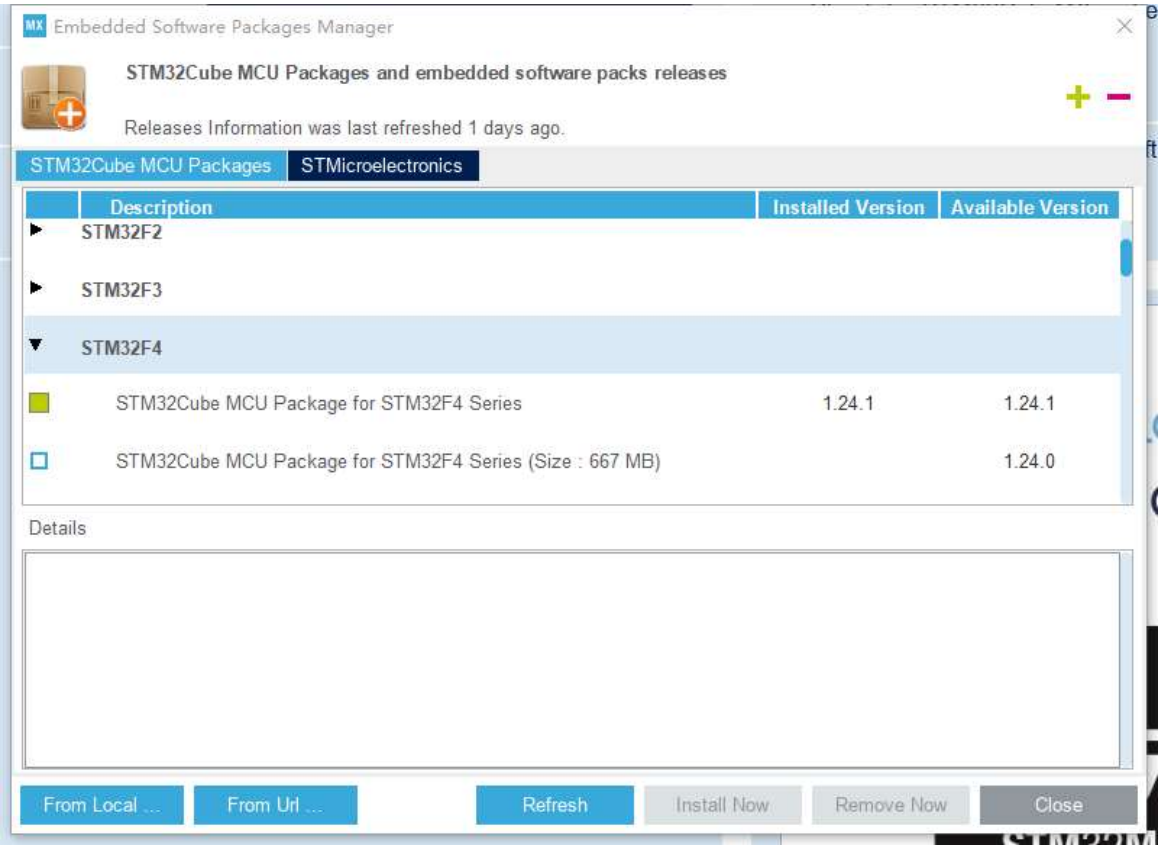
# Onboard assignment for CAN in STM32F427IIHx

## Prerequire:

Installed STM32CubeMX with **STM32Cube MCU Package for STM32F4 Series Ver 1.24.1**




Installed Keil-ARM

Installed J-Link driver

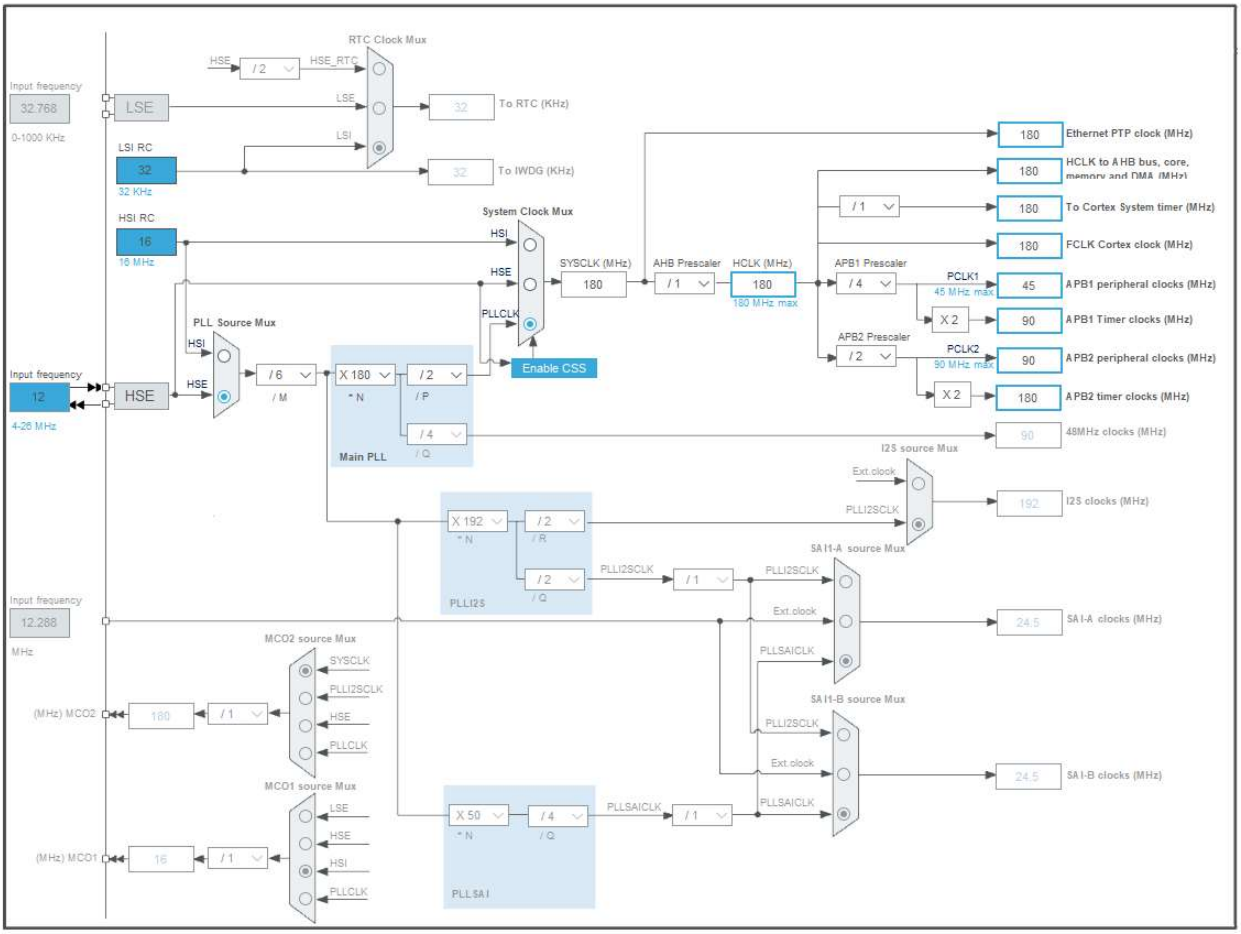
# Create starter code

In STM32CubeMX, click access to MCU selector. Type in “STM32F427IIHx” in the search box, double click the result.

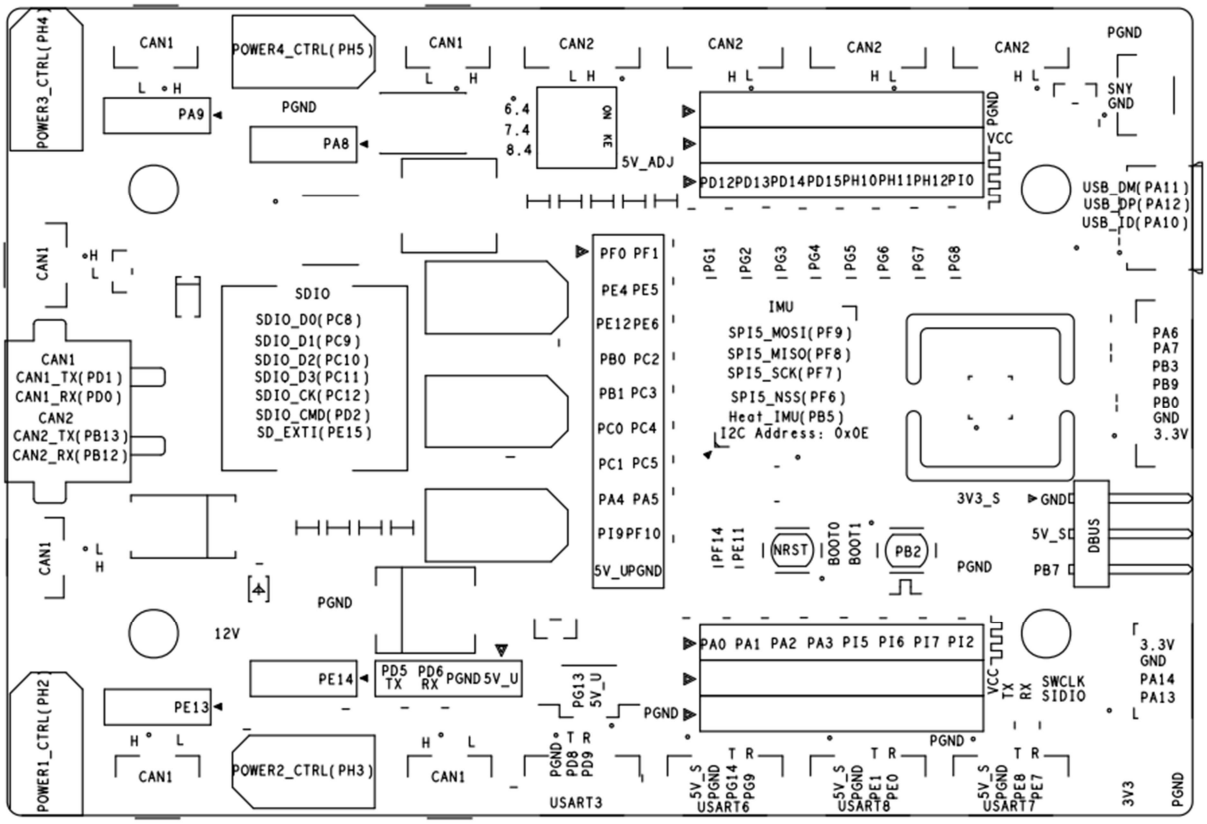
In system core tab, Click RCC and choose Crystal/Ceramic Resonator in High Speed Clock

Then switch to clock configuration tab, set input frequency to 12 and choose HSE as PLL source Mux, and PPLCLK as System Clock Mux. Then in HCLK entry box, type 180. CubeMX will search solution for you and config the clock setting automatically.

The result clock config should look like this:



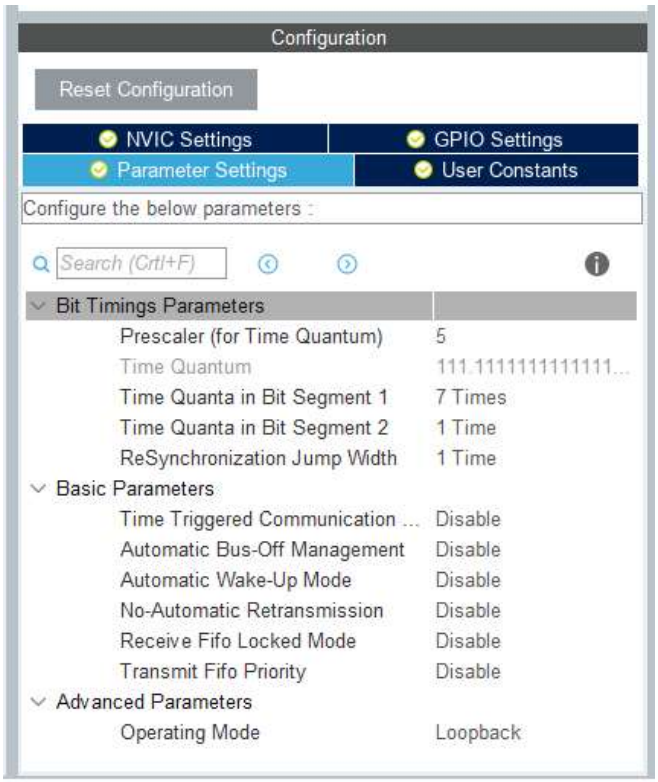

Go back to Pinout & Configuration tab, configure PG1 to PG8 as GPIO_Output.




Add appropriate label for each GPIO pin such as LED1, LED2 …

In connectivity tab, click CAN1 and click Master Mode

In configuration tab, configure according to image below




Then assign PD0 for CAN1_RX and PD1 for CAN1_TX

In NVIC Setting, enable CAN1 RX 0 interrupts

In Project Manager tab, put a name for the project and edit the project location.

In toolchain / IDE box, choose MDK-ARM V5

Click GENERATE CODE

And open project after finish.

# Assignment description:

Read though https://github.com/RoboGrinder-ECE/Wiki/blob/master/STM32L4%20CAN%20new%20API%20and%20migration.pdf

Try to understand how to config the CAN and how to work with the HAL driver.

Follow the pdf to set up and config the CAN, in the infinite loop in main function, keep sending the can message with

CAN stdID = 0x123

CAN DLC = 1

TxData[0] = 1

The value in TxData[0] will increase 1 after each loop, and reset to 1 when reach to 9;

The loop should run 2 times per second.

In the CAN RX Callback, you should check if the RxHeader.stdId = 0x123, then check the data[0].

If the data equals to 1, then togglePin LED1,

If the data equals to 2, then tooglePin LED2,

…

…

…

For all 8 LEDS.

After Finish, flash the program into development board and check it the LED lights up in sequence.

Submit this assignment though Github, push your code into a new branch in github.