Lab 1 – IoT and Embedded Systems

# Prerequisities

For this lab you will need:

* STM32L4 Discovery kit IoT node (B-L475E-IOT01A)
* A USB Cable
* [ARM Keil Studio Cloud](https://www.keil.arm.com/mbed/?utm_source=mbed_com&utm_medium=compiler_button&utm_campaign=mbed_com_promo) (Requires registration)
* MATLAB (version 2021b)

# Part 1 : Ιntroduction to Keil Studio

Arm Keil Studio is a free to use, browser-based IDE for the evaluation and development of embedded, IoT, and Machine Learning software for Cortex-M devices. To work with development boards over USB, you must use Keil Studio in a desktop browser that supports the WebUSB standard: Google Chrome or Microsoft Edge (Chromium).

After registration, you can access the IDE. To verify the connection, we will setup an example from the environments predefined code templates.

First, create a new project. Click File > New… > Mbed Project or Click on New Project on Let’s get started section:

A screenshot of a computer

Description automatically generated with medium confidence

Choose mbed2-example-blinky. The name of the project is automatically set. Change it to “Lab 1”. Make sure you have chosen the “Make this the active project” option.

Graphical user interface, text, application

Description automatically generated Graphical user interface, application

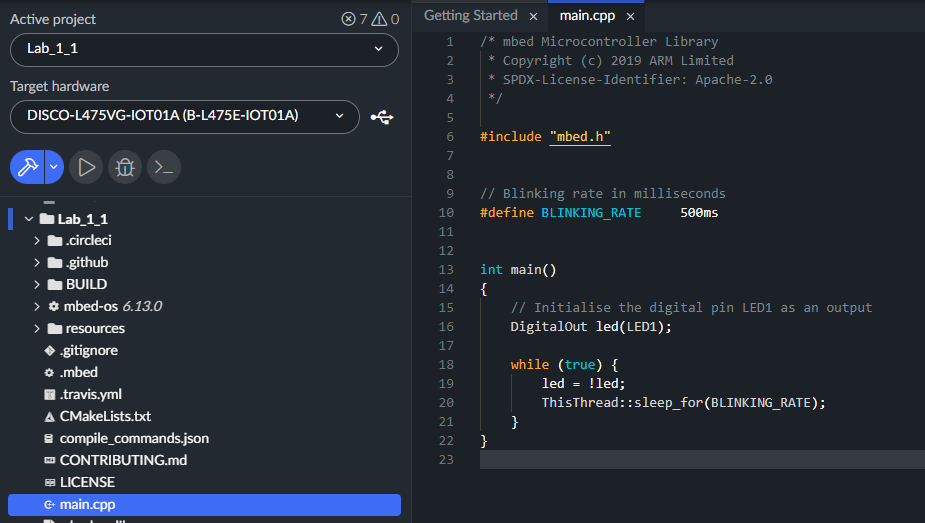
Description automatically generated

Then connect STM32L4 Discovery kit IoT node to your workstation (should be discoverable if you choose STM32 STLink). Several LEDs should light up on your board. Back on the IDE, go to Target Hardware and choose your device:

Graphical user interface, application

Description automatically generated

After that, simply run the project. After the build, the code is downloaded from your browser. You can review the code by opening main.cpp:



# Part 2 : Create a Serial connection

The objective of this exercise is to simulate the exchange of data among an edge device – your node- and a hub – your computer. The scenario involves the following:

* The hub sends some data to the node
* A led blinks to indicate data are read from the serial
* The node responds with the data that received

To establish communication, you first need to create a serial connection to both ends. First go to Keil and replace main.cpp code with the following:

#include "mbed.h"

// Maximum number of element the application buffer can contain

#define MAXIMUM\_BUFFER\_SIZE                                                  32

// Create a DigitalOutput object to toggle an LED whenever data is received.

static DigitalOut led(LED1);

// Create a BufferedSerial object with a default baud rate.

static BufferedSerial serial\_port(USBTX, USBRX);

int main(void)

{

    // Set serial port properties

    serial\_port.set\_baud(9600);

    serial\_port.set\_format(8, BufferedSerial::None, 1);

    // Application buffer to receive the data

    char buf[MAXIMUM\_BUFFER\_SIZE] = {0};

// Continuously

    ????{

// Read serial condition

         ????

            // Toggle the LED.

            ????

            // Echo the input back to the serial

            ????

        }

    }

}

Replace the question marks with the required code. Next, open Matlab environment and create a new script named “Lab1\_Serial\_Comm”. Paste the following code in it:

%% Script to read constantly the Serial synchronously

clear all

close all

%Clear communication channel

delete(instrfind)

% Init Serial with the same settings as in node’s code

COM\_PORT\_NUMBER = "????"; % run the command ‘seriallist’ to retrieve this

baud\_rate = ????;

data\_bits = ????;

myComPort = serial(COM\_PORT\_NUMBER,'BaudRate',baud\_rate,'DataBits',data\_bits);

% Init variables

datain = '';

dataout = 'Hello World';

% Open communication channel

fopen(myComPort);

% Continuously

???

% Send one or more characters to the STM32 device

????

% Pause for a while to slow down the exchange

????

% Read the available UART data as ‘uint8’ and display the result

????

disp(char(datain'));

end

% Close communication channel

fclose(myComPort);

Replace the question marks with the required code. For the COM\_PORT\_NUMBER, run the command ‘seriallist’ in MATLAB command prompt to derive the active ports in your computer.