NIT MEGHALAYA

EMBEDDED SYSTEM DESIGN AND APPLICATION (CS 313)

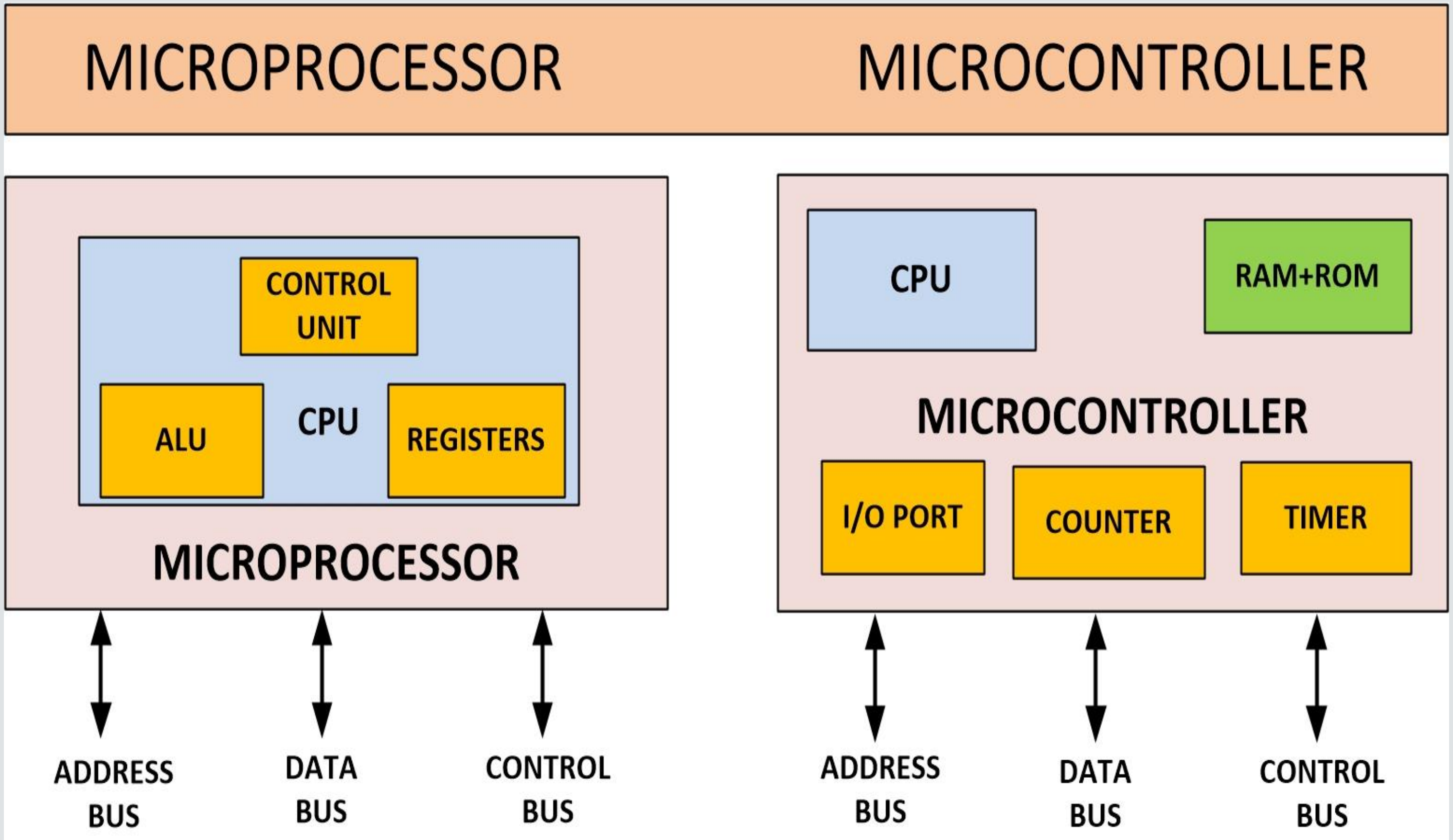
NIT MEGHALAYA

LAB 1 :- INTRODUCTION

Embedded System

- An **embedded system** is a system that has software embedded into computer-hardware, which makes a system dedicated for an application or specific part of an application.
- Embedded system can be **microprocessor** or **microcontroller** based.

Microprocessor and Microcontroller



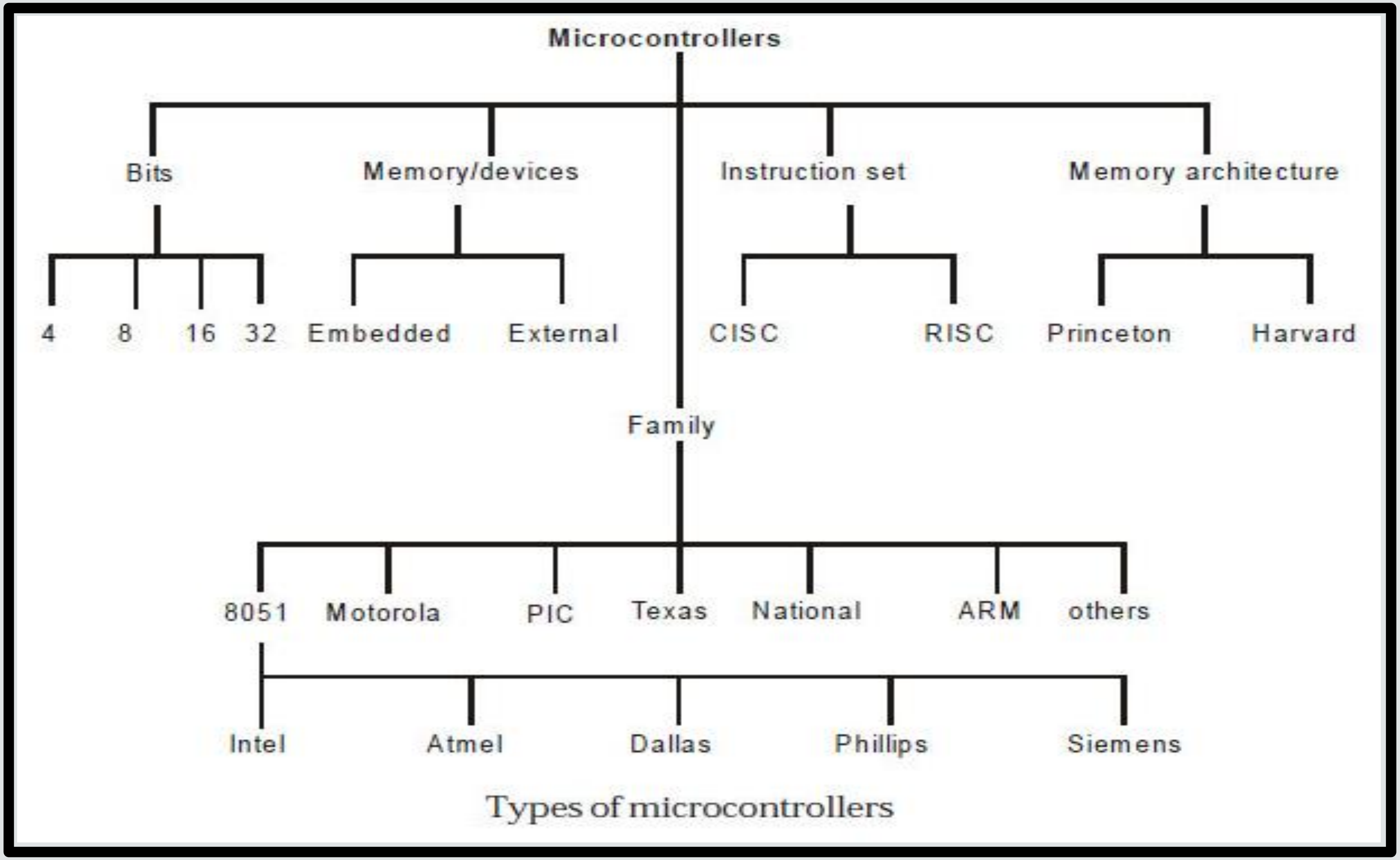
MICROPROCESSOR

- Assimilates function of CPU.
- Used in design of general purpose systems.
- Overall cost and power consumption of a system built using a microprocessor is high.
- Not used in real time systems.

MICROCONTROLLER

- Can be considered as a small computer.
- Used in automatically controlled devices.
- Overall cost and power consumption of a system built using a microcontroller is less.
- Used to handle real time tasks.

Families of Microcontrollers



Types of microcontrollers

The First Microcontroller

- During 1970 and 1971 **Gary Boone** of Texas Instruments invented first microcontroller **TMS1802NC**.

- It had five thousand transistors providing 3000 bits of program memory and 128 bits of access memory!! So, it was possible to program it to perform a range of functions.

About the KIT (STM32F401)

- Developed by ST Microelectronics.
- CPU – ARM Cortex© M4 (ARM-Advanced RISC Machines)
- 512 KB Flash Memory (Programmable) and 96-KB SRAM
- USB 2.0 type A to mini B
- mbed-enabled (mbed.org): **Mbed** is a platform and operating system for internet-connected devices based on 32-bit ARM Cortex-M microcontrollers. Such devices are also known as Internet of Things devices. The project is collaboratively developed by Arm and its technology partners.

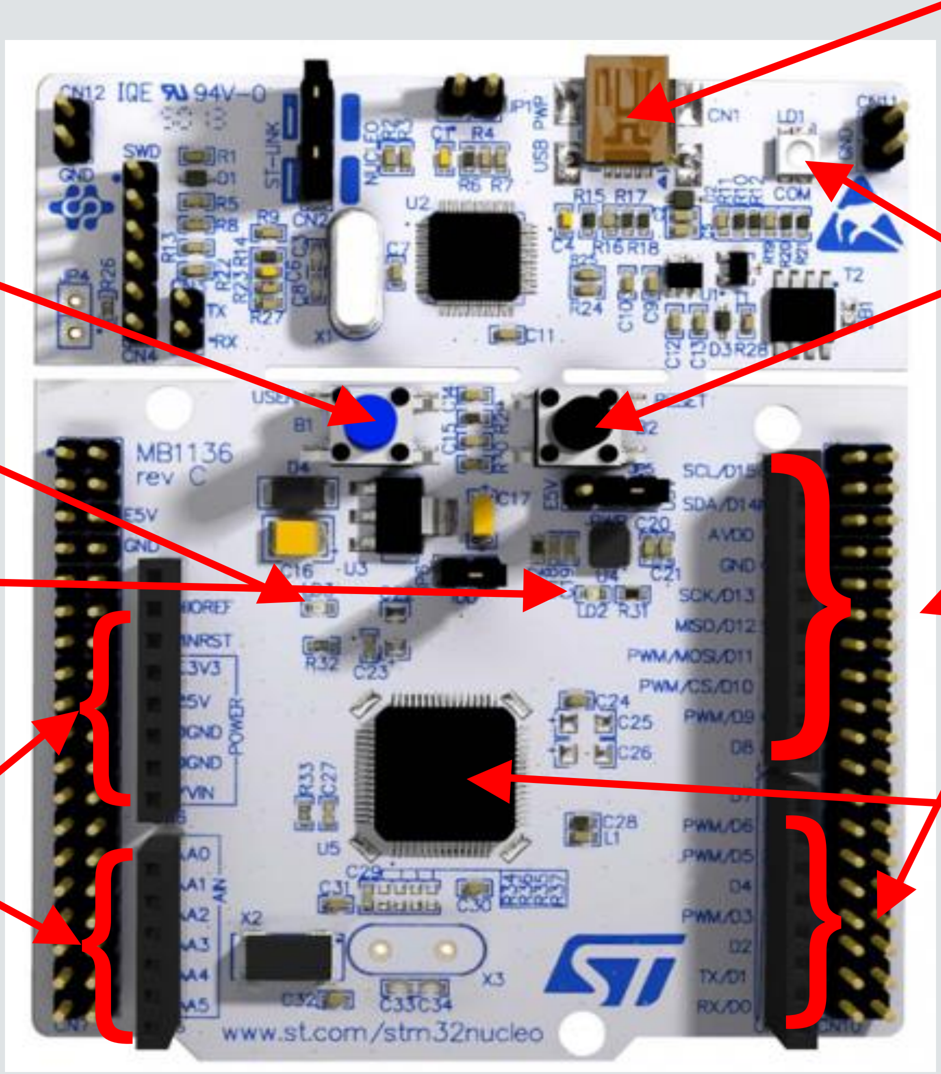
About the KIT (STM32F401)

- Support of wide choice of Integrated Development Environments (IDEs) including IAR™, ARM® Keil®, GCC-based IDEs
- http://www.st.com/en/evaluation-tools/nucleo-f401re.html

What is Arduino?

- Open-source platform used for building electronics projects.
- Consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software.
- Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable.
- The Uno is one of the more popular boards in the Arduino family and a great choice for beginners.

STM32F401 Nucleo kit



ST Link USB Type A to mini B

Reset Button

User Button

Red LED (power)

Red/Green LED (com)

Green LED (test)

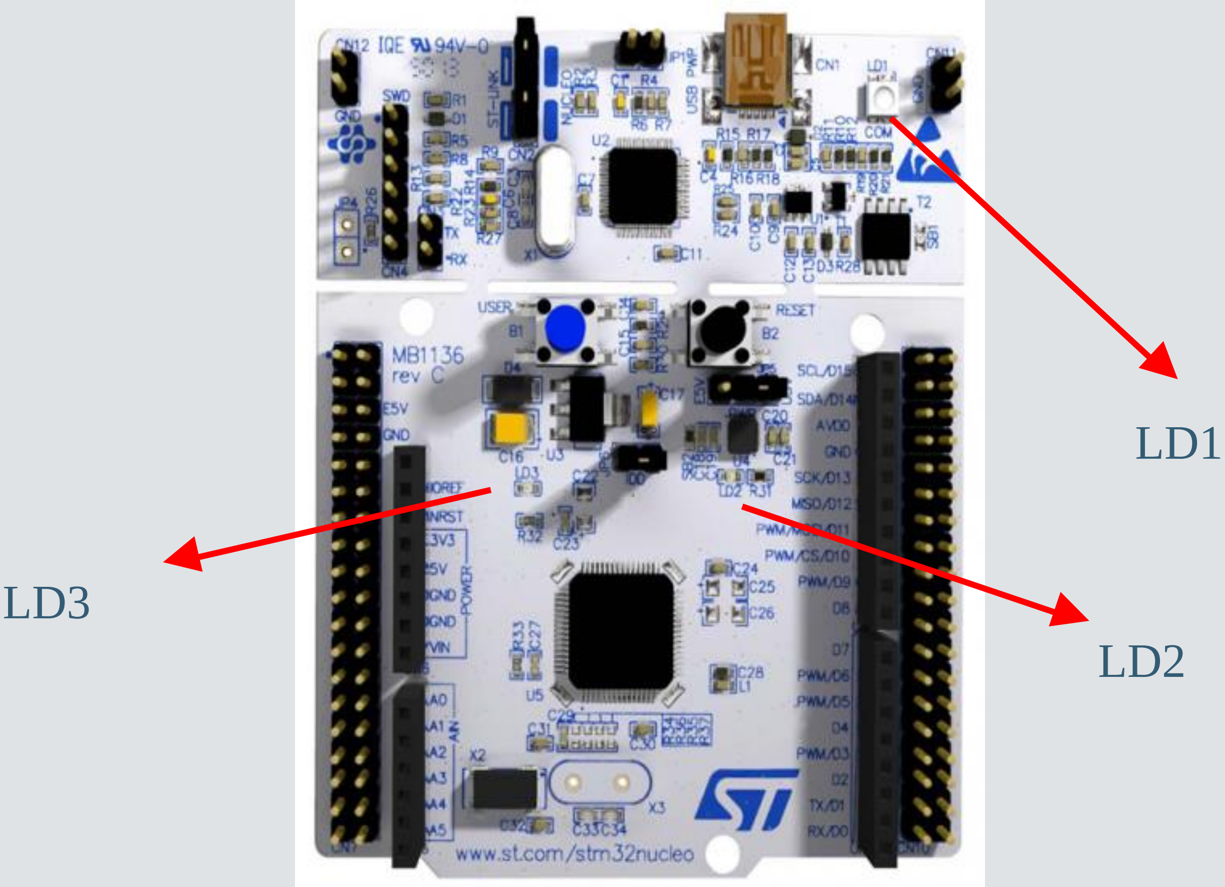
Arduino Connector and Digital output

Arduino Connector and Analog input

STM32 Microcontroller

LEDs Description

- There are three LEDs on the STM32 NUCLEO kit.
  - LD1
  - LD2
  - LD3



LD 1

- The tricolor LED (green, orange, red) LD1 (COM) provides information about ST-LINK communicate on status.
- LD1 default color is red. LD1 turns to green to indicate that communication is in progress between the PC and the ST-LINK/V2-1, with the following setup:
  - Slow blinking Red/Off: at power-on before USB initialization
  - Fast blinking Red/Off: after the first correct communication between the PC and ST-LINK/V2-1 (enumeration)
  - Red LED On: when the initialization between the PC and ST-LINK/V2-1 is complete
  - Green LED On: after a successful target communication initialization • Blinking Red/Green: during communication with target
  - Green On: communication finished and successful
  - Orange On: Communication failure

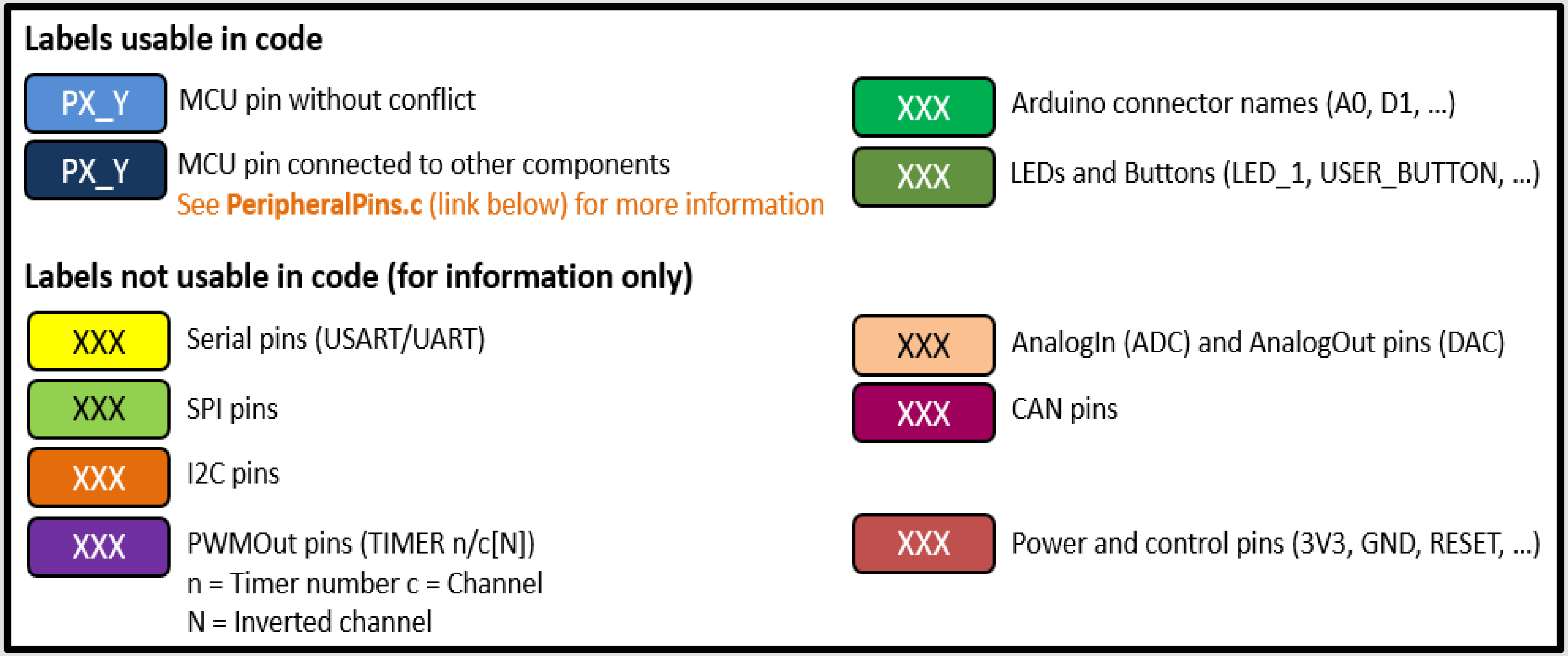
LD2

- **User LD2**: the green LED is a user LED connected to Arduino signal D13 corresponding to STM32 I/O PA5 (pin 21) or PB13 (pin 34) depending on the STM32 target :
  - the I/O is HIGH value, the LED is on
  - the I/O is LOW, the LED is off

LD3

- **LD3 PWR**: The red LED indicates that the STM32 part is powered and +5V power is available.

PINS LABEL



Labels usable in code

PX_Y MCU pin without conflict

PX_Y MCU pin connected to other components
See **PeripheralPins.c** (link below) for more information

XXX Arduino connector names (A0, D1, ...)

XXX LEDs and Buttons (LED_1, USER_BUTTON, ...)

Labels not usable in code (for information only)

XXX Serial pins (USART/UART)

XXX SPI pins

XXX I2C pins

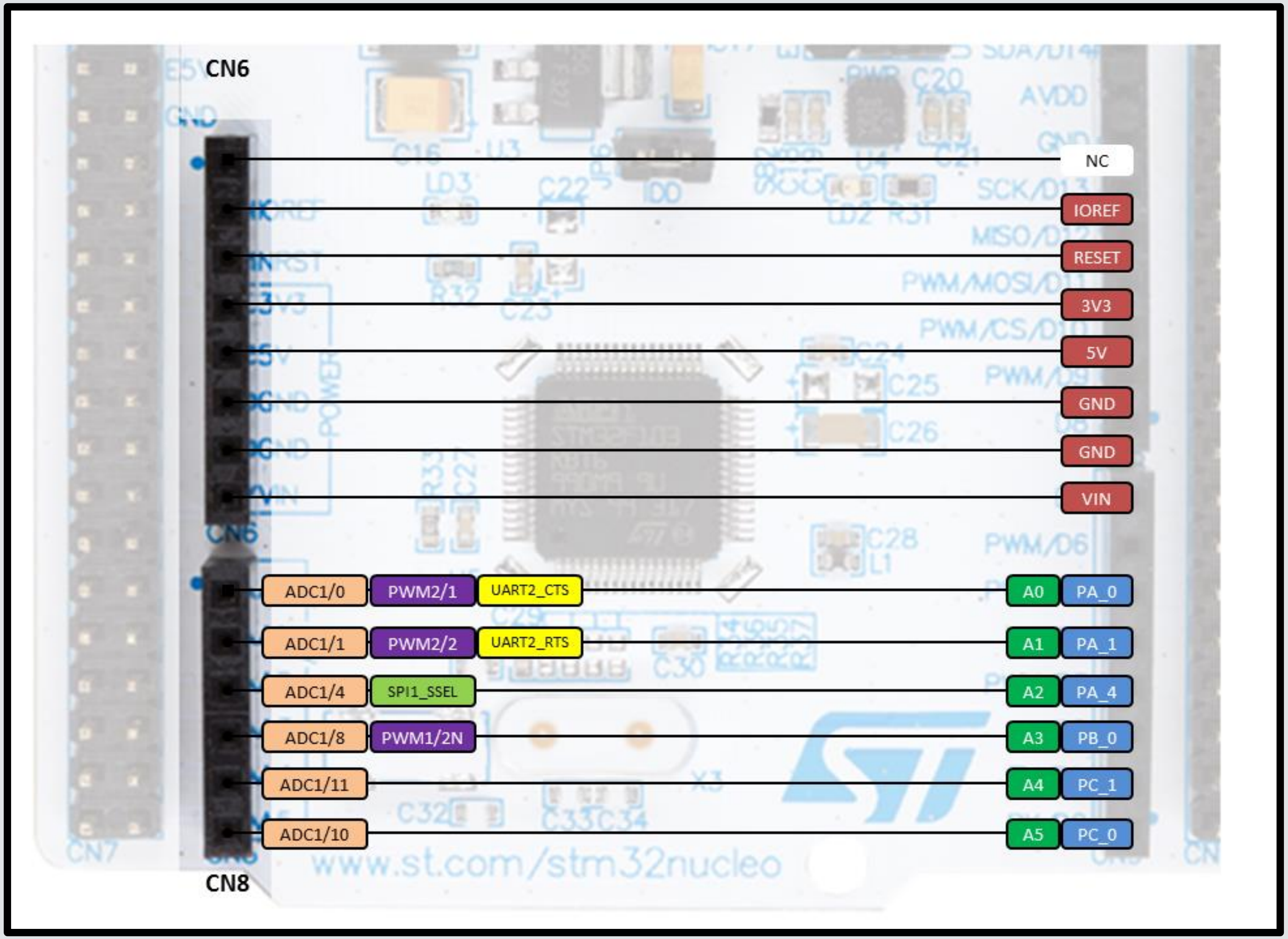
XXX PWMOut pins (TIMER n/c[N])
n = Timer number c = Channel
N = Inverted channel

XXX AnalogIn (ADC) and AnalogOut pins (DAC)

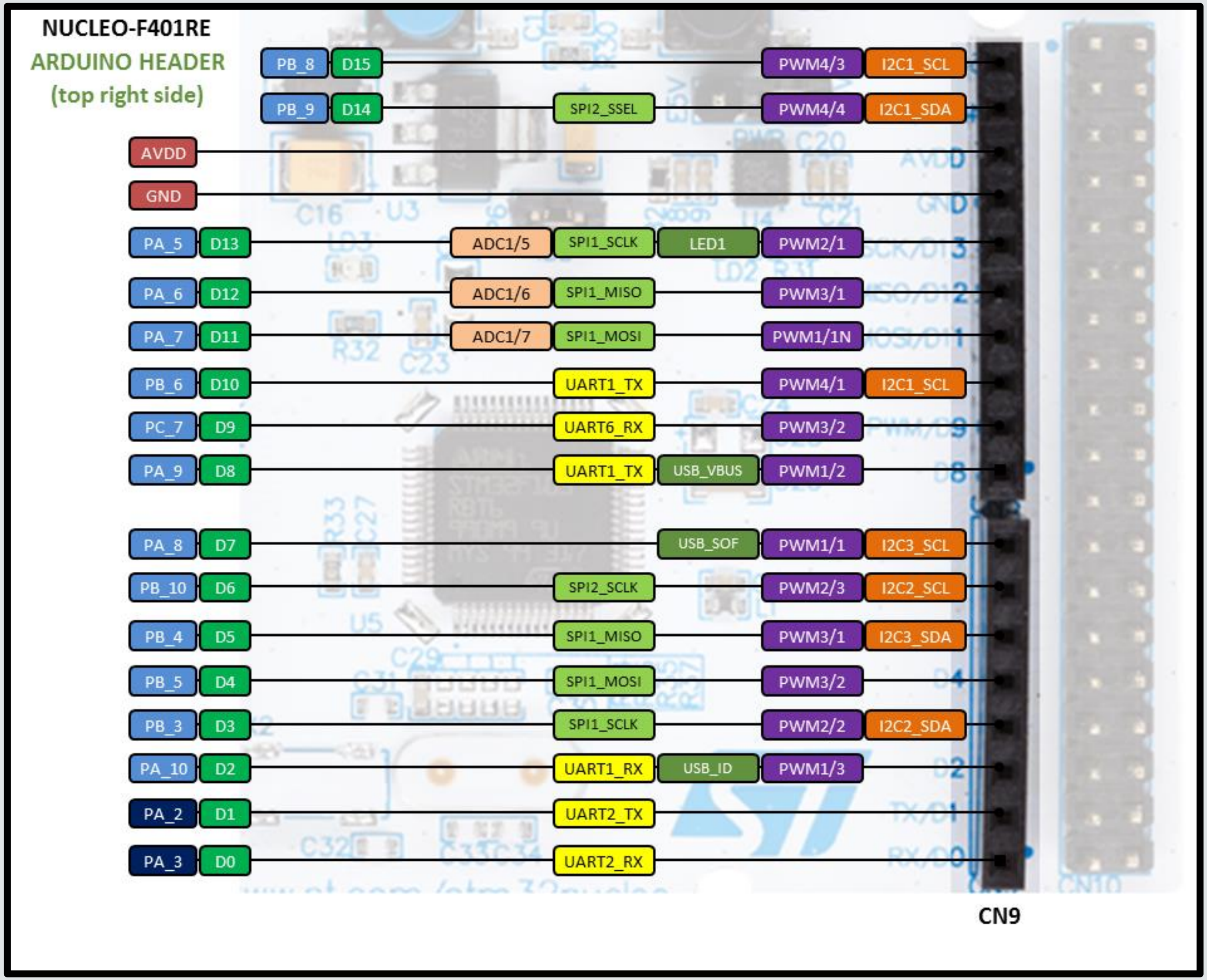
XXX CAN pins

XXX Power and control pins (3V3, GND, RESET, ...)

Arduino Header CN6 & CN8



Arduino Header CN9



NUCLEO-F401RE
ARDUINO HEADER
(top right side)

PB_8 D15 PWM4/3 I2C1_SCL

PB_9 D14 SPI2_SSEL PWM4/4 I2C1_SDA

AVDD

GND

PA_5 D13 ADC1/5 SPI1_SCLK LED1 PWM2/1

PA_6 D12 ADC1/6 SPI1_MISO PWM3/1

PA_7 D11 ADC1/7 SPI1_MOSI PWM1/1N

PB_6 D10 UART1_TX PWM4/1 I2C1_SCL

PC_7 D9 UART6_RX PWM3/2

PA_9 D8 UART1_TX USB_VBUS PWM1/2

PA_8 D7 USB_SOF PWM1/1 I2C3_SCL

PB_10 D6 SPI2_SCLK PWM2/3 I2C2_SCL

PB_4 D5 SPI1_MISO PWM3/1 I2C3_SDA

PB_5 D4 SPI1_MOSI PWM3/2

PB_3 D3 SPI1_SCLK PWM2/2 I2C2_SDA

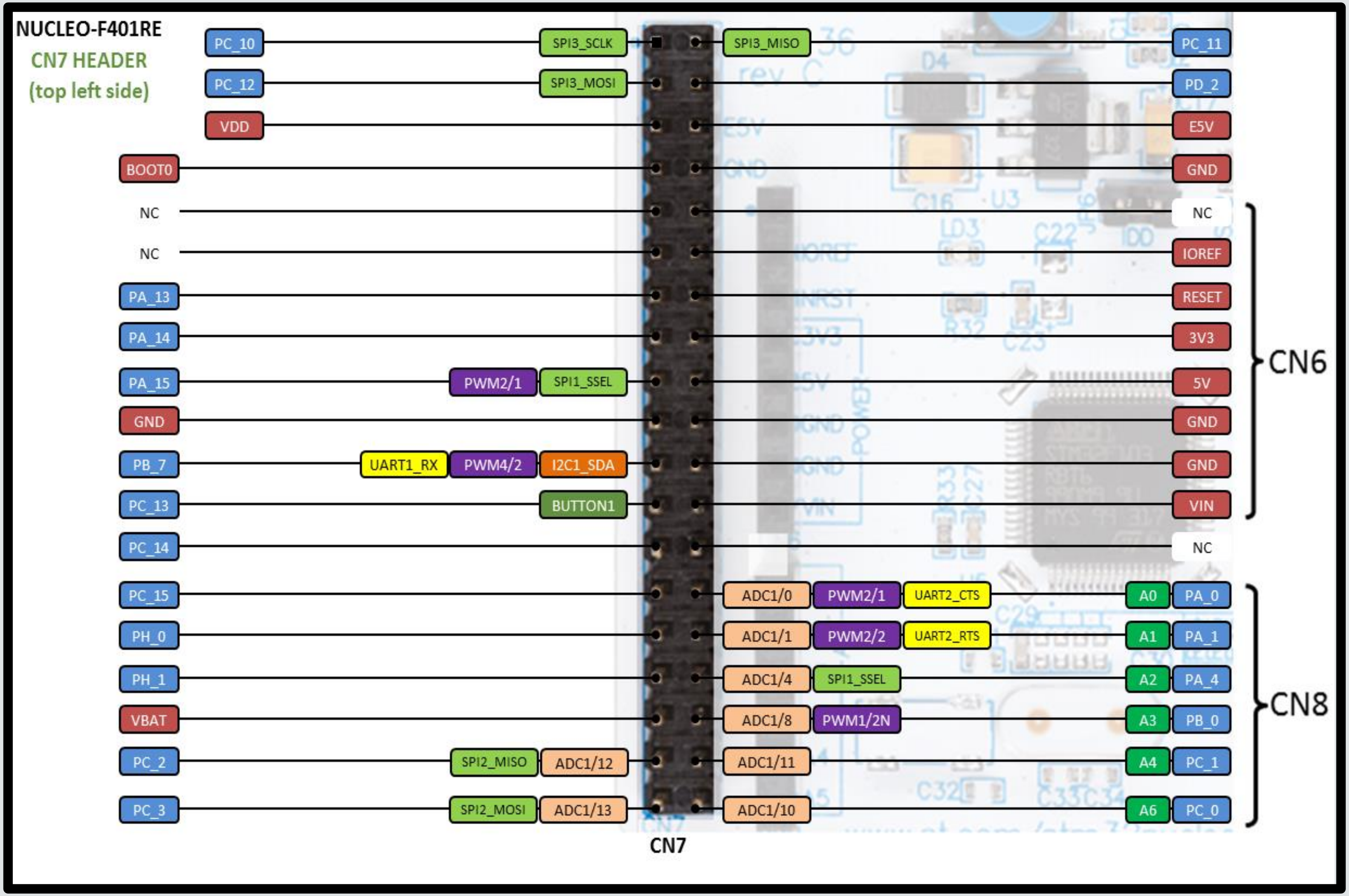
PA_10 D2 UART1_RX USB_ID PWM1/3

PA_2 D1 UART2_TX

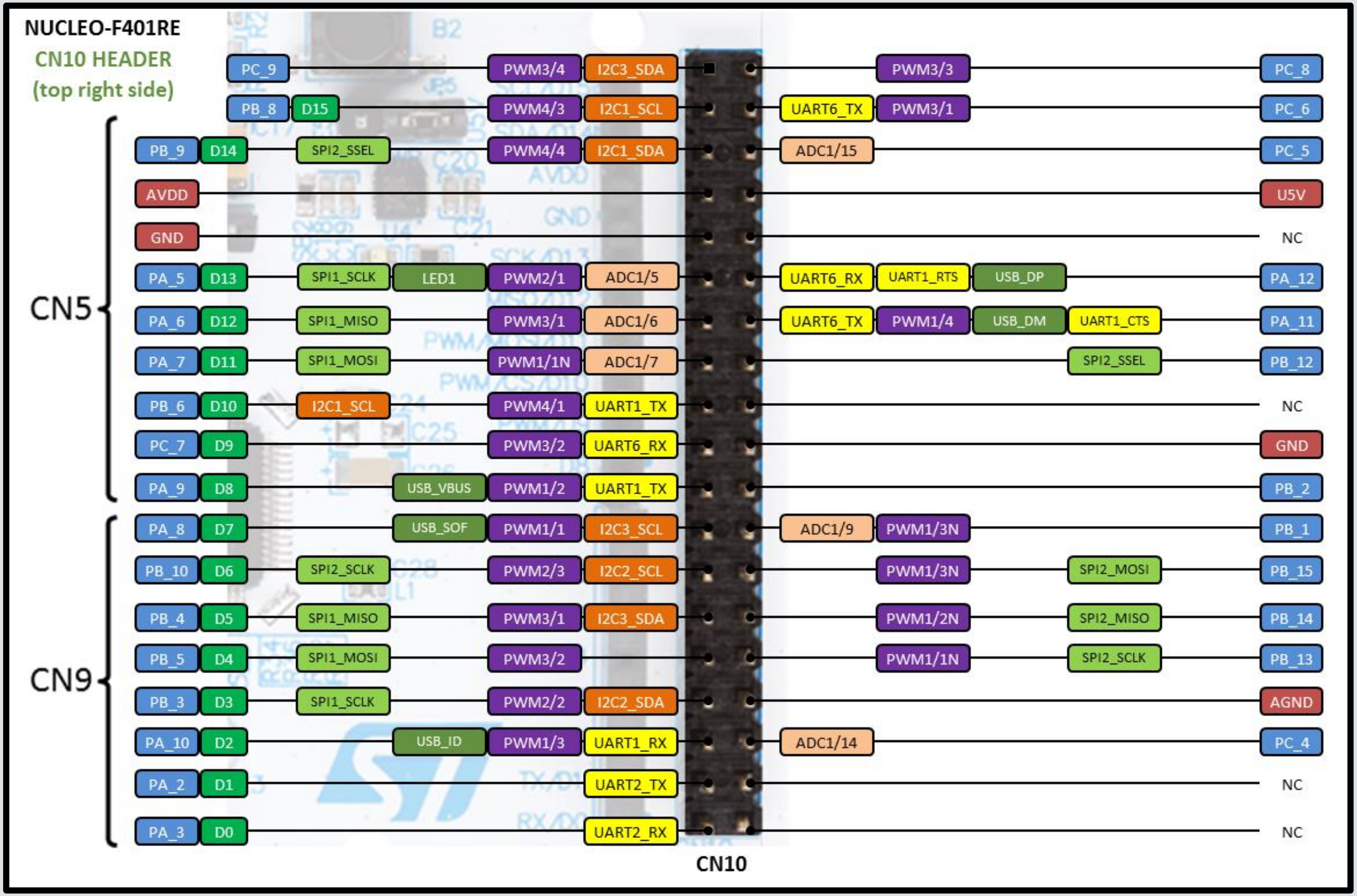
PA_3 D0 UART2_RX

CN9

CN7 Header


NUCLEO-F401RE
CN7 HEADER
(top left side)
PC_10 SPI3_SCLK SPI3_MISO PC_11
PC_12 SPI3_MOSI PD_2
VDD E5V
BOOT0 GND
NC NC
NC IOREF
PA_13 RESET
PA_14 3V3
PA_15 PWM2/1 SPI1_SSEL 5V
GND GND
PB_7 UART1_RX PWM4/2 I2C1_SDA GND
PC_13 BUTTON1 VIN
PC_14 NC
PC_15 ADC1/0 PWM2/1 UART2_CTS A0 PA_0
PH_0 ADC1/1 PWM2/2 UART2_RTS A1 PA_1
PH_1 ADC1/4 SPI1_SSEL A2 PA_4
VBAT ADC1/8 PWM1/2N A3 PB_0
PC_2 SPI2_MISO ADC1/12 ADC1/11 A4 PC_1
PC_3 SPI2_MOSI ADC1/13 ADC1/10 A6 PC_0
CN6
CN8
CN7

CN10 Header



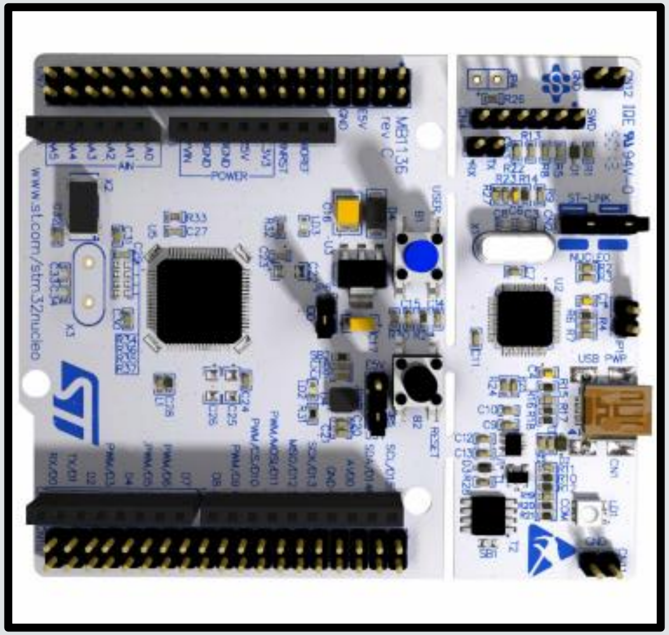
NUCLEO-F401RE

CN10 HEADER (top right side)

	PC_9				PWM3/4	I2C3_SDA		PWM3/3			PC_8
	PB_8	D15			PWM4/3	I2C1_SCL	UART6_TX	PWM3/1			PC_6
CN5	PB_9	D14	SPI2_SSEL		PWM4/4	I2C1_SDA	ADC1/15				PC_5
	AVDD										U5V
	GND										NC
	PA_5	D13	SPI1_SCLK	LED1	PWM2/1	ADC1/5	UART6_RX	UART1_RTS	USB_DP		PA_12
	PA_6	D12	SPI1_MISO		PWM3/1	ADC1/6	UART6_TX	PWM1/4	USB_DM	UART1_CTS	PA_11
	PA_7	D11	SPI1_MOSI		PWM1/1N	ADC1/7				SPI2_SSEL	PB_12
	PB_6	D10	I2C1_SCL		PWM4/1	UART1_TX					NC
	PC_7	D9			PWM3/2	UART6_RX					GND
	PA_9	D8		USB_VBUS	PWM1/2	UART1_TX					PB_2
CN9	PA_8	D7		USB_SOF	PWM1/1	I2C3_SCL	ADC1/9	PWM1/3N			PB_1
	PB_10	D6	SPI2_SCLK		PWM2/3	I2C2_SCL		PWM1/3N		SPI2_MOSI	PB_15
	PB_4	D5	SPI1_MISO		PWM3/1	I2C3_SDA		PWM1/2N		SPI2_MISO	PB_14
	PB_5	D4	SPI1_MOSI		PWM3/2			PWM1/1N		SPI2_SCLK	PB_13
	PB_3	D3	SPI1_SCLK		PWM2/2	I2C2_SDA					AGND
	PA_10	D2		USB_ID	PWM1/3	UART1_RX	ADC1/14				PC_4
	PA_2	D1				UART2_TX					NC
	PA_3	D0				UART2_RX					NC

Requirements

- STM32F401 Evaluation/Development board
- USB mini to USB Type B Connector
- Development environment (IDEs or mbed account).

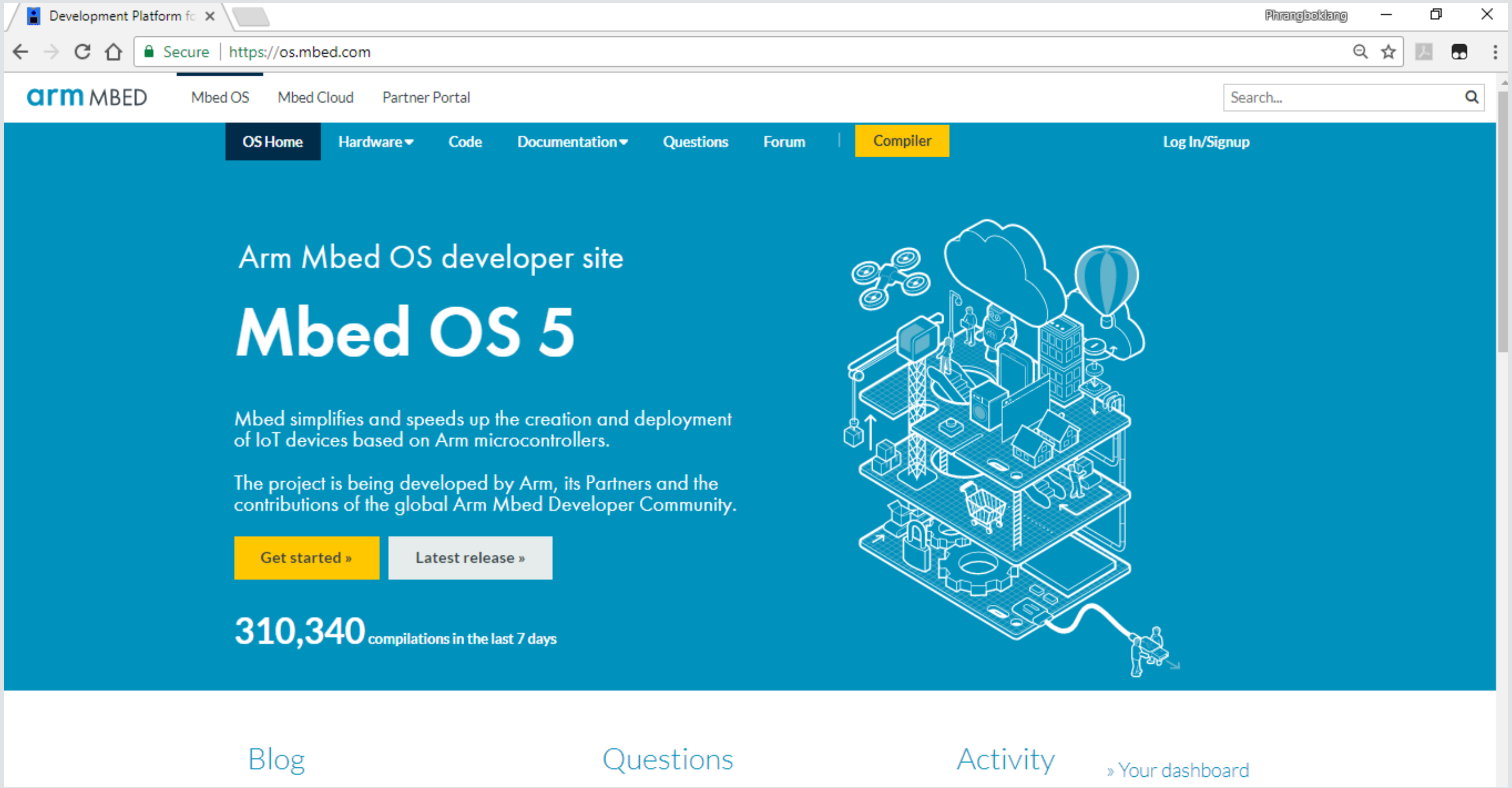


Start Development

- During this course we will use the online compiler from http://developer.mbed.org to compile our projects.
- The compiler supports programs written in C or Python.

STEPS to Start Development

- STEP 1 – Go to http://developer.mbed.org



Development Platform fc ×

Phrangboklang

Secure | https://os.mbed.com

Arm Mbed OS developer site

Mbed OS 5

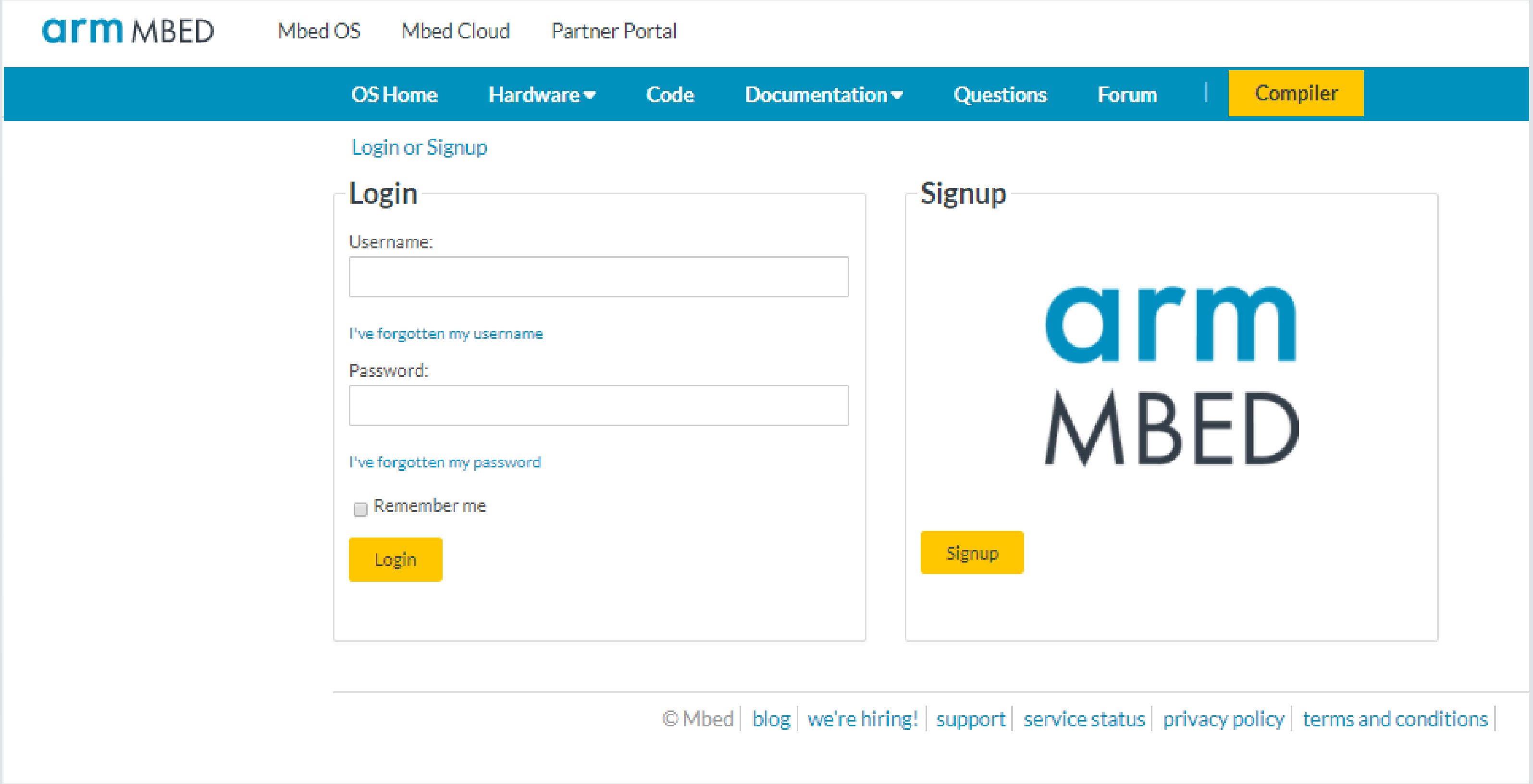
Mbed simplifies and speeds up the creation and deployment of IoT devices based on Arm microcontrollers.

The project is being developed by Arm, its Partners and the contributions of the global Arm Mbed Developer Community.

310,340 compilations in the last 7 days

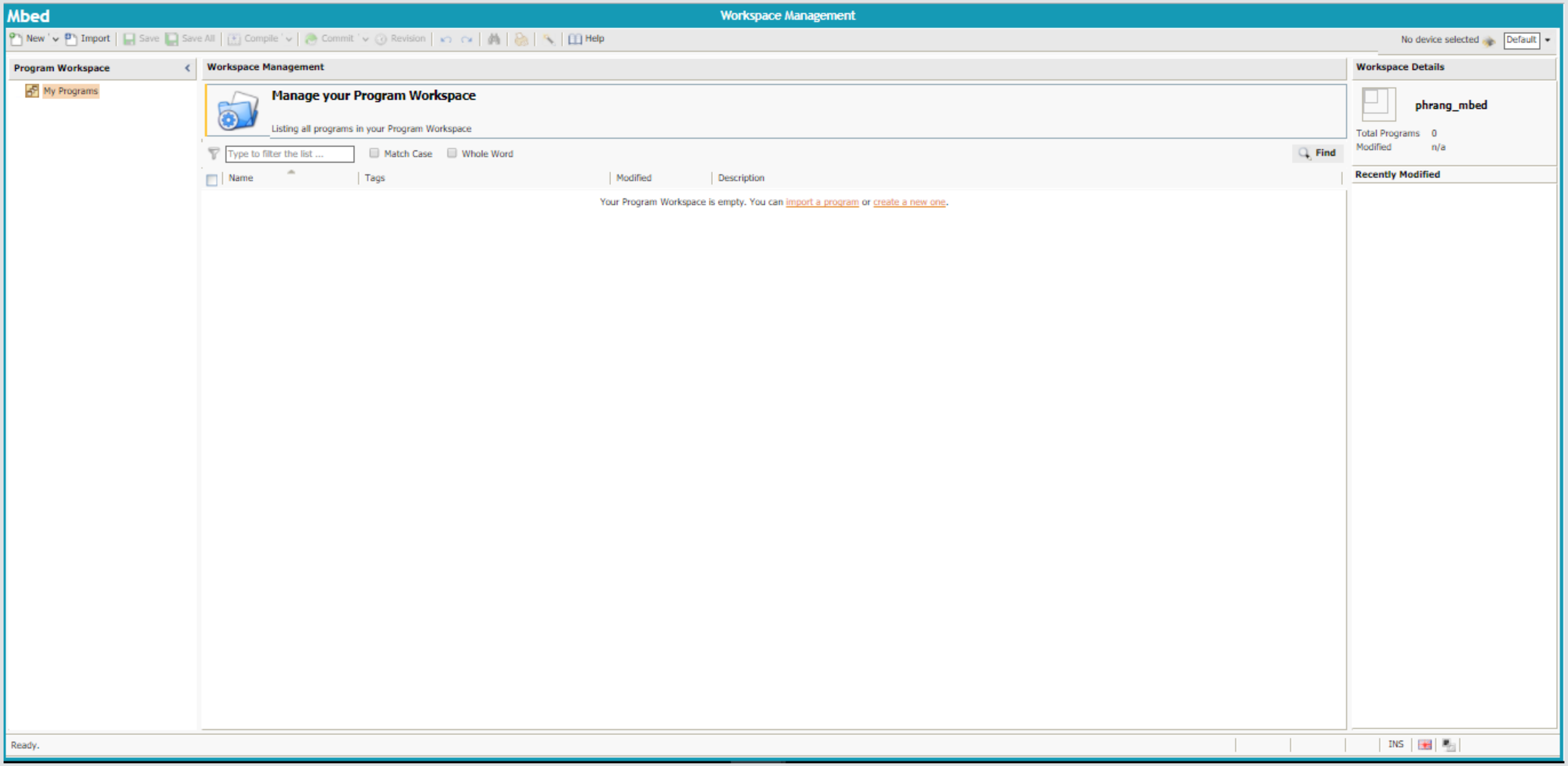
STEPS to Start Development

- STEP 2 – Create an account



STEPS to Start Development

- STEP 3 – Go to Compiler



Manage your Program Workspace

Listing all programs in your Program Workspace

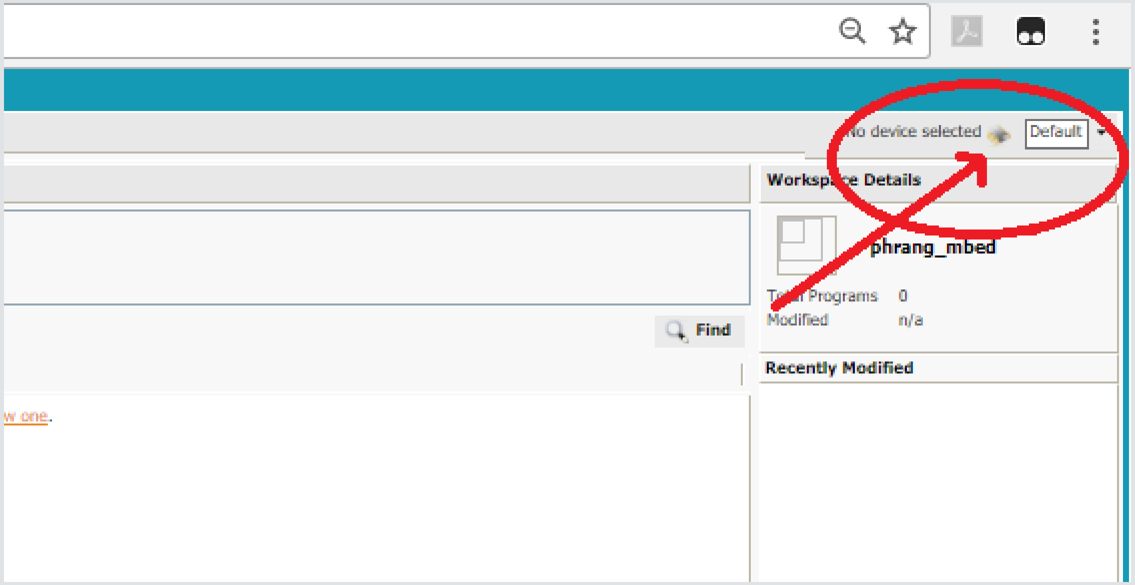
Your Program Workspace is empty. You can import a program or create a new one.

phrang_mbed

Total Programs	0
Modified	n/a

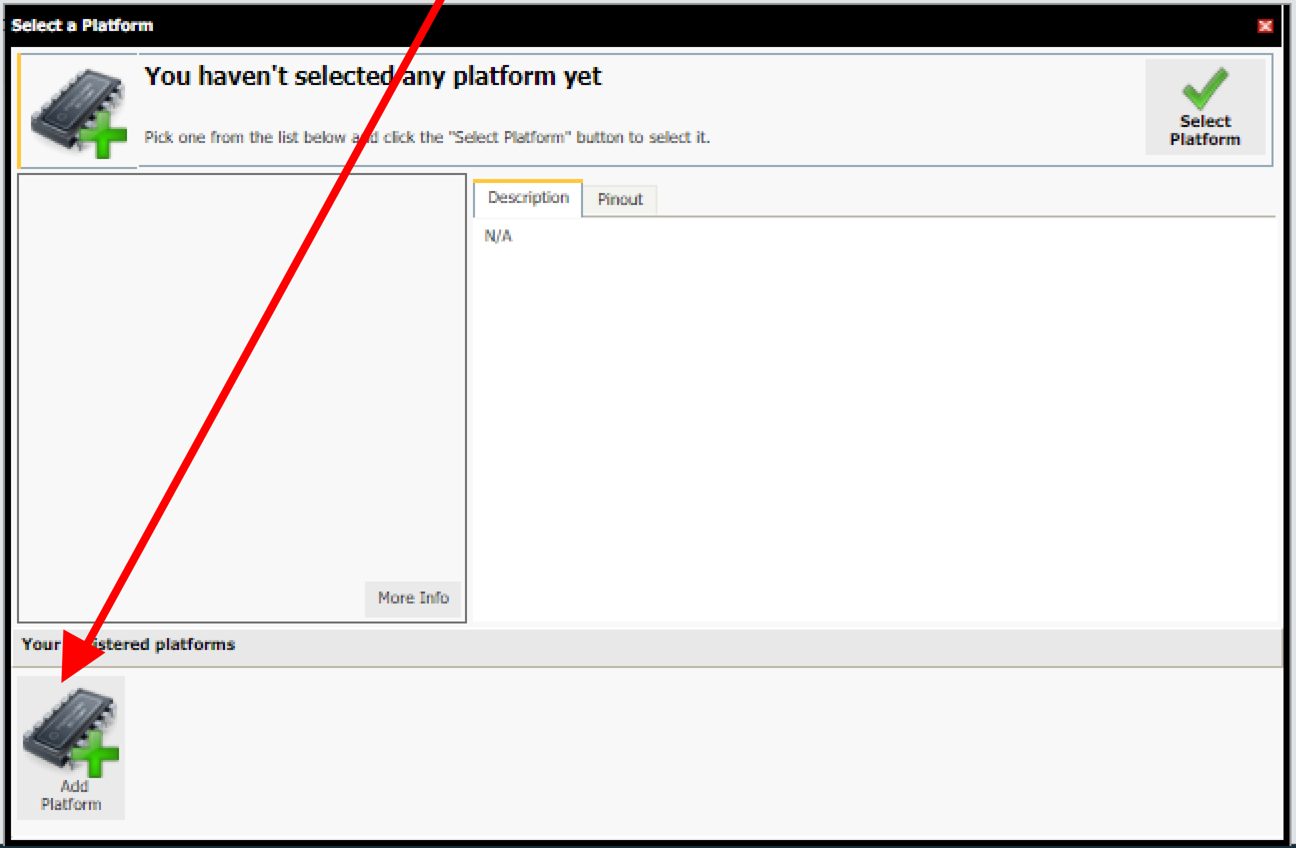
STEPS to Start Development

- STEP 4 – Selecting Device
  - Click on the device selection on the top RHS of the Complier Screen



STEPS to Start Development

- STEP 5 – Selecting Platform
  - Click on Add Platform



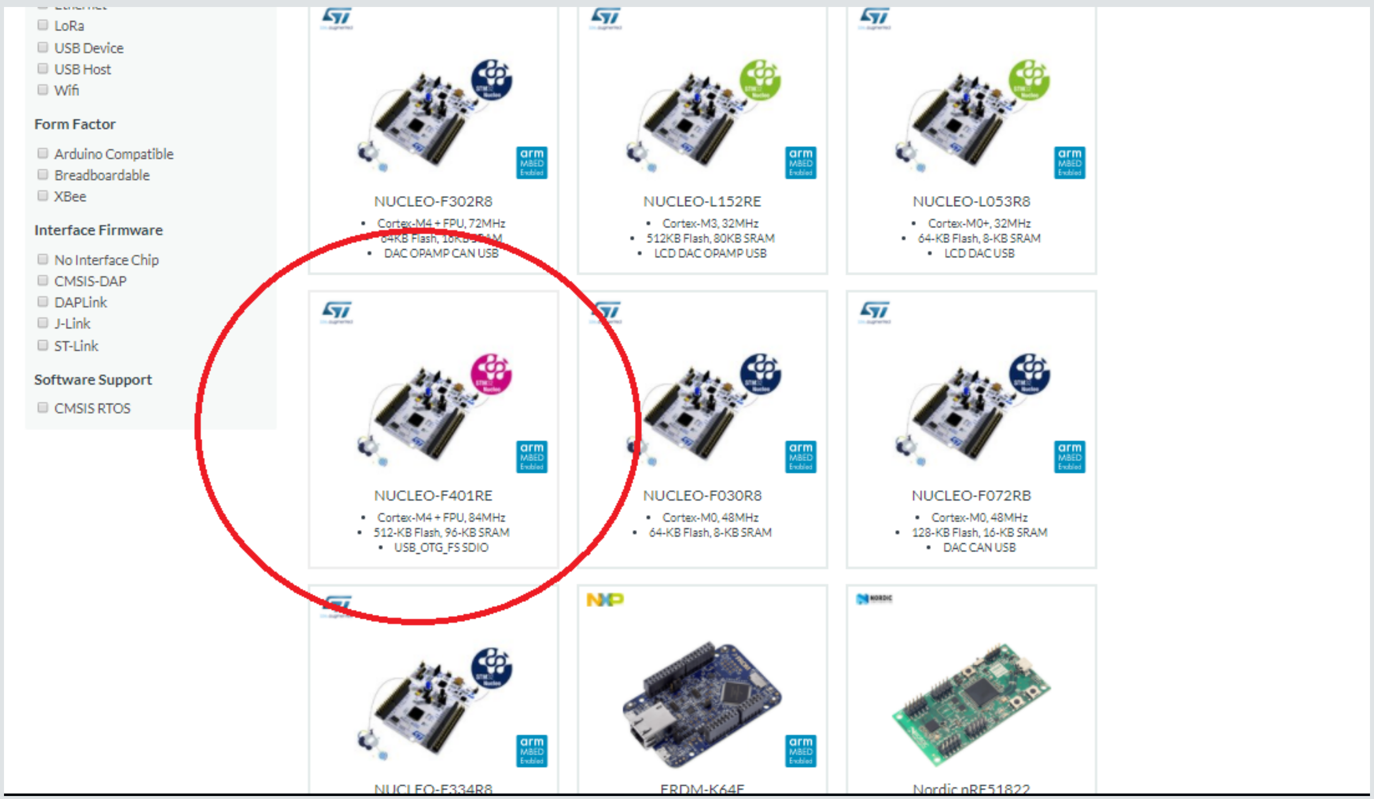
Select a Platform

You haven't selected any platform yet

Pick one from the list below and click the "Select Platform" button to select it.

STEPS to Start Development

- STEP 6 – Selecting Board



Ethernet
- ☐ LoRa
- ☐ USB Device
- ☐ USB Host
- ☐ Wifi

Form Factor

- ☐ Arduino Compatible
- ☐ Breadboardable
- ☐ XBee

Interface Firmware

- ☐ No Interface Chip
- ☐ CMSIS-DAP
- ☐ DAPLink
- ☐ J-Link
- ☐ ST-Link

Software Support

- ☐ CMSIS RTOS

NUCLEO-F302R8
- Cortex-M4 + FPU, 72MHz
- 64KB Flash, 16KB SRAM
- DAC OPAMP CAN USB

NUCLEO-L152RE
- Cortex-M3, 32MHz
- 512KB Flash, 80KB SRAM
- LCD DAC OPAMP USB

NUCLEO-L053R8
- Cortex-M0+, 32MHz
- 64-KB Flash, 8-KB SRAM
- LCD DAC USB

NUCLEO-F401RE
- Cortex-M4 + FPU, 84MHz
- 512-KB Flash, 96-KB SRAM
- USB_OTG_FS SDIO

NUCLEO-F030R8
- Cortex-M0, 48MHz
- 64-KB Flash, 8-KB SRAM

NUCLEO-F072RB
- Cortex-M0, 48MHz
- 128-KB Flash, 16-KB SRAM
- DAC CAN USB

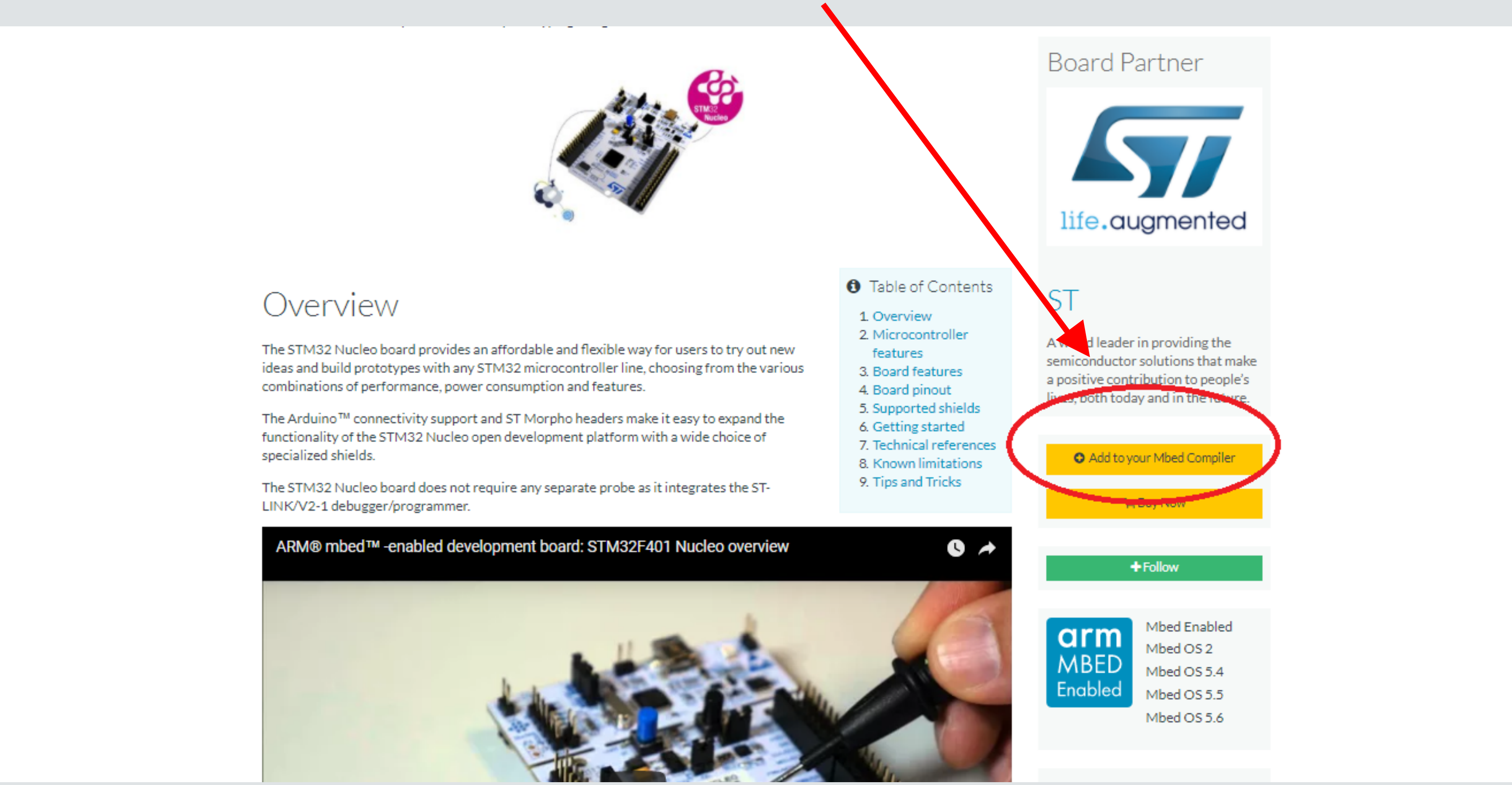
NUCLEO-F334R8

FRDM-K64F

Nordic nRF51822

STEPS to Start Development

- STEP 7 – Add Board to mbed Compiler



Overview

The STM32 Nucleo board provides an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller line, choosing from the various combinations of performance, power consumption and features.

The Arduino™ connectivity support and ST Morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.

The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger/programmer.

Table of Contents

1. Overview
2. Microcontroller features
3. Board features
4. Board pinout
5. Supported shields
6. Getting started
7. Technical references
8. Known limitations
9. Tips and Tricks

ARM® mbed™ -enabled development board: STM32F401 Nucleo overview

Board Partner

ST

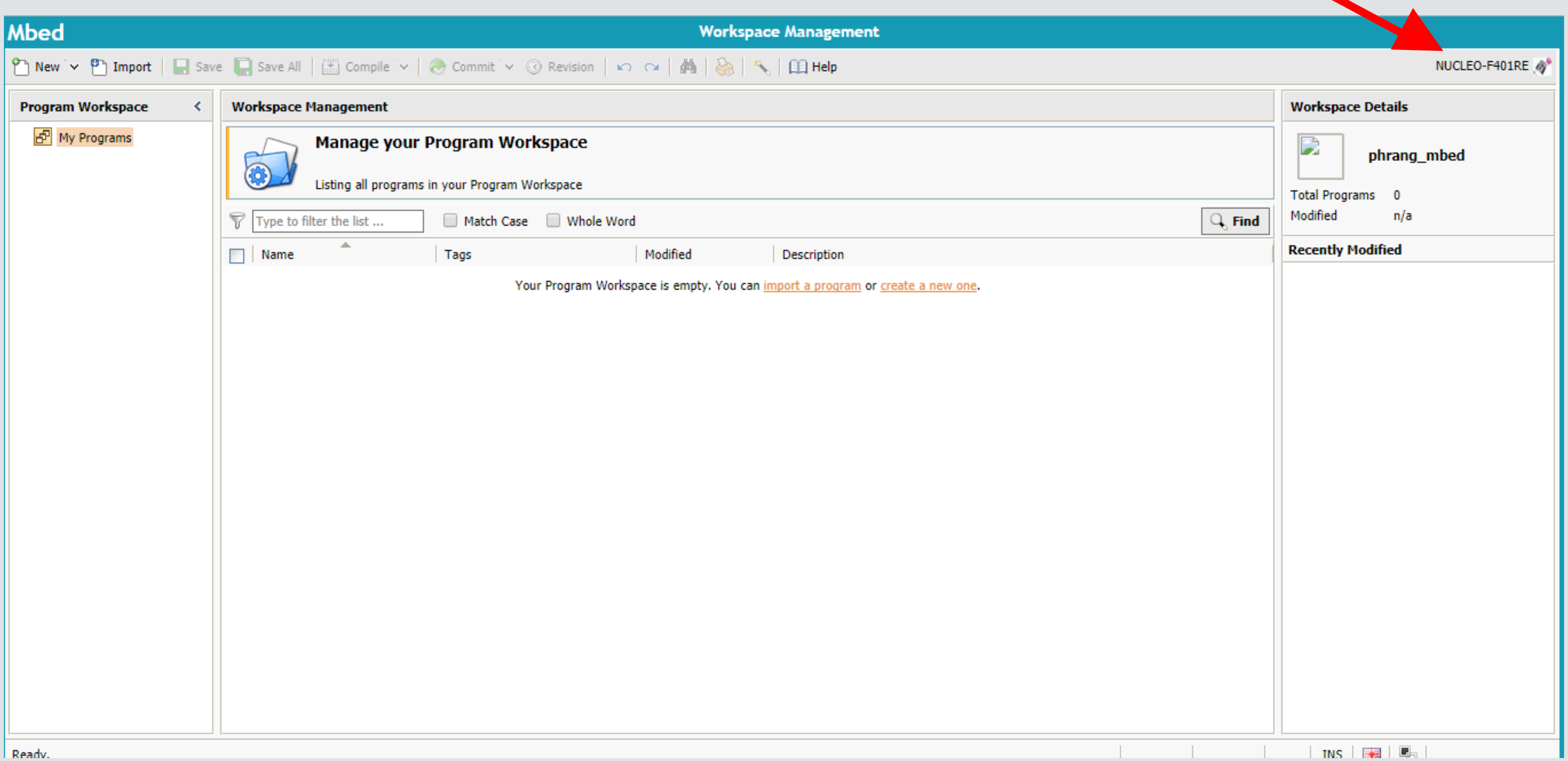
A world leader in providing the semiconductor solutions that make a positive contribution to people's lives, both today and in the future.

Adding other Boards

- You can select any board as in step 6.
- The same can be added to the mbed compiler as in step 7
- In this lab we will use the NUCLEO-F401RE board
- Next, we again click on the compiler to begin writing our first programs.

STEPS to Start Development

- STEP 8 – Goto the compiler and verify if NUCLEO-F401RE is selected as the compiler platform



Manage your Program Workspace

Listing all programs in your Program Workspace

Type to filter the list ... □ Match Case □ Whole Word

Your Program Workspace is empty. You can import a program or create a new one.

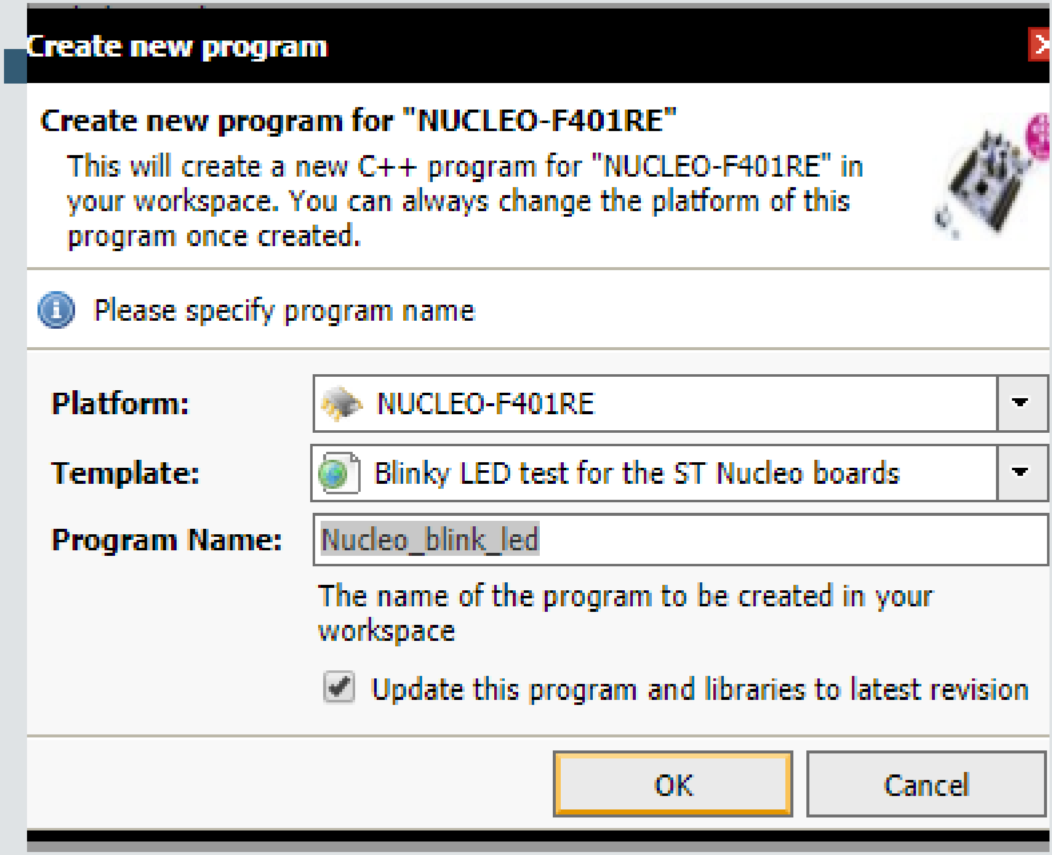
phrang_mbed

Total Programs	0
Modified	n/a

Ready.

Program: Blinking LED

- Click on new -> New Program



Create new program

Create new program for "NUCLEO-F401RE"

This will create a new C++ program for "NUCLEO-F401RE" in your workspace. You can always change the platform of this program once created.

Please specify program name

Platform: NUCLEO-F401RE

Template: Blinky LED test for the ST Nucleo boards

Program Name: Nucleo_blink_led

The name of the program to be created in your workspace

☑ Update this program and libraries to latest revision

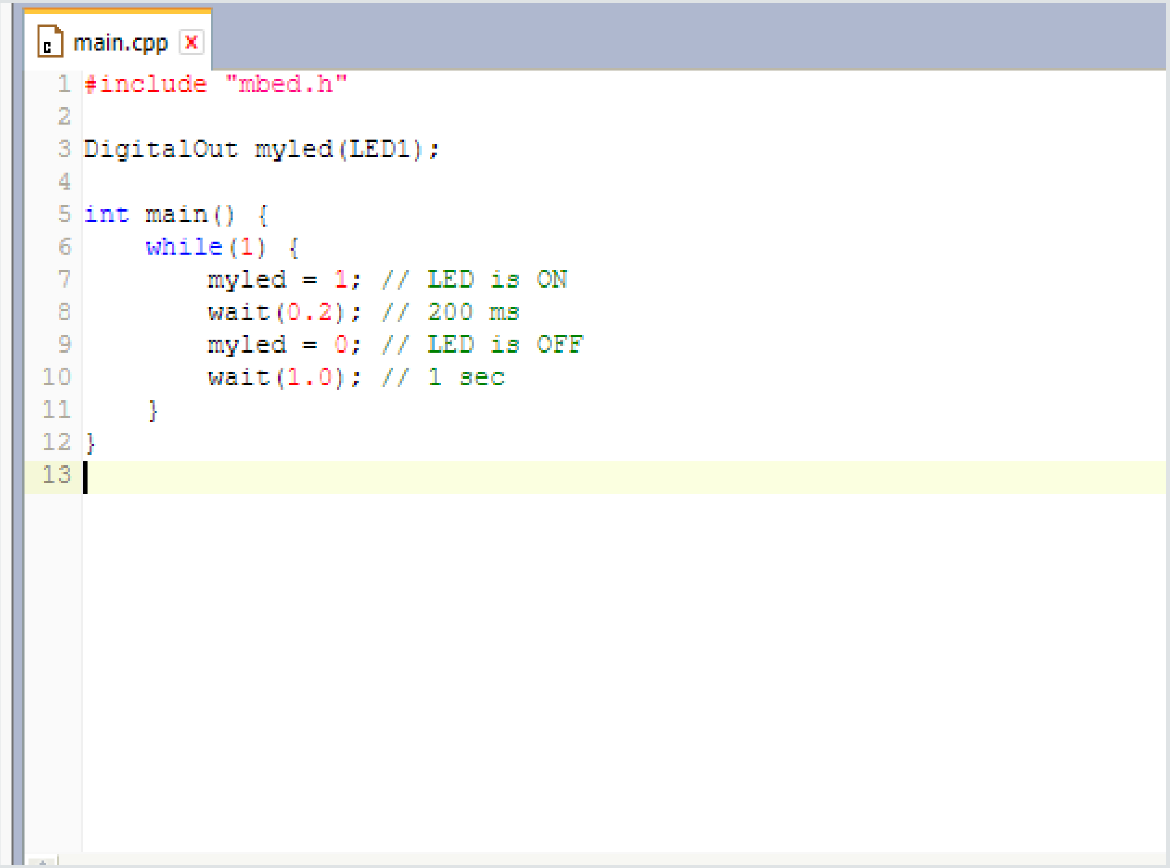
select the kit (need only if there are multiple kits)

select template as the blinking LED test program

Name of the program

Writing program

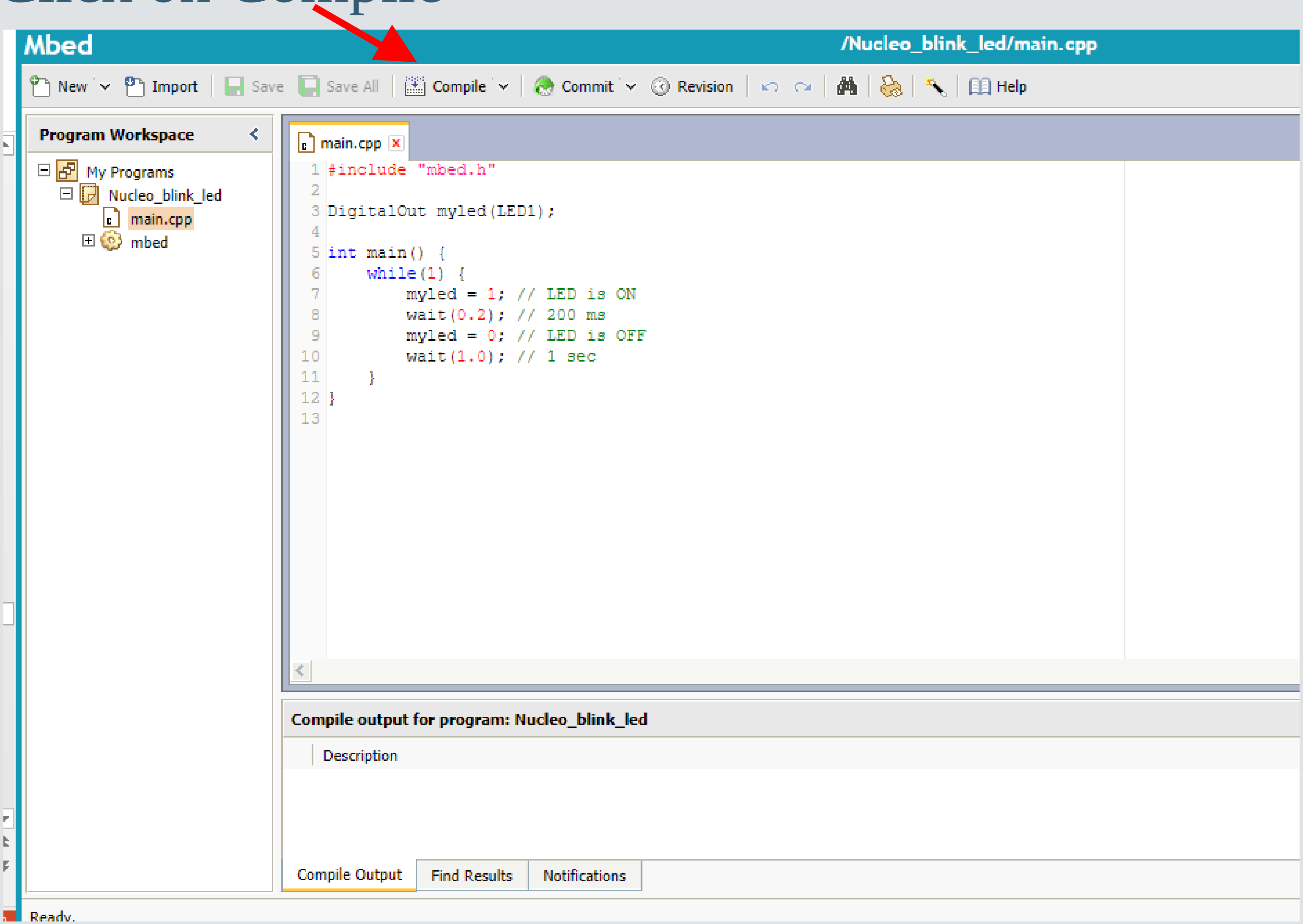
- Click on main cpp



```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {
    while(1) {
        myled = 1; // LED is ON
        wait(0.2); // 200 ms
        myled = 0; // LED is OFF
        wait(1.0); // 1 sec
    }
}
```

Compiling your code

Click on Compile



```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {
    while(1) {
        myled = 1; // LED is ON
        wait(0.2); // 200 ms
        myled = 0; // LED is OFF
        wait(1.0); // 1 sec
    }
}

```

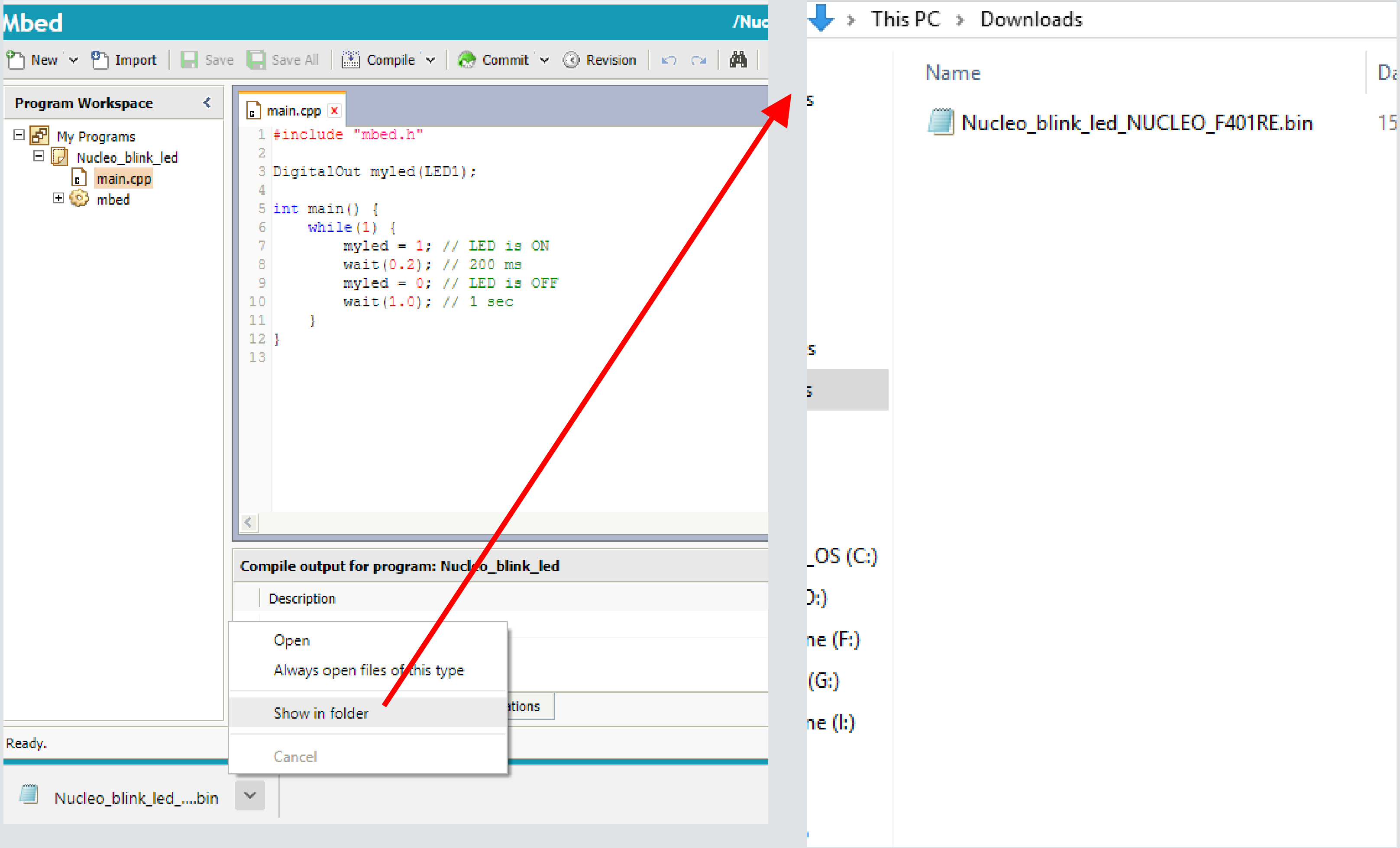
Compile output for program: Nucleo_blink_led

Description

Ready.

Compiling

- On successful compiling you will notice a file is downloaded
- The file is saved by default in the download folder



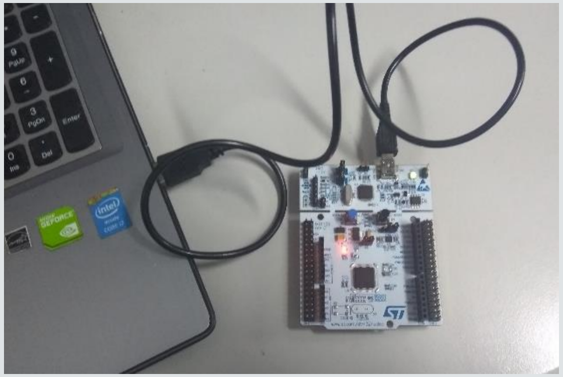
```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {
    while(1) {
        myled = 1; // LED is ON
        wait(0.2); // 200 ms
        myled = 0; // LED is OFF
        wait(1.0); // 1 sec
    }
}
```

Compile output for program: Nucleo_blink_led

Nucleo_blink_led_NUCLEO_F401RE.bin

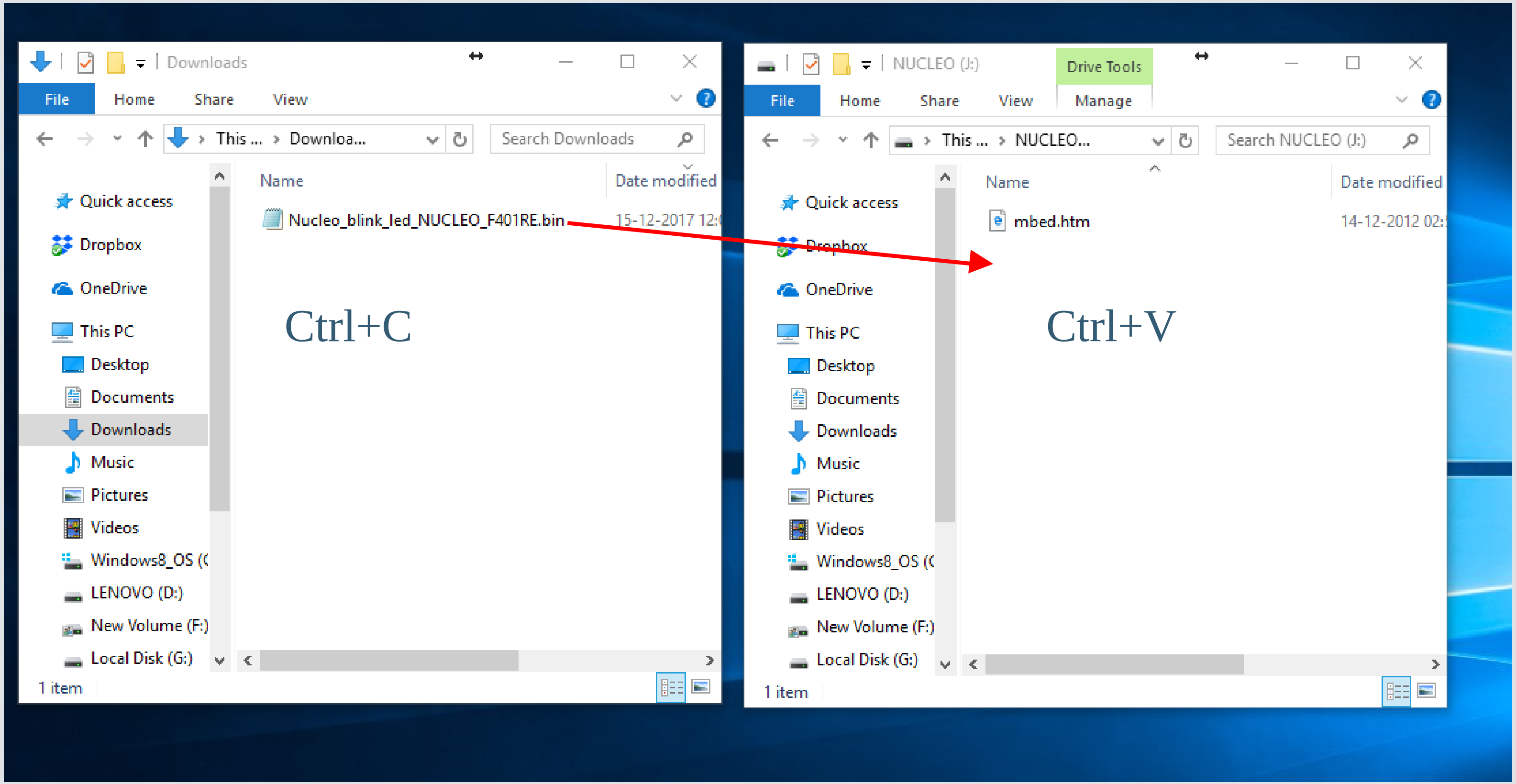
STM32 Connection to PC

- Connect your STM32 NUCLEO to your computer via the USB cable
- Observe that the LEDs are turned on and the USER LED will start blinking



Execution

- Copy the code from the Download folder and paste to the NUCLEO folder



Success

- If everything works, then you have successfully uploaded your program to your evaluation board.
- Now you are ready to develop more systems using this kit.

Thank you

राष्ट्रीय प्रौद्योगिकी संस्थान मेघालय

NATIONAL INSTITUTE OF TECHNOLOGY MEGHALAYA