Hands on session 2: Led Switch debounce

Session Introduction: This session introduces concept of floating button, pullup and pulldown resistors and debouncing problem with switches.

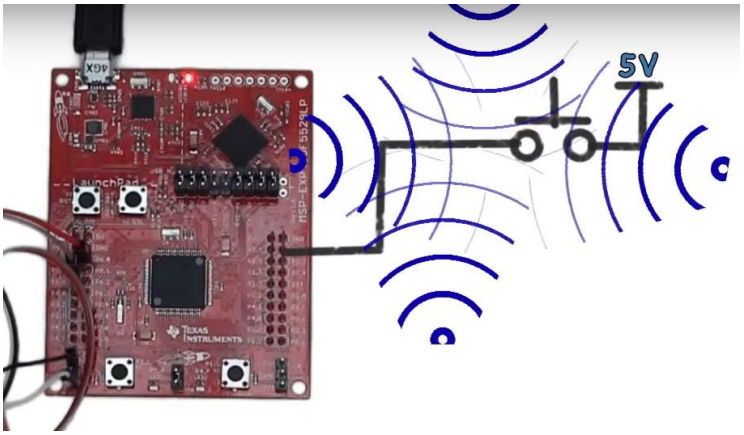
Session Objectives:

- To learn about floating pins and why it is a problem.
- To use pullup and pulldown resistors to solve this problem.
- Introduce debouncing problem and its solution.

Floating Pins

Let's say we have a push button.

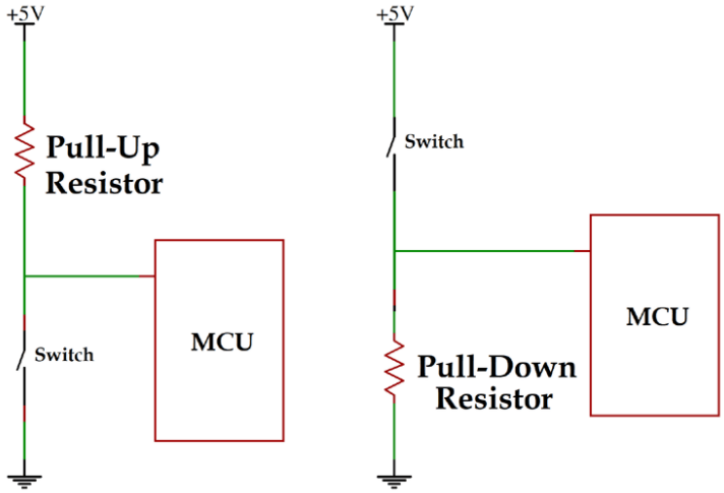
Connected to a microcontroller (in our case esp32) to read digital inputs. We would assume that when the button is pressed it microcontroller will register a high and otherwise it will be low but that's not the case.



Because the gate is not connected to ground; rather, it is floating. The microcontroller may register a low, but it might just as well register a high. By not being connect to a source, Vcc, or GND, the I/O pin is susceptible to electrical noise that makes the I/O randomly fluctuate between low and high. Such sources include thermal noise and electromagnetic interference (EMI) since the leads of the chip act like tiny antennas when they are floating.

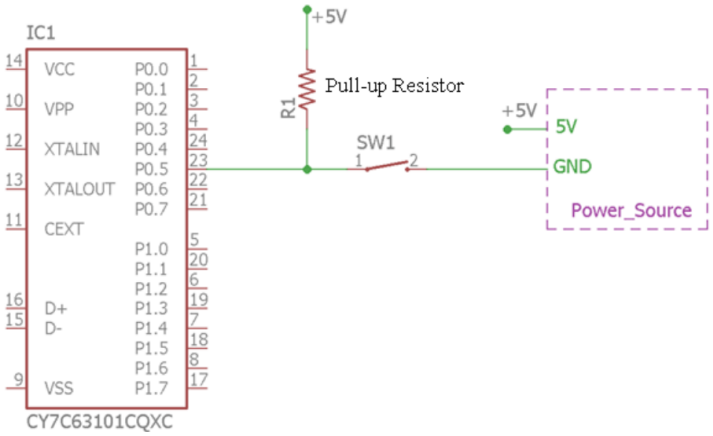
Pullup and pulldown

To counter this problem pullup or pulldown resistors can be used.



A **Pull-up resistor** is used to make the default state of the digital pin as High or to the logic level and a **Pull-Down resistor** does exactly opposite, it makes the default state of the digital pin as Low.

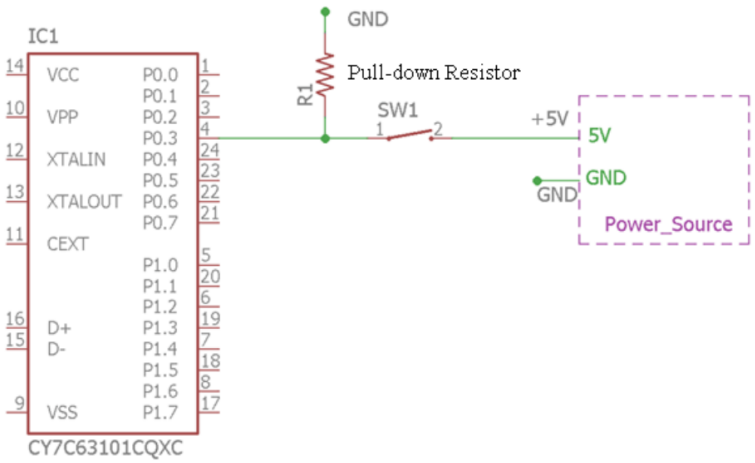
Pull-up Resistors: If we need the high state as default and want to change the state to Low by some external interaction, we can use the Pull-up resistor like the image below.



The digital logic input pin P0.5 can be toggled from logic 1 or High to the logic 0 or Low using the switch SW1. The **R1 resistor is acting as a pull-up resistor**. It is connected with the logic voltage from the supply source of 5V. So, when the switch is not being pressed, the logical input pin has always a default voltage of 5V or the pin is always High until the switch is pressed and the pin is shorted to ground making it logic Low.

However, as we stated that the pin cannot be directly shorted to the ground or Vcc as this will eventually make the circuit damaged due to short circuit condition, but in this case, it is again getting shorted to the ground using the closed switch. But, look carefully, it is not actually getting shorted. Because, as per the ohms law, due to the pull-up resistance, a small amount of current will flow from the source to the resistors and the switch and then reach the ground.

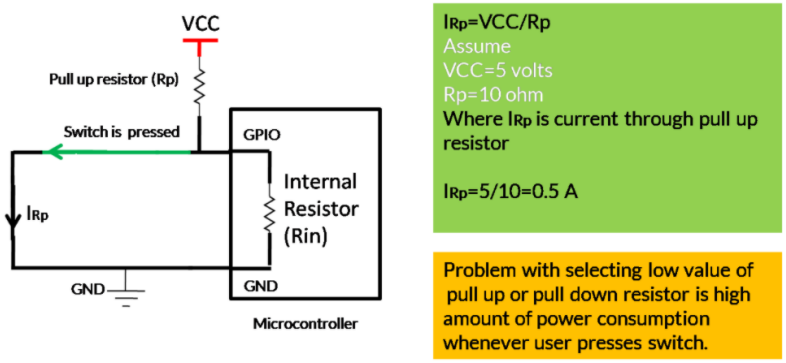
Pull Down Resistor: The same thing is true for the Pull-down resistor**.** Consider the below connection



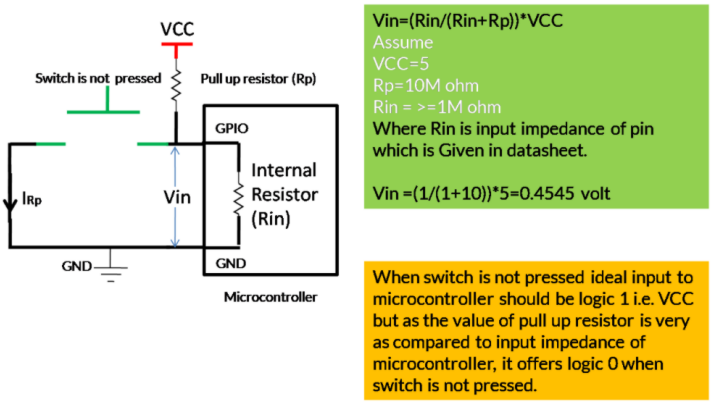
The **pull-down resistor R1 which is connected with the ground or 0V**. Thus, making the digital logic level pin P0.3 as default 0 until the switch is pressed and the logic level pin became high. In such case, the small amounts of current flows from the 5V source to the ground using the closed switch and Pull-down resistor, hence preventing the logic level pin to getting shorted with the 5V source.

Selecting value of the resistor

1. **why we do not use a small value of pull up resistor / pull down resistor**: If we select a **small value** of pull up / pull down resistor like 10Ohm or 100Ohm etc, then when the switch is pressed, a high amount of current will flow through circuit and there is no flow of current through internal resistor (input impedance of microcontroller pin) as it has a high resistance path which is shown below. Problem with selecting a small value of the resistor is that it consumes more power whenever the user presses the switch. Actual problem comes when you are trying to operate microcontroller on battery supply, because of large current flows through the circuit when the switch is pressed and the battery will discharge soon so because of this we can't select a value of resistor as small.



2. **why we do not use a large value of pull up resistor / pull down resistor:** If we select **a large value** of pull up/ pull down resistor like 10M Ohm or 100M ohm etc, then when the switch is not pressed circuit will behave like voltage divider which is shown below. Ideally, pull resistor provides logic 1 when a switch is not pressed but as the value of pull resistor is larger than the input impedance of microcontroller input pin, it offers logic 0 to the microcontroller when the switch is not pressed which is shown below.



3. So, to avoid the above two problems, the value of pull up resistor /pull down resistor should not be small or large. **Normally it is considered as 1% of the input impedance of microcontroller pin**. so normally the value of pull up/ pull down is 10k because the internal input impedance of microcontroller pin is normally in the range of 1M – 10M ohm.

Debouncing Problem:

There is a thing called bounciness, it relates to the physical properties of buttons. When you press a button down, it may not immediately make a complete connection. In fact, it may make contact on one side – then both – and then the other side – until it finally settles down.

This making and breaking contact is called bouncing. It is not a manufacturing defect of the button; bouncing is implicit in most physical switches.

Bouncing happens in a matter of milliseconds but your microcontroller is moving so fast that it will detect a transition between two states every time the button bounces.

There are both hardware and software solutions to this debouncing problem in this course we will solve this debouncing problem using software solutions.

Components Needed:

- Switch button
- Led
- Resistors 220 ohm and 1k ohm
- ESP 32