Introduction

Background -

For the first lab, using a touch button to cycle through different colors of a RGB LED is a very simple but effective introduction to the course. This project teaches us the fundamentals of hardware-software interaction, circuit prototyping, microcontroller programming and digital input/output control. This project is also particularly helpful because a common feature in consumer electronics is state based LED control, such as different colors indicating different status of appliances, handheld devices and user interfaces. [1]

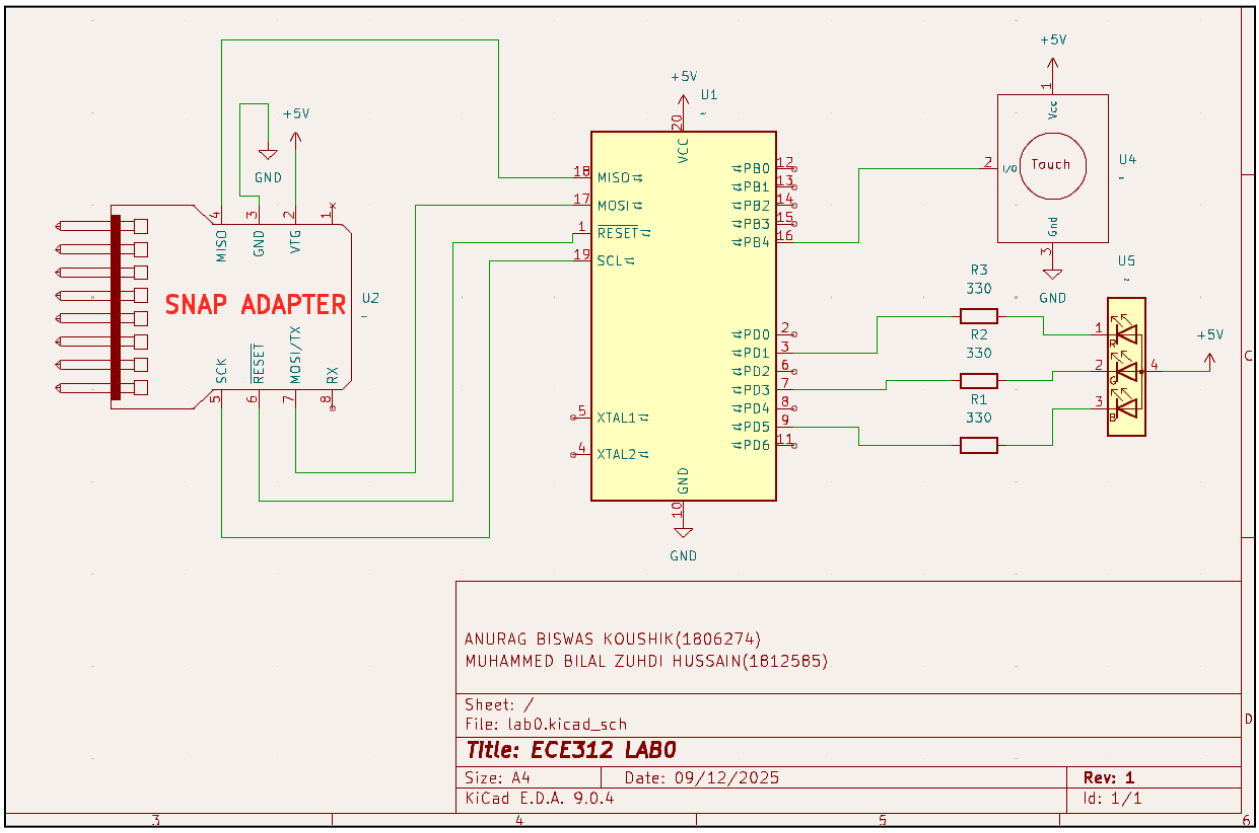
Product Functionality -

The product functions as a small state machine with four defined states. The goal is to have the RGB LED start at the color white, then with each touch to the button, cycle to the next selected color. After the system goes through all the colors, the cycle is to be reset, meaning the color sequence starts from white again cycling to the next colors with each touch to the button. The touch input is also to be debounced, meaning that the last state of the LED is preserved and that it does not cycle through the sequence while the output from the button is high. [1] This functionality demonstrates how embedded systems can implement finite state machines to process user input and drive hardware outputs.

Set-up Environment -

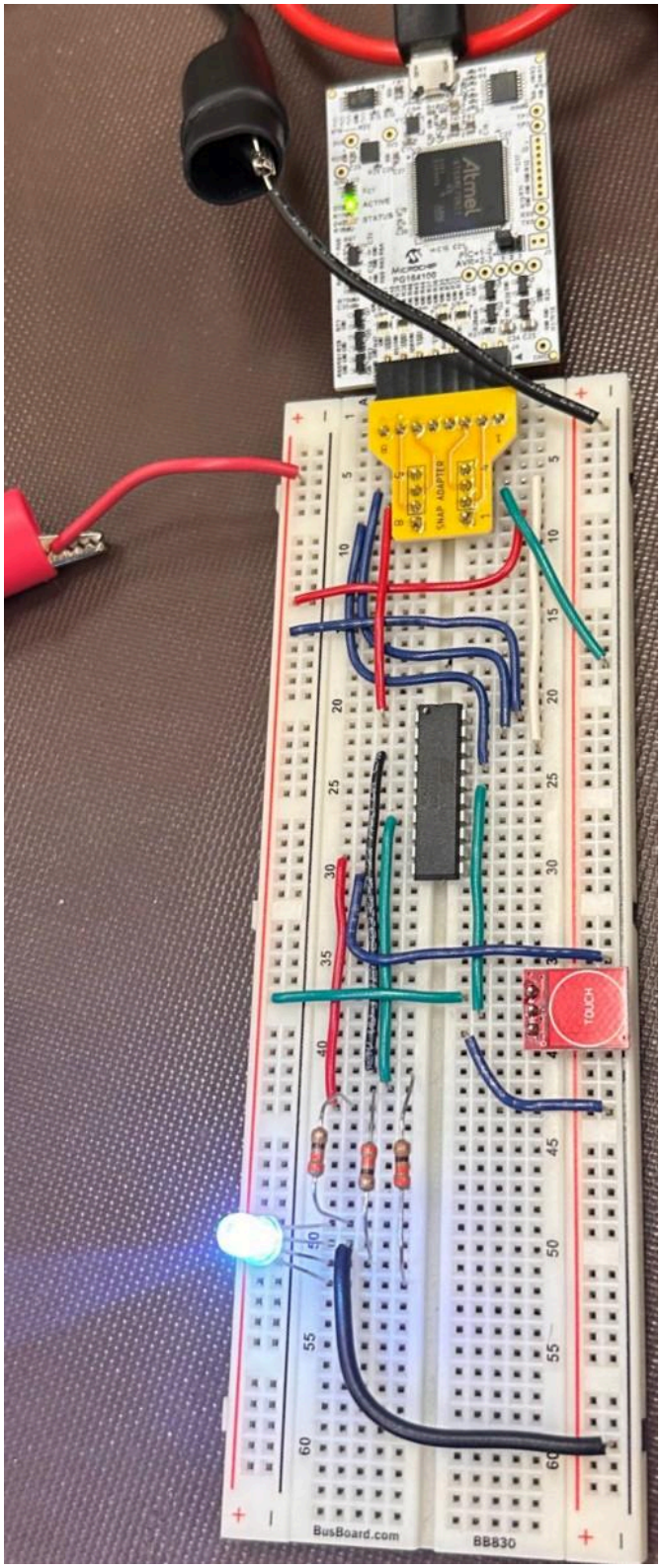
We were instructed to use an ATtiny2313A microcontroller which is programmed via a Microchip SNAP in ISP mode. We used the MPLAB X IDE to write the C code and upload it to the MCU. The schematic for this lab was designed in KiCad using the ECE312 library that was provided to us on Canvas. [1] The circuit consisted of the ATtiny2313A microcontroller, an RGB LED, common cathode or common anode, determined experimentally using a multimeter in diode mode, 330 ohm resistors and a touch button used as input. We were provided with a DC power supply which we set at 5V at 0.3A.

Circuit Schematic



Circuit schematic developed on KiCAD software

Prototype Circuit



Physical prototype of the 4 state led system on breadboard

Conclusion

At the end of the lab, we were able to successfully design and implement the embedded system of controlling a RGB LED using a touch button. The system had four states, containing the four colors and with each press of the touch button, the machine advanced to the next state and then returned to the first state after all the cycles, restarting the sequence again. The system demonstrated reliable detection of each input to the button and the circuit had proper use of current limiting resistors.

There were a number of challenges that we encountered while developing the project. Determining which type of RGB LED we had was critical, as that would change the wiring of the circuit and also a lot of the code used to make the project functional. Figuring out the SNAP module also required a lot of attention as we faced problems with the connections with the lab computer. The SNAP module refused to connect to the lab computer completely and we had to get the TA to assist us. Then there was also the default debugWIRE mode causing difficulties programming the MCU. We had a lot of trouble figuring out how to stop the microcontroller from cycling through the loop of advancing to the next state of the sequence while the touch button was touched. Without proper implementation of logic, while the button was pressed, the microcontroller would keep on advancing to the next state at its core clock speed, which was very fast to the human eye so it appeared white which caused the randomness in the color when the button was let go as the exact time of touch was not consistent every press. We introduced a stall in the loop using a specific value to be assigned to a variable only when the button is pressed and another value to be assigned to the variable when the button is let go, so without the stall variable reverting to the normal state, the state will not advance.

For future iterations of this project, there are a number of ways that the system could be improved to enhance handling and functionality. A hardware debounce could be implemented to further improve the reliability as well as other controls can be implemented such as brightness control, different modes such as strobe and preset patterns of colors. Furthermore, additional colors could be added but adding more states to the system.

References

[1] “Lab 0: Orientation — Control an RGB LED with a Touch Button,” Canvas, University of Alberta. [Online]. Available: Canvas Portal, University of Alberta. [ACCESSED: 25/09/2025].