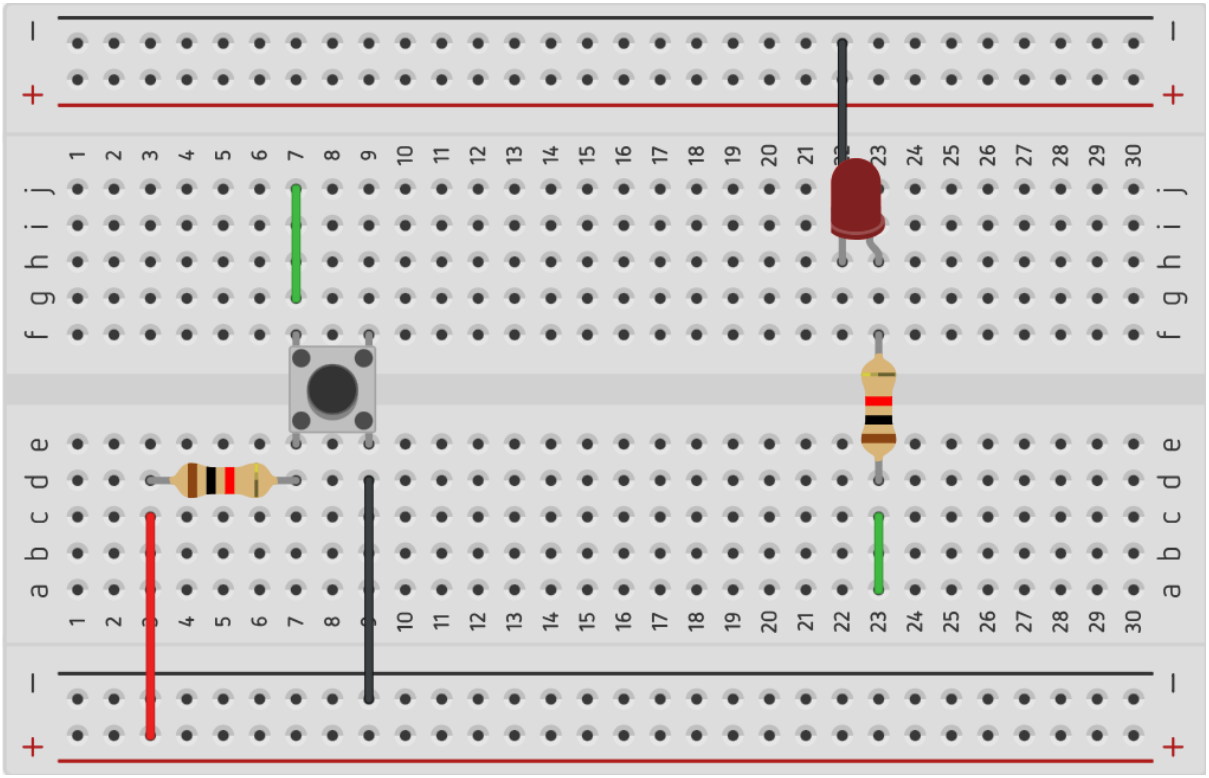
ES LAB 1: LED & PUSHBUTTON.



Circuit used: the red wire is to simulate being connected to the 5V pin. The black wire is to simulate being connected to the GND pin. The green wires are to simulate being connected to my chosen pins that you will read about later.

Student: Johnson Domacasse
Date: 12 sept 2023
Student#: 4471709
Teacher: Suzana Andova

Introduction:

The first lab assignment for ES is to gives us an understanding of how we can use simple bit manipulation, low-level programming and basic circuitry to understand how an Stm32 works. Through out this lab there will be a number of things we go through in order create what needs to be made.

Procedure:

In this section of the report I will be going over what research I did, I step by step guide on where I got the information from and how I translated this into a solution. For reference I will be using the following documents in search for information: UM1724 user manual, STM32F303xE data sheet and the RM0316 reference manual. I also did some self study by going through some videos and PowerPoint presentation that were provided to me. See the above image for my “circuit”.

LED setup:

In order to get the LED to work, we must configure a few things on the board itself. I followed a step by step guide on how I got to my solution.

For input I have to set up a few things. First the clock. Then set up the MODER to output. Then set up the OTYPER to push-pull. Finally I use the ODR to manipulate the bits that I need to write to my output pins.

By following the simple guide I was able to set up my LED. For this Lab I am using pin **PA5** and **PA6** for my LEDs. PA5 internal and the PA6 is a separate pin I chose *specifically*. I tested by simply turning it on the while loop to see if there was an issue in my configuration. Luckily there weren’t any. Now that I had the both LEDs working I went on to the next section.

Button setup:

In order to get the button to work, we use a similar approach as the LED with some minor changes. I set up the clock. Then up the MODER to input. Then set up the PUPDR to no pull-up/pull-down. Finally I use the IDR to read from the pins.

Like before, I used the simple guide to set up my button configurations. For this Lab I will be using the internal button assigned at PC13. I will use another *specifically* chosen pin PC7 for my external button. I tested to see if it works by using a simple program to turn the LED on if the button is pressed. I did this for both buttons to see if there were any issues. They work fine.

Final solution:

Now That I have both my buttons and LEDs working it was a matter of combining the two in order for them to work. When the first button is pressed, it will turn an LED on and when it is released it will turn the LED off. When the second button is pressed, it will turn toggle the other LED on and off until the button is released. I also stated I chose Port A and port C ports specifically. This is because I want to conserve some power. I could’ve also used port B or D or any of the other ports. The reason for the decision was simply to conserve power usage.

Something cool:

For my something cool section I wanted to see how power is being transferred to the LED pins when the buttons are pressed. Ideally I would've used an oscilloscope for this. I used a multi meter to measure the voltage everytime the button is pressed. When the button is pressed the results will show that there is some voltage being sent to the LED.