**Tutorial - Part1: Mbed**

**I. Overview**

In this tutorial, you will use ‘mbed’ online compiler to handle several peripherals of MCU (ARM-Cortex M4). Using the given platform, you will perform some tasks about GPIO, timer and interrupt.

The objectives of this lab are

* Practice to use ‘mbed’ library in online compiler.
* Understand digital in/out peripheral in MCU.
* Understand timer and interrupt function of MCU.
* Handle GPIO, timer and interrupt using ‘mbed’ library.

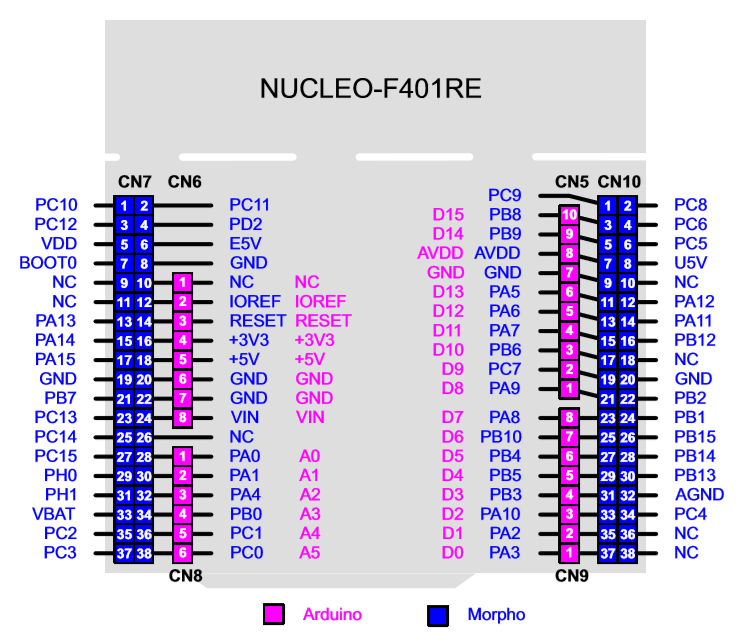
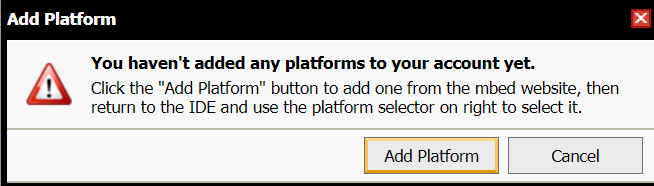


Figure 1. Pin configuration for NUCLEO-F401RE board

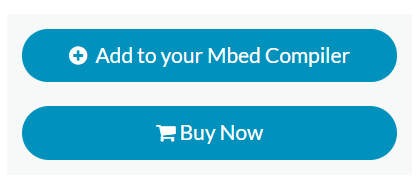
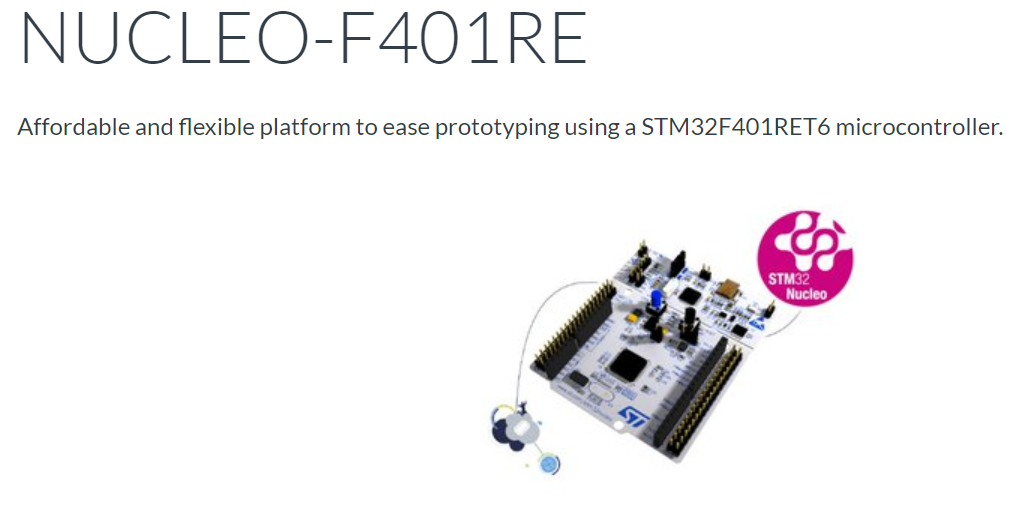
**II. Tutorial**

**A. Creating Programs**

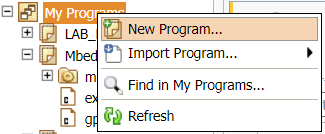
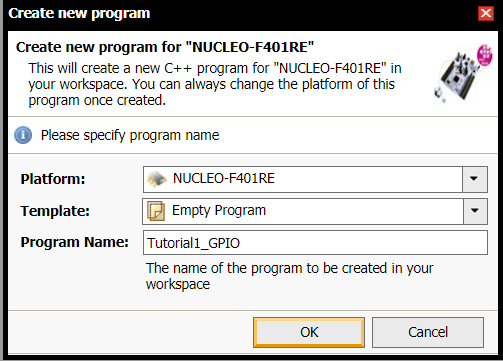
* Open <https://www.mbed.com/en/>
* Click on yellow **Compiler** button.
* Create new program as ‘Tutorial1\_mbed’. If it asks for ‘add new platform’,



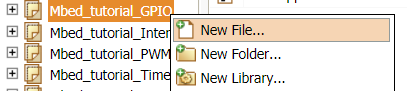
then ‘**Add Platform**’ 🡪 search for ‘**NUCLEO-F401RE**’ board 🡪 Click ‘**Add to your Mbed Compiler**’ and ‘**Open Mbed Complier**’



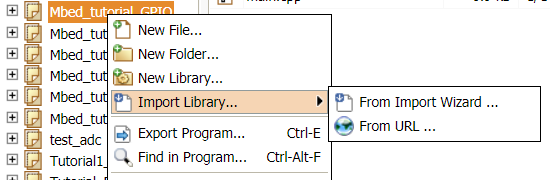
* Then, create new program as ‘**Tutorial1\_GPIO**’

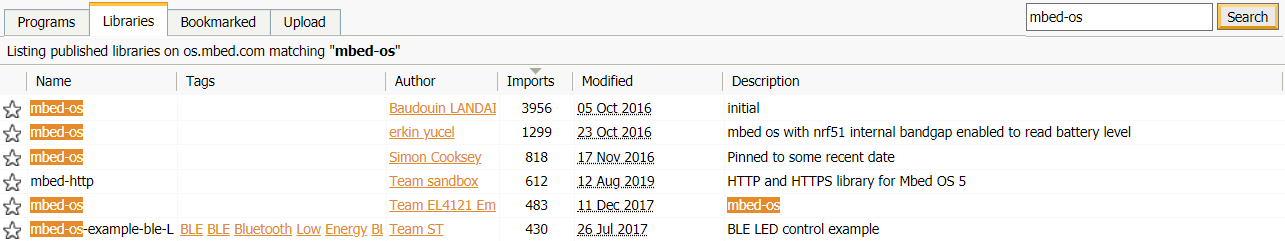
 

* Create new file ‘**main.cpp’**.

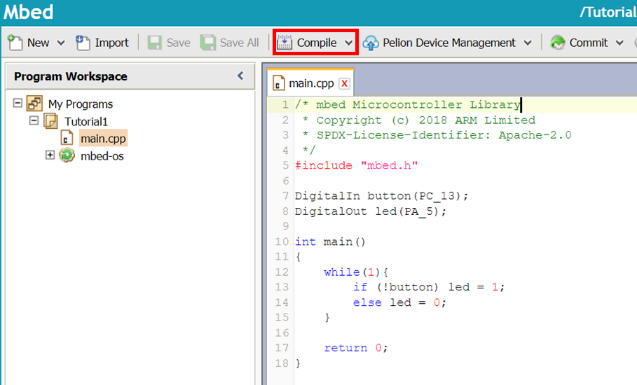


* Import ‘mbed-os’ library to program. Search ‘mbed-os’ and double-click ‘**mbed-os**’ by Baudouin LANDAI which is the first one.





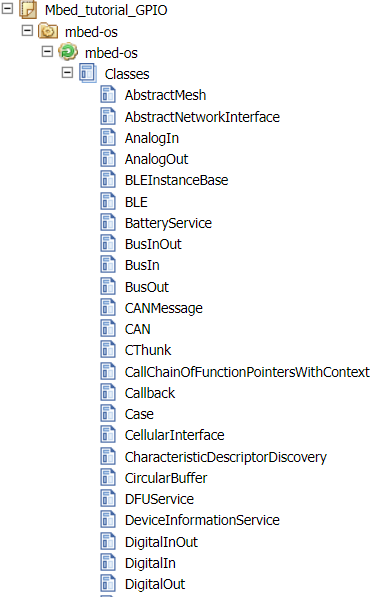
* Write the source code on ‘main.cpp’. (Specific codes will be given in next parts B ~ D.)
* To compile the program, click on ‘**Compile’** button.



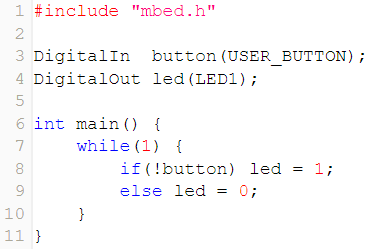
* Then, the binary file of the project will be created and downloaded on your computer.
* Connect the MCU board to your PC via USB cable and check if a new memory drive of “NODE\_F401RE (E:)” is created.
* To load the program onto the MCU, copy the downloaded binary file to the drive “NODE\_F401RE (E:)”. If the program is loaded successfully then LED(LD1) will be green light.

**B. GPIO (General purpose Input/Output)**

* If you want to use digital input/output pins, you can use ‘DigitalIn, DigitalOut’ classes. You can search many useful class commands of ‘mbed’ library as shown below.



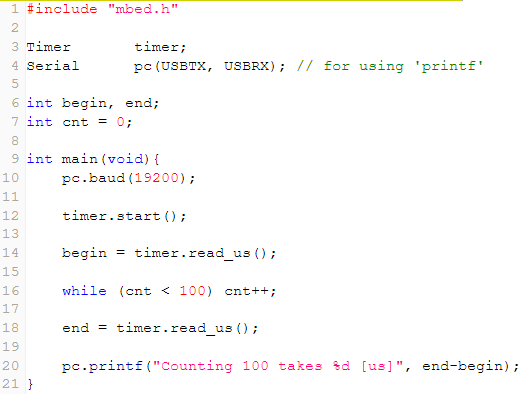
* Create new program as ‘**Tutorial1\_GPIO**’.
* Write the following source code on ‘main.cpp’



* Click on **Compile** button. Then, the binary files will be created and downloaded.
* Then, the binary file “Tutorial1\_GPIO.NUCLEO\_F401RE.bin” of the project will be created and downloaded on your computer.
* Connect the MCU board to your PC via USB cable.
* Push the reset button(black) and verify the performance. The LED should be turned on when the button is pressed.
* To load the program onto the MCU, copy the downloaded binary file to the drive “NODE\_F401RE (E:)”. If the program is loaded successfully then LED(LD1) will be green light.

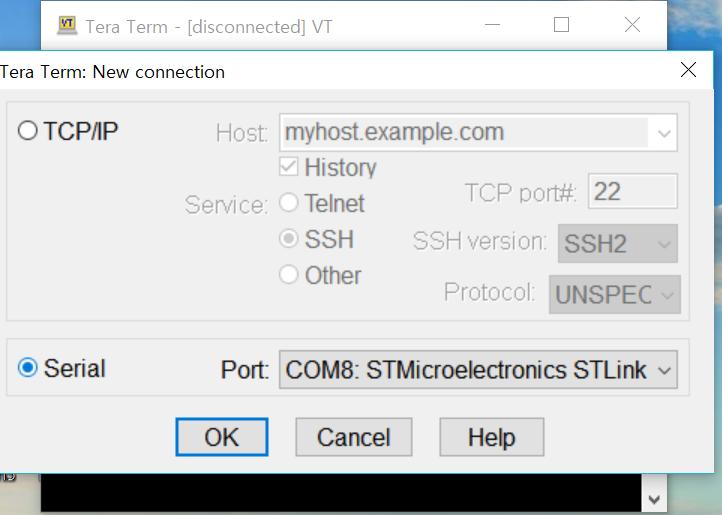
**C. Timer**

* If you want to measure time taken in certain process, you can use ‘Timer’ class.
* Create new program as ‘**Tutorial1\_Timer**’.
* Write the following source code on ‘main.cpp’. You will measure time to count 100 from 0.

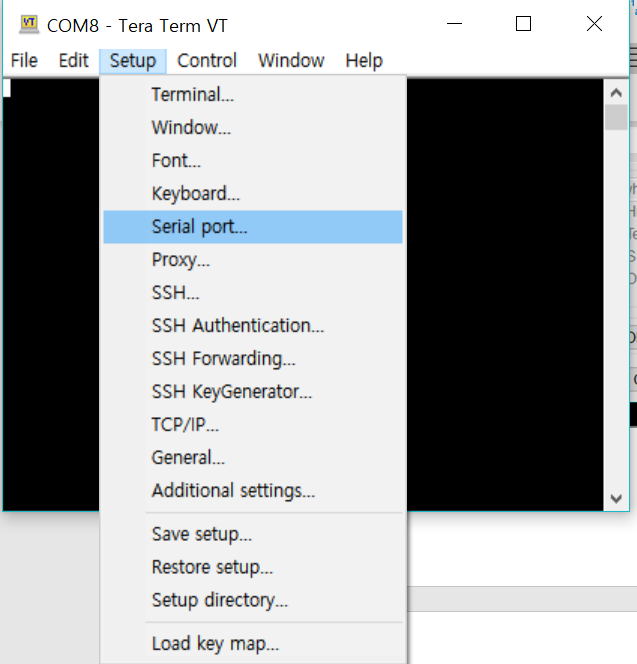
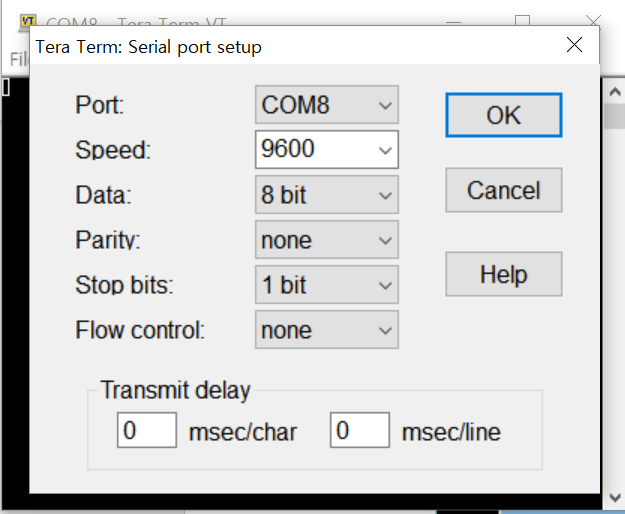


* Click on **Compile** button. Then, the binary files will be created and downloaded.
* Then, the binary file will be created and downloaded on your computer.
* To load the program onto the MCU, copy the downloaded binary file to the drive “NODE\_F401RE (E:)”. If the program is loaded successfully then LED(LD1) will be green light.
* To verify the performance, firstly open ‘Tera Term’ and follow the sequences.

Choose ‘**Serial**’ tab 🡪 Select ‘**COMx: STMicroelectronics STLink**’ port



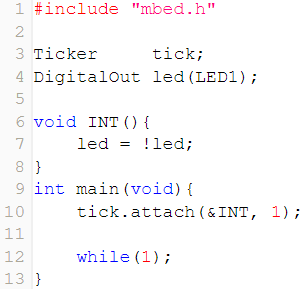
Open Serial port(시리얼 포트) in Setup(설정) tab, check if the baud rate is set as **9600** [bps].

* Push the reset button(black), and verify the time taken in counting 100. You can measure time taken in any other processes like toggling LED, multiplication or division, etc. If the process takes long time, you can also measure time in [ms] unit using ‘timer.read\_ms()’ command.

**D. Ticker (Timer Interrupt)**

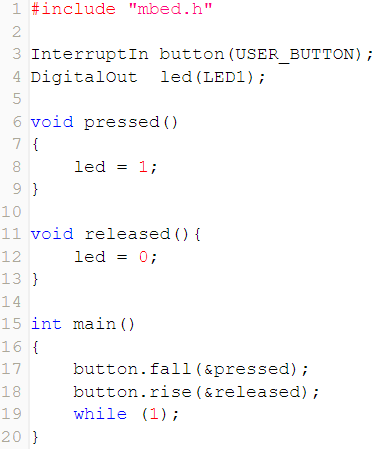
* If you want to use call interrupt in every second, you can use ‘Ticker’ class.
* Create new program as ‘**Tutorial1\_Ticker**’.
* Write the following code on ‘mbed’ complier. You will make LED blink every second, even though there is no specific code in infinite loop syntax. This function is called ‘timer interrupt’ in MCU.



* Click on **Compile** button. Then, the binary files will be created and downloaded.
* Then, the binary file will be created and downloaded on your computer.
* To load the program onto the MCU, copy the downloaded binary file to the drive “NODE\_F401RE (E:)”. If the program is loaded successfully then LED(LD1) will be green light.
* Verify the performance. LED(LD2) should blink every second.

**E. Interrupt**

* If you want to use external interrupt, you can use ‘InterruptIn’ class.
* Create new program as ‘**Tutorial1\_Interrupt**’.
* Write the following code on ‘mbed’ complier. The performance will be same with that of B.GPIO part, but the main function has empty infinite loop unlike B.GPIO. This represents the advantage of using interrupt function.



* Click on **Compile** button. Then, the binary files will be created and downloaded.
* Then, the binary file will be created and downloaded on your computer.
* To load the program onto the MCU, copy the downloaded binary file to the drive “NODE\_F401RE (E:)”. If the program is loaded successfully then LED(LD1) will be green light.
* Verify the performance. LED(LD2) should be turned on when user button is pushed.