Jeffrey Razon

**CPE301 – SPRING 2018**

Design Assignment 02

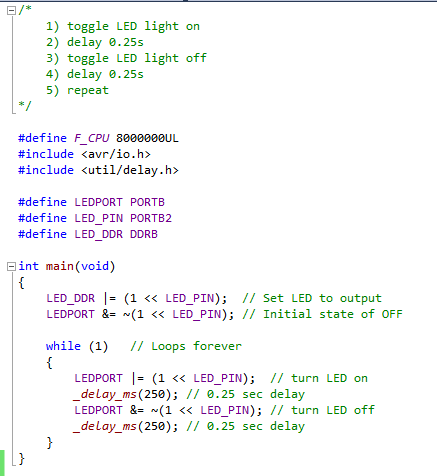
**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

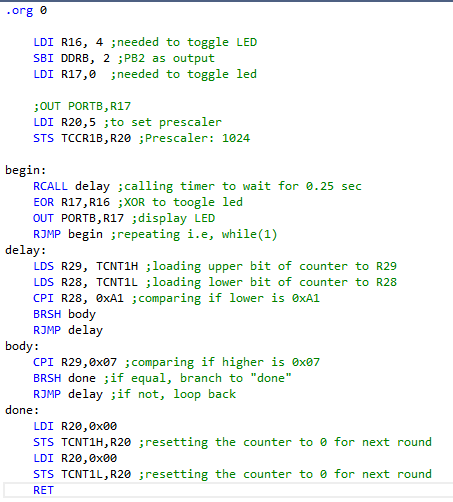
|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 1 | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 2. | INITIAL CODE OF TASK 1/A |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 4. | SCHEMATICS |  |  |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 5. | SCREENSHOT OF EACH DEMO |  |  |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

**Task 1/A**: Design a delay subroutine to generate a waveform on PORTB.2 with 50% DC and 0.5 sec period. (I was told that just having the LED light turn on/off every 0.25s would suffice.)

a) C code



b) Assembly code



c) Timing Proof

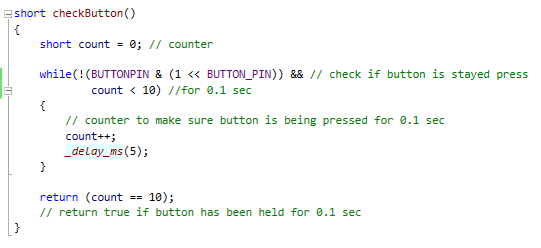
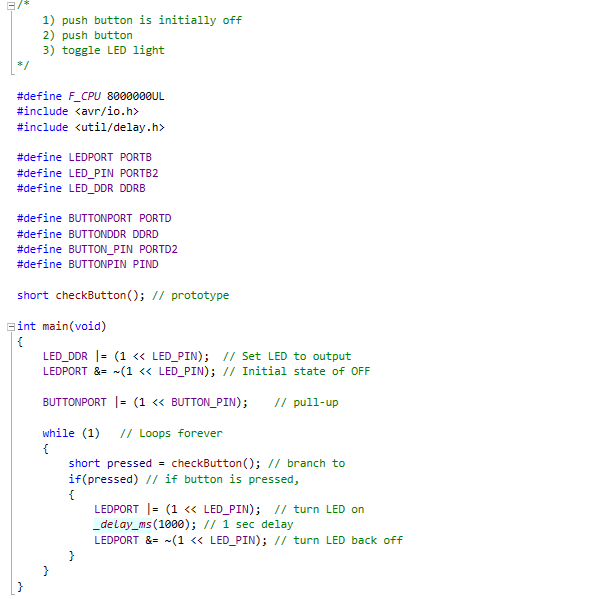
In /da02\_screenshots/TimingProof folder, under the names:

Assembly code: *ASM\_T1.PNG*

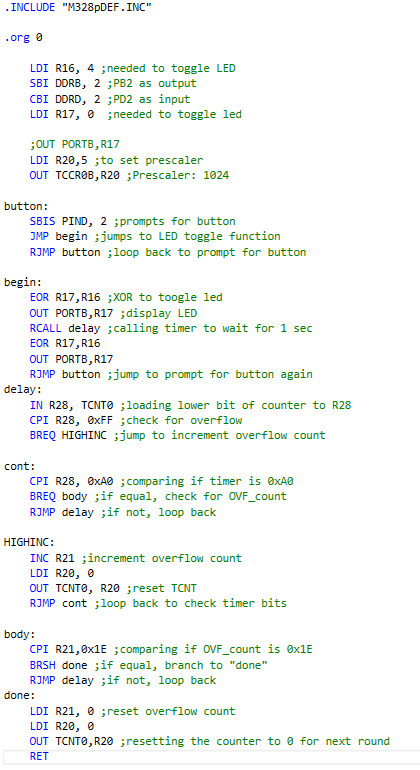
C code: *C\_T1.PNG*

**Task 2/B**: Connect a switch to PORTD.2 (active high - turn on the pull up transistor) to poll for an event to turn on the led at PORTB.2 for 1 sec after the event.

a) C code



b) Assembly code



c) Timing Proof

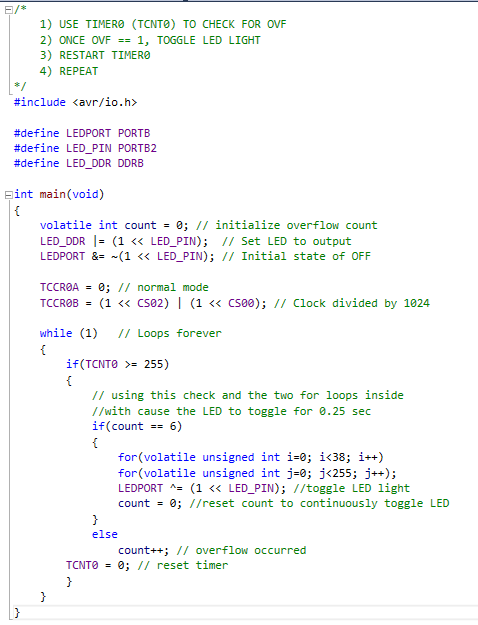
In /da02\_screenshots/TimingProof folder, under the names:

Assembly code: *ASM\_T2.PNG*

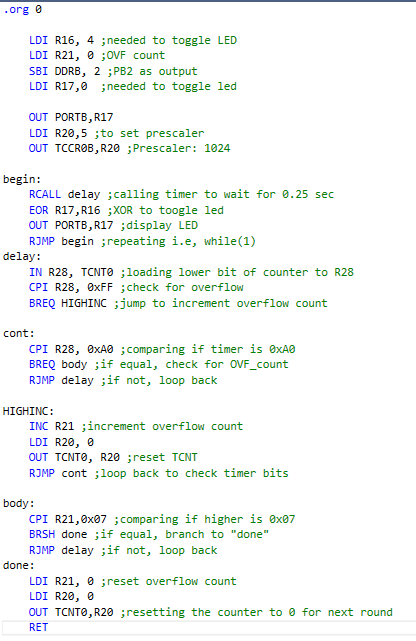
C code: *C\_T2.PNG*

**Task 3/C**: Implement Task 1 using Timer 0. Count OVF occurrence if needed. Do not use interrupts.

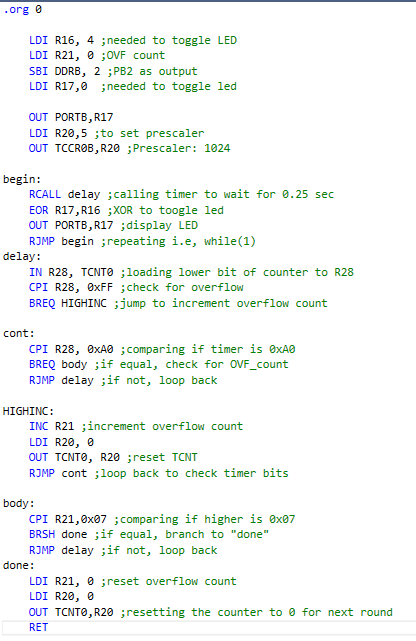
a) C code



b) Assembly code



(assembly code cont.)



c) Timing Proof

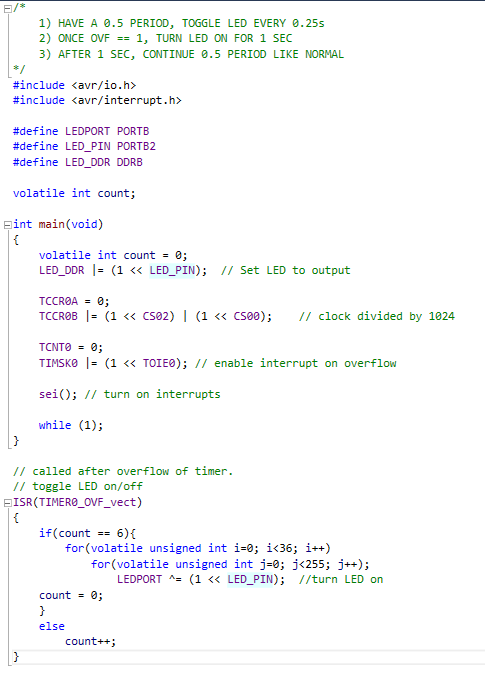
In /da02\_screenshots/TimingProof folder, under the names:

Assembly code: *ASM\_T3.PNG*

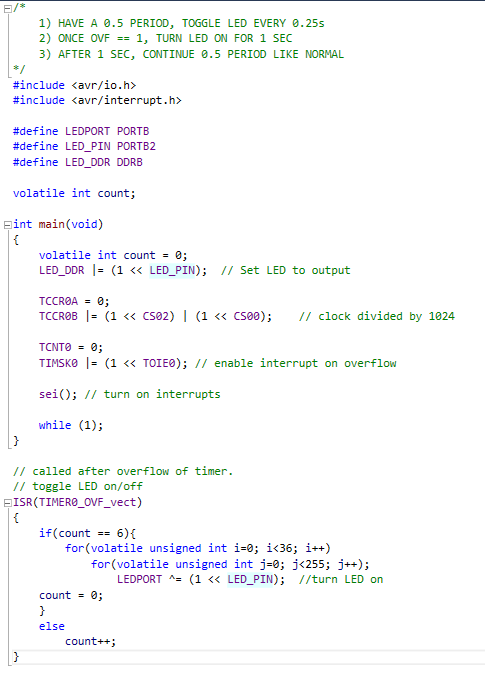
C code: *C\_T3.PNG*

**Task 4/D**: Implement Task 1 using TIMER0\_OVF\_vect interrupt mechanism.

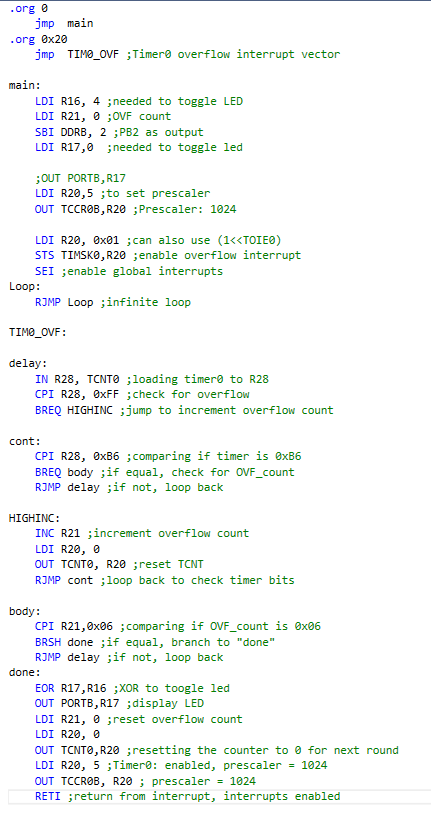
a) C code



(c code cont.)



b) Assembly code



c) Timing Proof

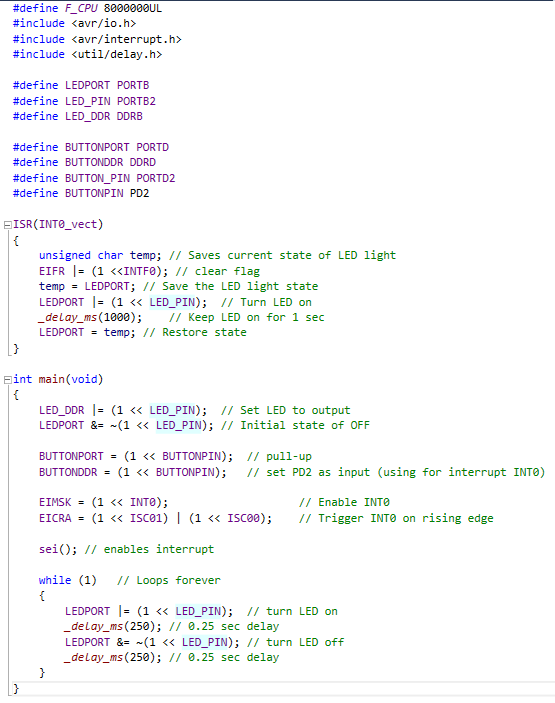
In /da02\_screenshots/TimingProof folder, under the names:

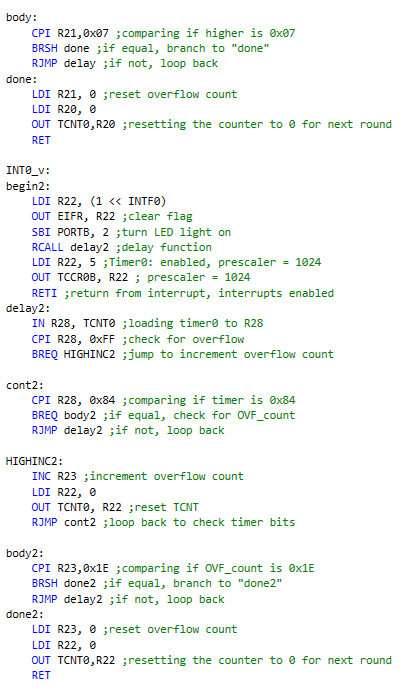
Assembly code: *ASM\_T4.PNG*

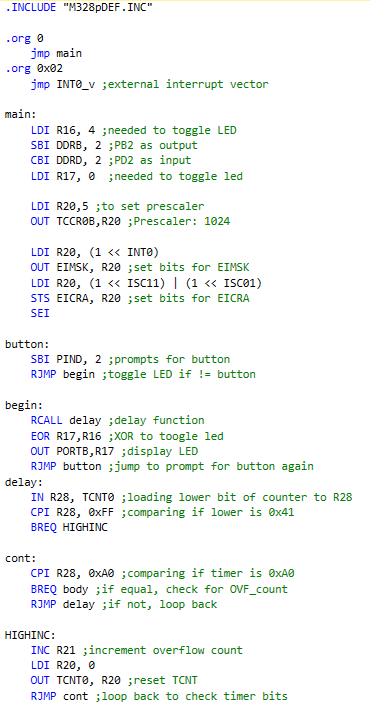
C code: *C\_T4.PNG*

**Task 5/E**: Implement Task 2 using INT0 interrupt mechanism.

a) C code



b) Assembly code



c) Timing Proof

In /da02\_screenshots/TimingProof folder, under the names:

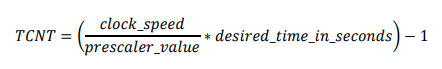
Assembly code: *ASM\_T5a.PNG* and *ASM\_T5b.PNG*

C code: *C\_T5.PNG*

**Calculations:**

For the purpose of obtaining the correct values for the delays, I used the TCNT formula to get the proper numbers for each task.

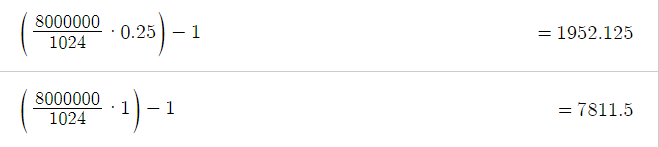
a) TCNT formula



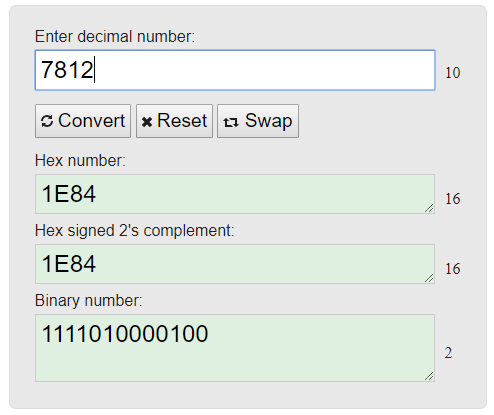
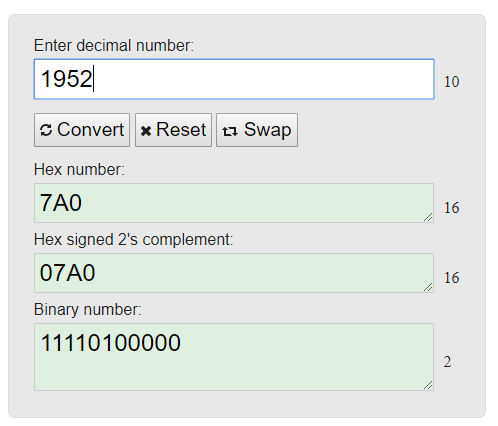
b) Plugging in values:

1st formula: 8000000 represents clock speed (in Hz), 1024 represents prescaler value, desired time is 0.25 seconds; the output value is 1952

2nd formula: 8000000 represents clock speed (in Hz), 1024 represents prescaler value, desired time is 1 second; the output value is 7812

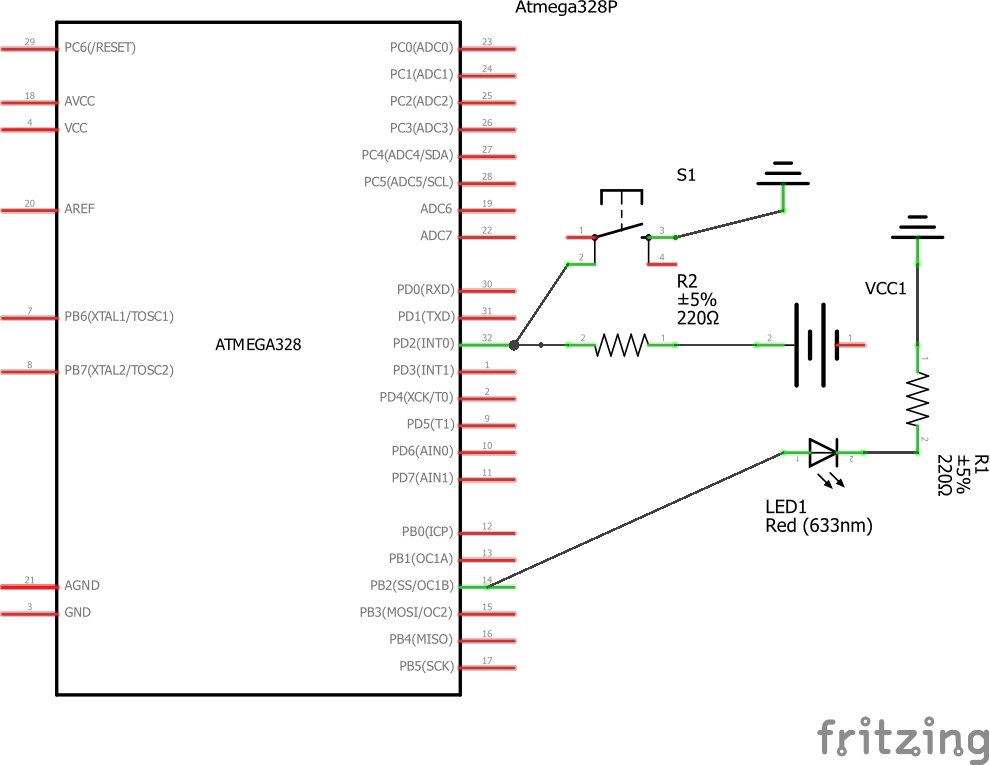
****

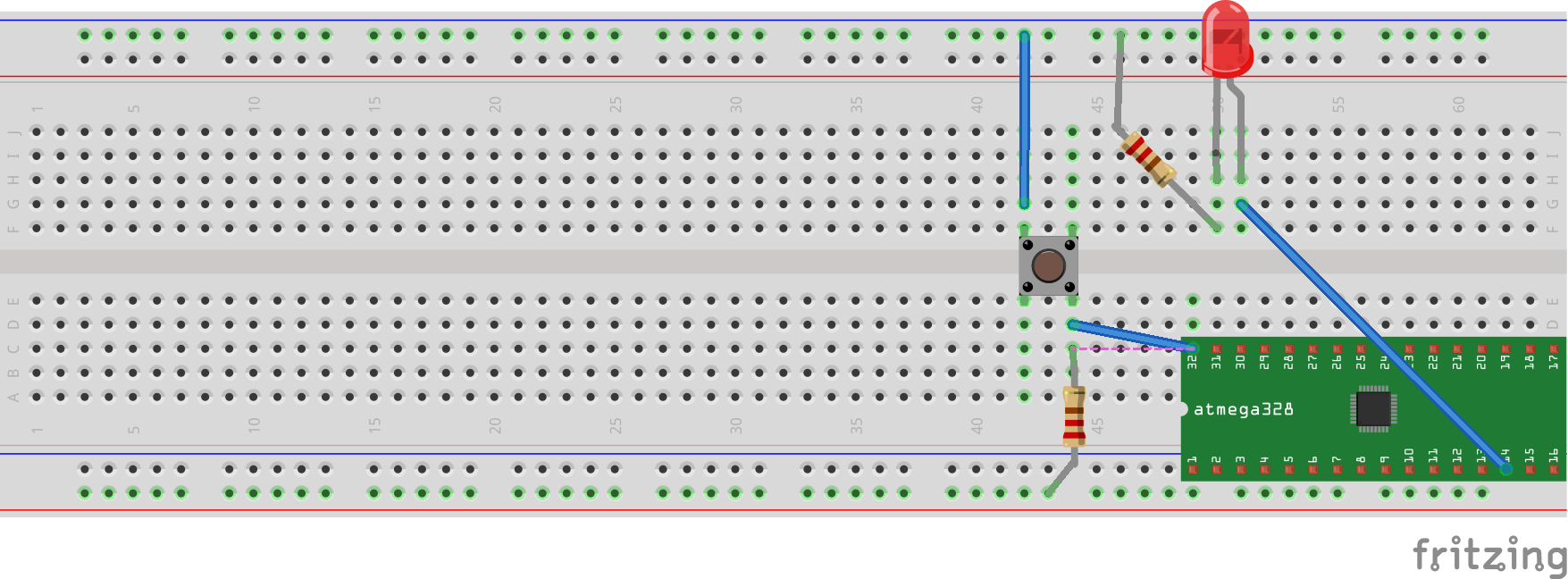
c) converting 1952 to hex d) converting 7812 to hex

****

**Schematics**:

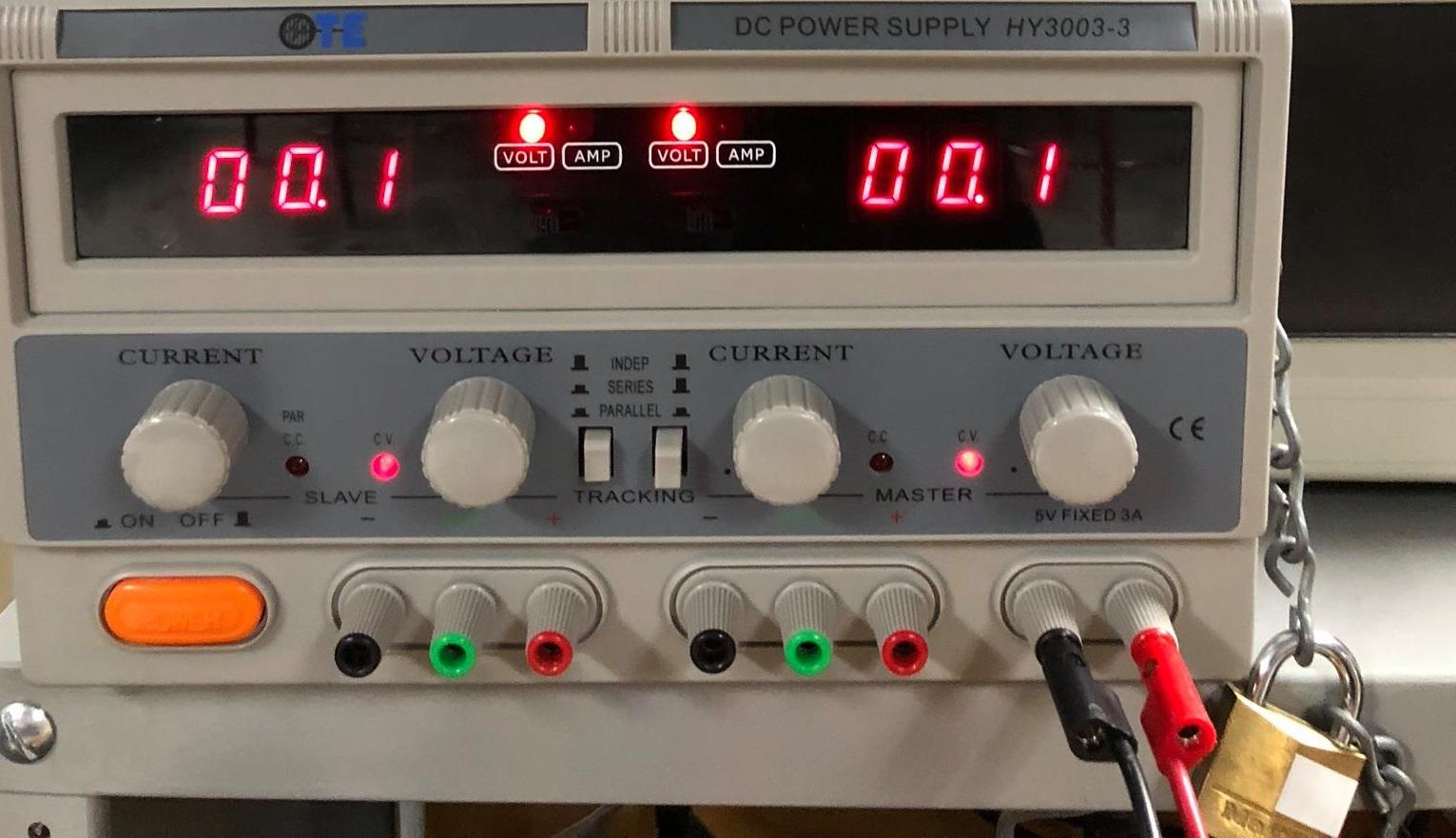
This schematic is the basic setup for this assignment. PORTB2 was used in all 5 tasks (website used to build this schematic did not let me connect the resistor and ATmega to the led, so the connection in the picture below was the best I could do). PORTD2 was used in task 2 and 5. The ATmega328p was connected to the ISR cable to my laptop and a VCC and GND source, which connected to the power supply (delivering a fixed 5V to microcontroller).

a) Fritzing schematics  


b) Fritzing breadboard 

**Physical Set-Up**:

b) the power supply (using fixed 5V)



a) the breadboard set-up



Other notes:

**da02\_screenshots** in the repository contains the photos of the timing proof, schematics, physical setup, and screenshots of all of my code for this assignment.

Brief explanations of the timing proof are in **TP\_README.txt** in the same folder.

Outputs of each demo are shown in the Youtube videos. (It’s difficult to prove with screenshots)

**GITHUB LINK:** https://github.com/JeffinVegas/EmbSys.git

**YOUTUBE LINK:** In the videos\_DA02.txt file

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“*This assignment submission is my own, original work*”.

Jeffrey Razon