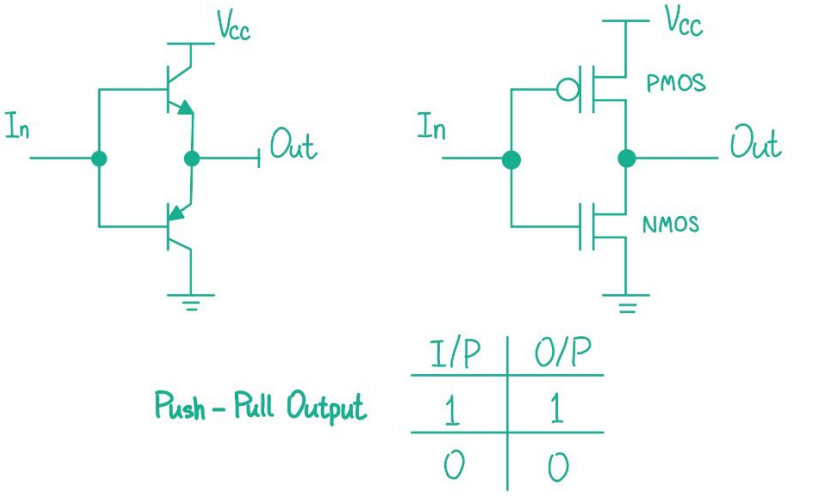
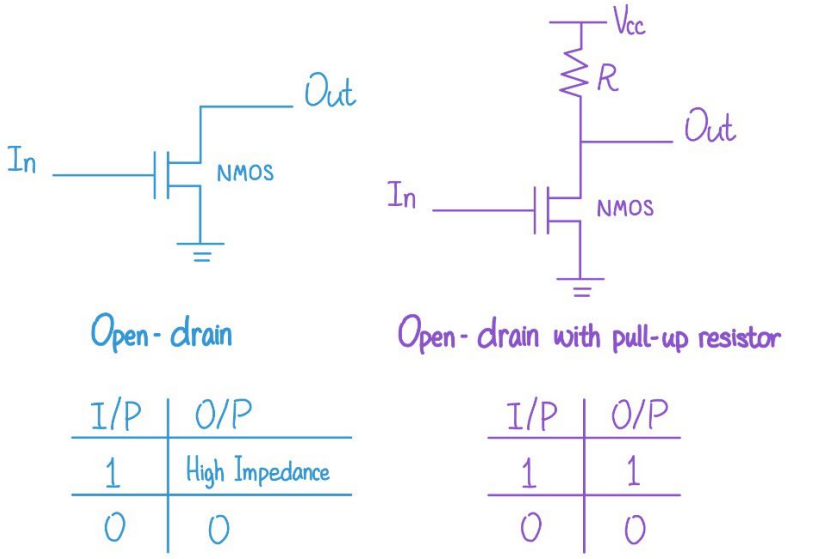
Tutorial 7: MCU

1. In GPIO configuration, what's the difference between output 'push-pull' and output 'open-drain'? To connect LEDs, which one to choose? How about connecting I2C devices? Why?

The 'Push-pull' and 'Open-drain' configurations are shown as follows.



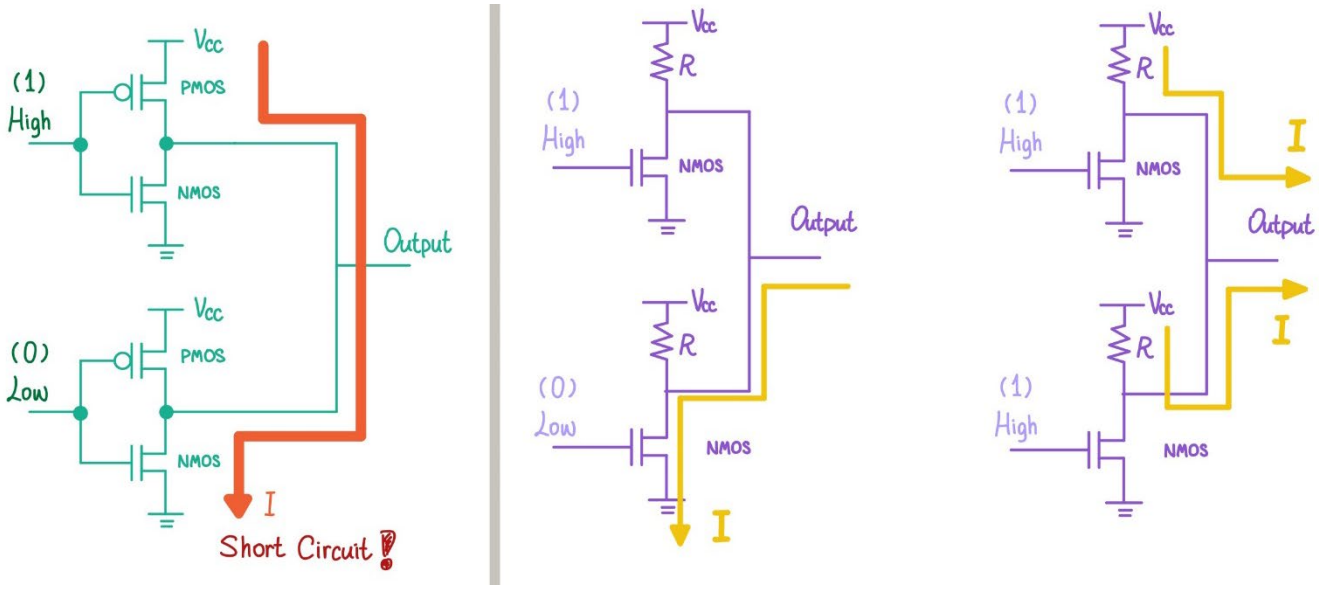


The main difference between 'Push-pull' and 'Open-drain' configurations are summarized as follows.

- **Push-pull** output is best suited for communication interfaces that have **single-direction lines** (e.g., SPI, UART etc.). **Open drain** is commonly used for **bidirectional single line communication interfaces**, where more than two devices are connected on the same line (e.g., I2C).
- **Open-drain** output has **higher power consumption** during active transfers due to the pull-up resistors that are used (as shown in the purple graph above).
- In general, the **push-pull** output has **faster slopes** than the open-drain output.

However, when you connect two push-pull outputs in parallel, if one input is HIGH while the other is LOW, the current will flow directly from the upper Vcc to the lower ground (as shown in the figure below). In that way, the resistance of the entire configuration is meagre, and a short circuit may happen. Therefore, the 'Push-pull' output cannot implement the Wire-OR connection.

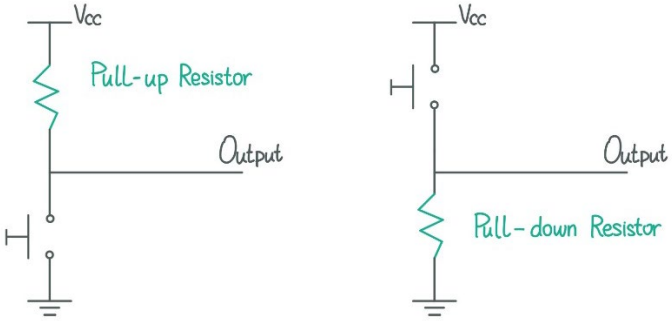
On the other hand, if you connect two 'Open-drain' outputs in parallel, they will behave like an AND gate.



LEDs only have single master-slave stage. Therefore, 'push-pull' is used.
I2C device can handle multiple master devices/peripheral (slave) devices. Therefore, 'open-drain' is used.

2. In GPIO configuration, under which circumstances, we need to configure the GPIO as 'pull-up', 'pull-down', or 'no pull-up or pull-down'? To connect LEDs, which one to choose? How about connecting I2C device? Why?

The configurations of pull-up and pull-down resistors are shown in the figure below.



The **pull-up** resistor connects the unused pin to Vcc to keep the pin **HIGH**.
The **pull-down** resistor connects the unused pin to ground to keep the pin **LOW**.

For LEDs, 'push-pull' is used. Therefore, it uses 'no pull-up or pull-down'.
For I2C device, 'pull-up' is used.

3. If external interrupt has been configured to 'pre-emption priority' 1, timer interrupt has been configured to 'pre-emption priority' 0, Which interrupt has higher priority?

Lower pre-emption value means higher "importance" urgency. In this case, timer interrupt has higher priority (since 0 < 1).

4. Suppose clock source = 16MHz, the timer clock = 8KHz, to create an timer output period = 100ms, what is 'Pre-scaler (PSC) value' and 'Auto-Reload value (ARR)'?

$$PSC = \frac{f_s}{f_c} - 1 = \frac{16M}{8K} - 1 = 1999$$

$$ARR = \frac{Interrupt\ period}{Clock\ period} - 1 = \frac{100m}{\frac{1}{8K}} - 1 = 799$$

5. Let's design a timer using 'PWM generation mode' and configure the PWM generation channel as 'PWM mode 1'. Suppose the clock source = 72MHz, PSC=7199, and we want to trigger two interrupts. The periods of the two interrupts are both 10 seconds. The first one is triggered at the end of the 3rd second in one cycle, and the other one is triggered at the end of one cycle. How should we configure the 'pulse' and 'counter period (ARR)'?

$$f_c = \frac{f_s}{PSC + 1} = \frac{72M}{7199 + 1} = 10KHz$$

$$PULSE = \frac{On\ time}{Clock\ period} - 1 = \frac{3}{\frac{1}{10K}} - 1 = 29999$$

$$ARR = \frac{Interrupt\ period}{Clock\ period} - 1 = \frac{10}{\frac{1}{10K}} - 1 = 99999$$

6. What's the difference between UART and I2C? Hint: full duplex? Half duplex? Transmission Speed? Connect to PC or peripherals?

UART	I2C	SPI
Full Duplex	Half Duplex	Full Duplex
1 – 150K bits/sec	10K – 3.4M bits/sec	60M bits/sec
To PCs	Multiple components	MCU to peripherals

Full duplex: Data can transmit in both direction at the same time.

Half duplex: Data can only transmit in one direction at the same time.