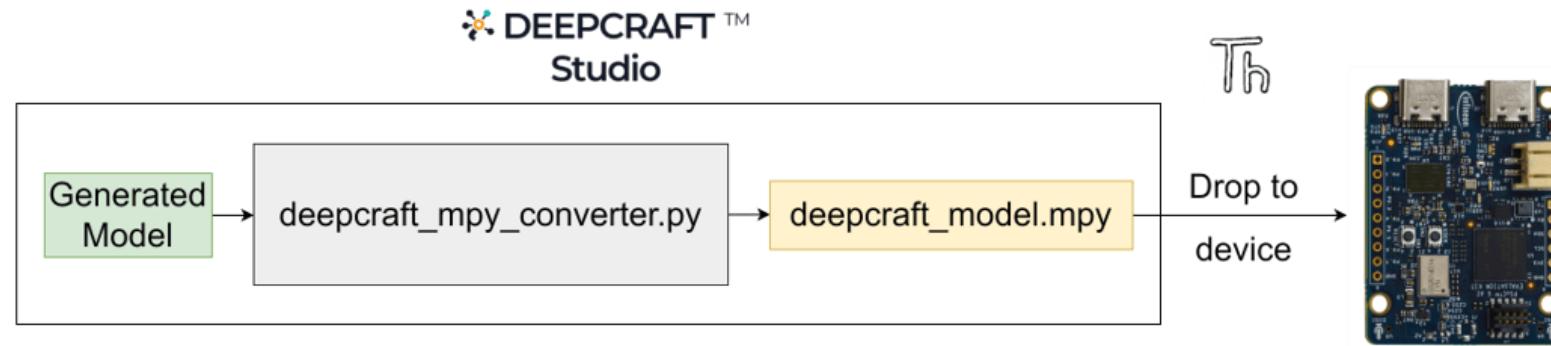


- Comprehensive **solution for developing AI** and machine learning **applications on edge devices**
- **End-to-end** development platform for machine learning
- Covers the **entire machine learning workflow**, including:
 - Collecting and annotating high-quality data
 - Managing, analyzing, and processing data
 - Building, evaluating, and selecting the best models
 - Deploying models on target edge devices

DEEPCRAFT™ User Journey



How do I integrate DEEPCRAFT™ AI models in MicroPython?



Steps from [Hackster.io tutorial](#):

1. After installing all required tools
 - [DEEPCRAFT™](#)
 - [DEEPCRAFT™ MicroPython Converter](#)
 - [GnuWin32 Make](#)
2. Generate the model in DEEPCRAFT™
3. Convert the .h/.c model to .mpy library
4. Upload to the .mpy file to the edge device
5. Use [model API](#) in your MicrPython application

