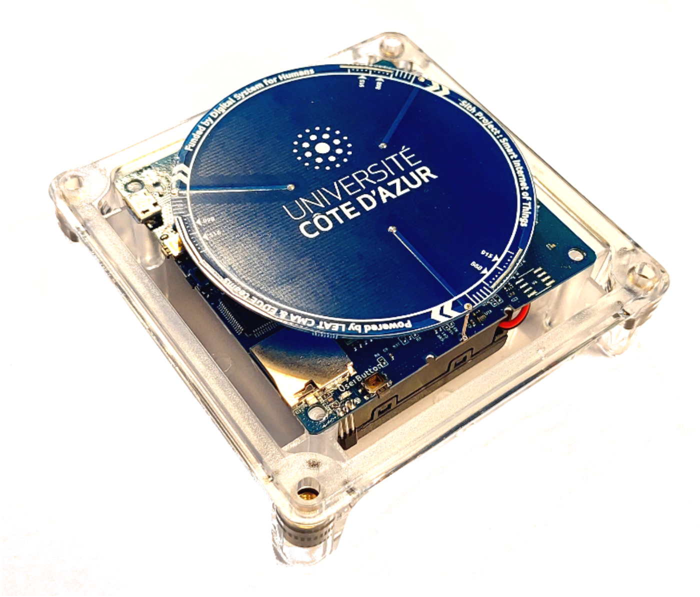
Lab I : Embedded Artificial Intelligence on microcontroller – Setup environment

Polytech Nice Sophia

During this lab, we will prepare the software environment needed to program AI algorithms onto MCU (MicroController Unit).This environment will be used during next Labs.

Hardware target

The target is a RFThings-AI Dev Kit board equipped with a STM32L476RGT6 Microcontroller. This MCU is based on the ARM Cortex M4 architecture and runs at a frequency of 80 MHz. The board provides 1 MB Flash and 128 KB SRAM.



Software environment

The following installation guide is targeting a **windows 10** operating system and will also use **docker**. See annex A for installation on Linux.
You must be sure to have at least 10 GB of free disk space for the installation of the tools. If you need space, you can for example uninstall Quartus that is not needed for the last step of the course.
We will mainly use the following software tools that you can download from a high bandwidth connection:
Docker desktop [512 Mo] from
https://hub.docker.com/editions/community/docker-ce-desktop-windows (no registration).
Note: Since the last version, Docker works with Windows 10 Pro Enterprise, Education, or Windows Home.

- Launch Docker Desktop. If you have User access limitation you have to add the docker-users group to your user account. Launch the compmgmt program (“Gestion de l’ordinateur”) from the start menu and go to users and groups (“Utilisateurs et groups”), in Users select your user account. In the member of ("Membre de") panel, click on Add (Ajouter) and add the docker-users groupe in the Object list. Log-out and in again from windows.

- Launch Docker again and Install the WSL 2 package (if missing) by following the instructions 4 and 5 from the link below:

```
https://docs.microsoft.com/fr-fr/windows/wsl/install-manual#step-4---download-the-linux-kernel-update-package
```

Installation steps

1. Connect to moodle, and download the provided TensorFlow script to train (during the lab) a predefined neural network (extract AI_first_notebook.zip that will provide the "d1-notebooks" folder).
2. Open a windows powershell and go to the directory where you unpacked the AI_first_notebook.zip archive. With the "ls" command you should see the d1_notebooks folder in this directory. Type the following command that will download the tensorflow image the first time you type it:

```
docker.exe run -it --rm --name jupyter-tensorflow  -p 8888:8888 -e JUPYTER_ENABLE_LAB=yes -v
$((pwd).tostring()+"/d1-notebooks:/home/jovyan/test") jupyter/tensorflow-notebook
```

Linux users:

```
docker run -it --rm --name jupyter-tensorflow -p 8888:8888 -e JUPYTER_ENABLE_LAB=yes -v
"$(pwd)"/d1-notebooks:/home/jovyan/test jupyter/tensorflow-notebook
```

If you have the following error : docker: invalid reference format, please type by hand the preceding command (no copy/paste)

3. Accept the request for access rights from Docker
4. Copy the link beginning by http://127.0.0.1... provided in the powershell and paste it in a web browser (Tested with Firefox 74.0).
5. Open the script provided in the test directory and run all cells.
6. Close the notebook panel and shutdown the Jupiter lab server using the "ctrl+c" combinaison in the powershell window.

You can just have a look to the code. Explanations will come during the course and the lab.

Hardware environment: Installation of the RF Things Board

Arduino IDE

Download and install Arduino IDE 1.8.16 (or newer) from https://www.arduino.cc/en/software.

Install the support package for the board with an STM32L4 microcontroller:

- Open the *Preferences* panel from the *File* menu.
- Add the following URL to the *Additional Boards Manager URLs*:
  https://rfthings.github.io/ArduinoBoardManagerJSON/package_rfthings-stm32l4_index.json
- Open the *Board Manager* from the *Tools* → *Board:* menu.
- Search for *STM32L4*.
- Select and install *RFThings STM32L4 Boards* (version 0.0.50-test).

Install the library for the inertial measurement unit sensor:

- Open the *Manage libraries...* panel from the *Tools* menu.
- Search for *20948*.
- Select and install *SparkFun 9D0F IMU Breakout - ICM 20948* (version 1.2.7 or newer).

Udev/Driver

Linux only

Place the udev rules file stm32dfu.udev.rules (from Moodle) in /etc/udev/rules.d, then reload udev rules:

```
sudo mv ~/Downloads/stm32dfu.udev.rules /etc/udev/rules.d/

sudo udevadm control --reload

sudo udevadm trigger
```

Windows only

- Download Zadig at https://zadig.akeo.ie/ and execute it.
- Put the board in bootloader mode (see Annex B).
- Select STM32 BOOTLOADER in the list.
- Make sure WinUSB driver is selected.
- Click Install Driver.
- Wait for it to finish.

MicroAI GUI

Ubuntu: install the package microai-gui_0.2.1-1_amd64.deb (from Moodle).
Windows: extract the package microai-gui_0.2.1_win32.zip (from Moodle).

Serial bridge

The serial bridge serial_client.py (from Moodle) requires Python 3 (3.6 or newer on Linux, 3.9 or newer on Windows) and the pyserial library.
Ubuntu: install the python3-serial package.
Windows: install Python 3.9 from https://www.python.org/downloads/ and type in a PowerShell prompt:

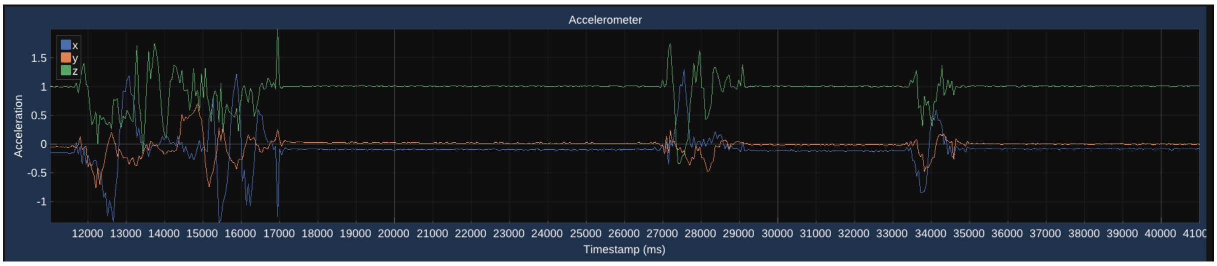
```
python3 -m pip install pyserial (under linux)

py -m pip install pyserial (under windows)
```

Deploy the Arduino sketch onto the board

- Open the *S4Lab5_collect_online* sketch from the S4Lab5_collect_online directory (from Moodle) with the Arduino IDE.
-
- In the *Tools* → *Boards* menu, select the *RFThings STM32L4 Boards* → *RFThings-DKAIoT* board.
- In the *Tools* → *Port* menu, select the serial port corresponding to the *RFThings Design KIT* board.
- Click on the right arrow in the toolbar to compile and deploy the sketch.
- After deployment, open the *Serial Monitor* from the *Tools* menu to confirm that the board is communicating.
- If some data is being received, close the *Serial Monitor*.

Collect and label data inside MicroAI GUI

- Open MicroAI GUI from the application menu under Ubuntu or by running the executable on Windows.
- Open a Terminal and run the *serial_client.py* script (as root using sudo under linux). You can pass the appropriate serial port as a parameter, such as COM4 on Windows (check in Device Manager). The accelerometer graph should start displaying realtime acceleration data as shown in figure below.

- 

- Capture some data and click on *Stop* once done.
- You can uncheck ***Auto fit X*** and ***Auto scroll***, and check ***Link plots X zoom to accelerometer*** to be able to zoom in/out on the accelerometer graph for easier labeling.
- Hold the *Shift* key and drag with middle click (or hold *Ctrl+Shift* and drag with right click) across the accelerometer graph to select a range of data.
- Click on *Add user label* and select *Positive* to mark positively labeled data. Every unlabelled data is assumed to be negative.
- Once all positive data has been labeled, click *Save* and enter a filename (keep *EllcieHAR* format).

Annex A - Installation guide under Linux

Tensorflow installation on Linux

Install Docker In the terminal:

```
sudo apt update

sudo apt install docker.io

sudo usermod -G docker -a $USER
```

Reboot and run JupyterLab with Python3/Tensorflow 2.1 on Docker
In the terminal:

```
setfacl -R -m u:1000:rwX,d:u:1000:rwX,d:u:$(whoami):rwX d1-notebooks

docker run -it --rm -p 8888:8888 -e JUPYTER_ENABLE_LAB=yes –v "$(pwd)"/d1-notebooks:/home/jovyan/test
jupyter/tensorflow-notebook
```

Open link `127.0.0.1/[…]` in the browser
Open `d1-notebooks/MNIST_LeNet-5.ipynb`
Click Cells menu → Run all

Annex: B: Bootloader/DFU mode

Bootloader or DFU mode is a special mode of the USB controller inside the STM32 microcontroller that allows reflashing a firmware (even if the current firmware is corrupt). To put the board into bootloader mode:

- Unplug the USB cable.
- Press and hold the BOOT button next to the microUSB port.
- Plug the USB cable in.
- The board should now be recognized as *STM Device in DFU Mode* or *STM32 BOOTLOADER*

Annex C : Common issues

C.1 Cannot open DFU device 0483:df11 (Windows)

See Section Udev/Driver

C.2 dfu-util: File not found (Linux)

Install the 32-bit glibc to run 32-bit binary. The package is called libc6-i386 on Ubuntu.

C.3 Board goes into bootloader mode but the countdown goes all the way up to 10 and fails. (Linux)

Put the board in bootloader mode (see Appendix A) and run:

```
~/.arduino15/packages/lacunaspace/hardware/stm32l4/0.0.28−lacuna−9/tools/linux/dfu−util −l
```

You should see Found DFU:

```
0483:df11
```

in the output If there is a permission error, see Section 1.1.1.

C.4 Permission denied while trying to open /dev/ttyACM0 (Linux)

Add your user to the **dialout** (or **uucp** on some distributions) group to obtain permissions for the serial port:

```
sudo usermod −G dialout −a $(whoami)
```

You need to log out and log back in after joining a group.

C.5 Input/output error, Port not found, acm_port_activate - usb_submit_urb(ctrl irq) failed (Linux)

If deployment fails with an error such as **Error opening serial port '/dev/ttyACM0'**. (Port not found), opening a terminal to the port fails **with Input/output error** and the kernel log (dmesg) shows an error similar to cdc_acm 1-5.3:1.0: acm_port_activate - usb_submit_urb(ctrl irq) failed, you may be hit by a bug introduced in recent kernel releases (from April 7, 2021 to April 27, 2021). **Please upgrade to the latest release (from April 28, 2021 onwards).**

C.6 Deployment fails for another reason

You can try to put the board in bootloader mode (see Appendix A) and deploy again.