# Chapter 2. General-purpose input/output - GPIO

This chapter will show how to

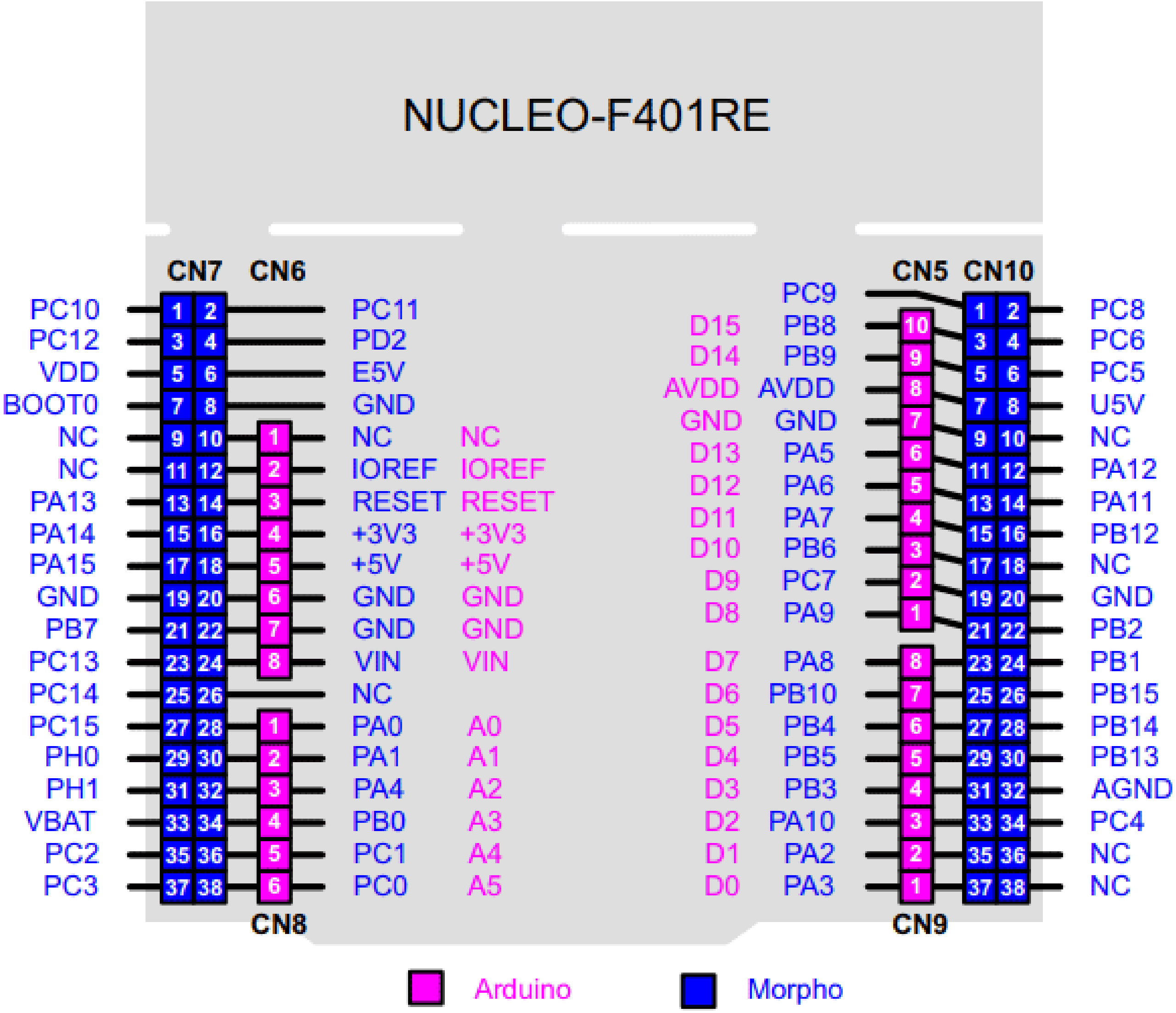
- Configure GPIO pins to input and output
- Set Pull-Down or Pull-up
- Use GPIO as input and output

## Introduction

Microcontrollers have a number of pins that are available for a variety of functions and for developments boards these are typically connected to a header. The features of the pins vary and some pins have multiple usecases meaning that the microcontroller must be setup with code enabling and describing the feature of a pin. The available features of a pin can be found in the pinout of a board. One of the features available for many pins is General Purpose Input Output (GPIO). GPIO sets a pin to be a digital input or output channel which has a wide range of usecases and the setup of this feature will be described in the following sections.

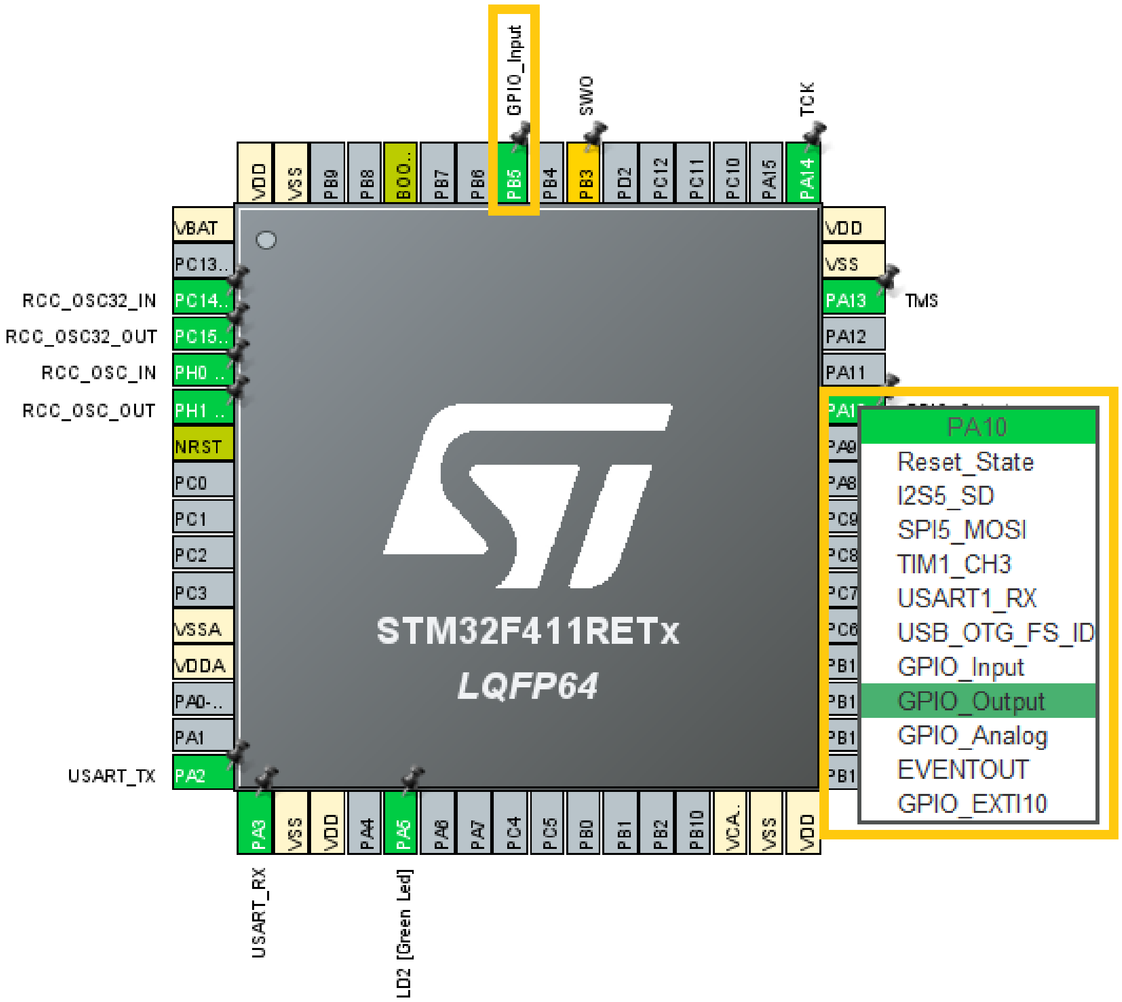
## Setup: Configure one input and one output

First we need to find two GPIO pins that are not reserved on the board. If placement of the pins is important or a special feature (Eg ADC) is required consider the pinout of your board, for this project a NUCLEO-411RE board has been used with the following pinout:



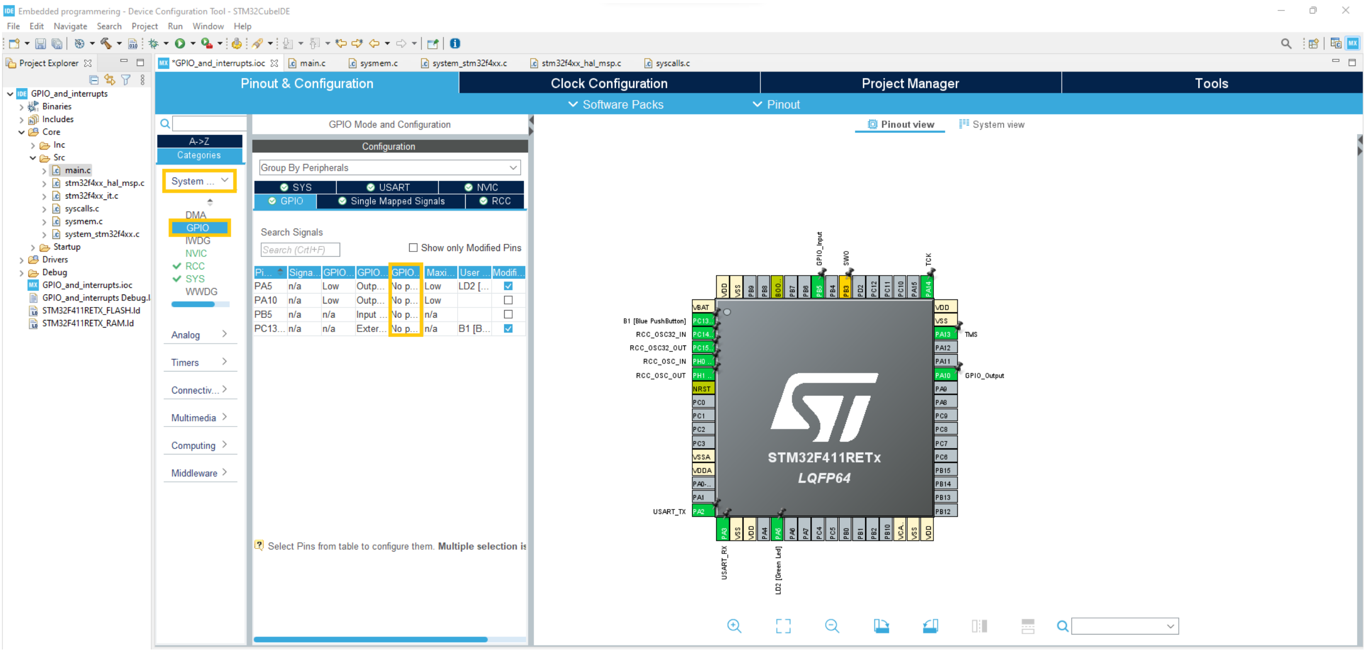
After deciding on pins open STM32Cube and go to the "Pinout and configuratin" file to set the GPIO pins. If not already open this interface is in the "project_name".ioc file in the project folder. From the pin configurator the characteristics of the pin must be declared and these settings will autogenerate the required code to your project. This means also means that a user should only add code in the sections marked with "user code" as it otherwise might be overwritten by the autogenerated code.

A pin is enabled by clicking on it and choosing from the list which feature to use. In this exercise GPIOA pin 10 and GPIOB pin 5 are enabled respectively as GPIO_output and GPIO_input:


GPIO_Input
SWO
TCK
VDD VSS PB9 PB8 BOO.. PB7 PB6 PB5 PB4 PB3 PD2 PC12 PC11 PC10 PA15 PA14
VBAT PC13.. PC14.. PC15.. PH0 .. PH1 .. NRST PC0 PC1 PC2 PC3 VSSA VDDA PA0-.. PA1 PA2
RCC_OSC32_IN
RCC_OSC32_OUT
RCC_OSC_IN
RCC_OSC_OUT
USART_TX
VDD VSS PA13 PA12 PA11 PA1 PA9 PA8 PC9 PC8 PC7 PC6 PB1 PB1 PB1 PB1
TMS
STM32F411RETx
LQFP64
PA10
Reset_State
I2S5_SD
SPI5_MOSI
TIM1_CH3
USART1_RX
USB_OTG_FS_ID
GPIO_Input
GPIO_Output
GPIO_Analog
EVENTOUT
GPIO_EXTI10
PA3 VSS VDD PA4 PA5 PA6 PA7 PC4 PC5 PB0 PB1 PB2 PB10 VCA.. VSS VDD
USART_RX
LD2 [Green Led]

# Exercise 1: Configure input with pull-down

This exercise will make an LED blink when a button is pushed. This exercise requires a few components to complete: 1 Led, 1 button, 1 resistor. After enabling GPIOA pin 10 and GPIOB pin 5 as described in the previous section the next step is to set the pins up with the correct settings. The setup of the pins is done in the left panel of the "Pinout and configuration": System Core-> GPIO:



| Pi... | Signa... | GPIO... | GPIO... | GPIO... | Maxi... | User ... | Modifi... |
|---|---|---|---|---|---|---|---|
| PA5 | n/a | Low | Outp... | No p... | Low | LD2 [... | ☑ |
| PA10 | n/a | Low | Outp... | No p... | Low | | □ |
| PB5 | n/a | n/a | Input ... | No p... | n/a | | □ |
| PC13... | n/a | n/a | Exter... | No p... | n/a | B1 [B... | ☑ |

Select Pins from table to configure them. **Multiple selection is**

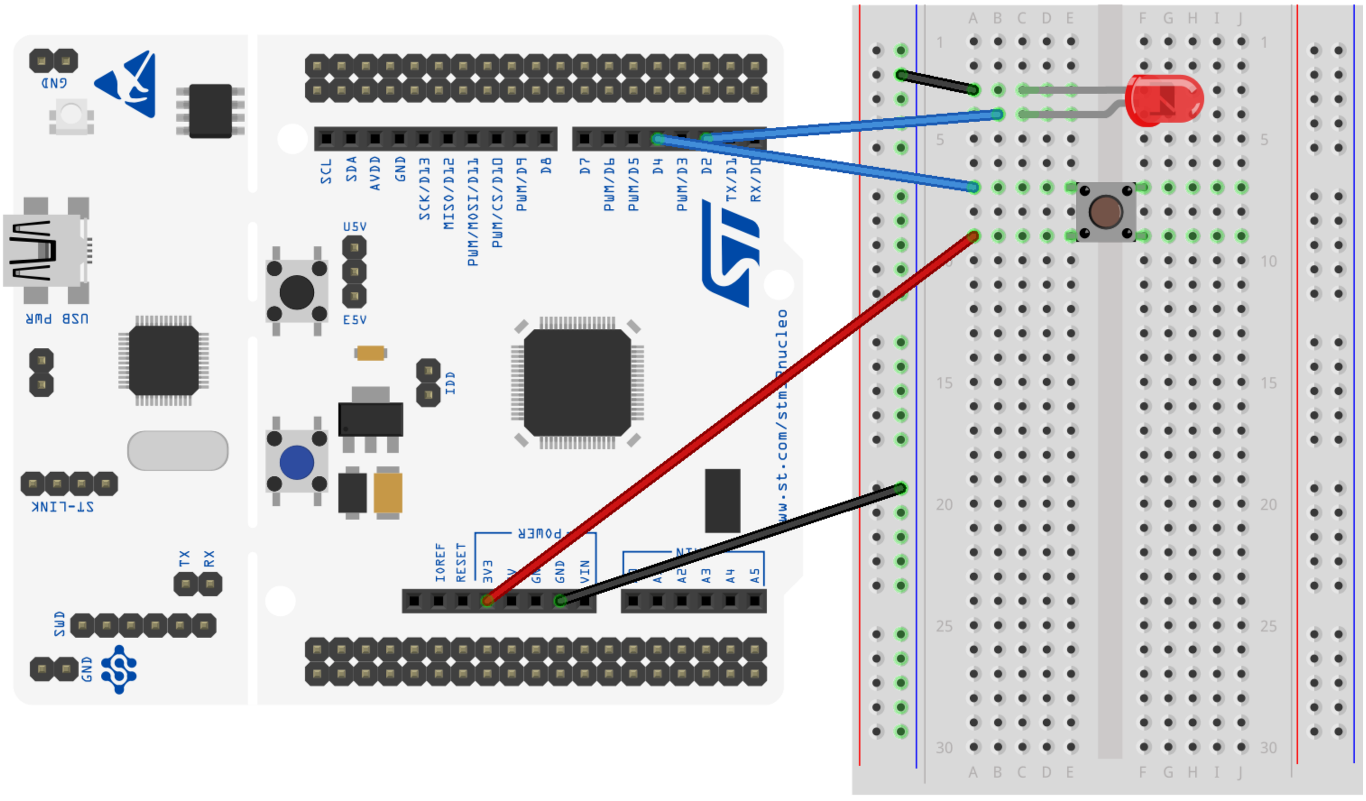
By default all GPIO are set without pull-up or -down. To change this for the output pin GPIOA pin 10, click on it and change GPIO Pull-Up/Pull-down to "Pull-down". This panel can also be used to configure the pins with output level, output mode, output speed or give them a user label. When satisified with the pin configuration hit "ctrl+s" to save and autogenerate the code for the pin configuration.

After the pin configuration the next step is to edit the main.c file. In the "user code"-section in main(void) add following code to make a HIGH output on GPIOA pin 10 when GPIOA pin 5 has a HIGH input:

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_5)){
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_SET);

  } else {
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, GPIO_PIN_RESET);
  }
```

To test if the code works a circuit with an LED and a button is made, (to simplify the circuit the series resistor for the LED is left out). The circuit and the NUCLEO-411RE can be seen below:



fritzing

## Exercise 2: Configure GPIO speed

If further interested in this topic one can play with the speed of the GPIO output by changing it in the pin configuration but this requires equipment to measure with. For furhter information about this experiment consider chapter 5 in "Programming with STM32 Getting Started with the Nucleo Board and C/C++" by Donald Norris.