Rudimentary Input to LED Display

Ryan Zacharias

Department of Engineering, Arkansas Tech University, United States
rzacharias@atu.edu

***Abstract*— This project granted insight into the methods a programmer might use in order to program a calculator display system in the future. By using C, MSP430G2553, and a series of push buttons, a programmer has the tools required to input information and display numbers according to the order they were inputted via an 8 bit LED.**

***Keywords*— C, MSP430G2553, push button, 8 bit LED, microprocessor**

I. Introduction

The contributions from this project shows that this system takes a trivial amount of time to develop due to its simplistic nature while also demonstrating complexities in troubleshooting bugs and determining what systems must go in place to prevent the system from becoming virtually unusable from user error.

II. Literature Reviews

Microcontroller Data Recorder for Solar Energy Hybird CPV-Thermal System Evaluation [1] uses the MSP430 to measure and record information regarding solar energy and its specific yield. The usage of the LED display allows its use in the field.

This project demonstrates that peripherals are important for interacting with computer systems.

III. Methods

Before I wrote code or configured the microprocessor to be compatible with my idea, I first took several moments to think through the prototype process.

I used a breadboard and a series of wires to connect output pins to the onboard MSP430G2553 LEDs, LED1 and LED2. In order to simulate a button press, I connected

I initiated the project by prototyping a basic version of the current completed program. I created two integer variables named “inputArray” and “DELAY”. “InputArray” is used to store inputs, and “DELAY” is used to quickly adjust the number of clock cycles I wanted to skip. Since the microprocessor has 1,000,000 cycles per second, I needed to use the “long” specifier to use 32 bits instead of 8 bits. In other words, the permitted positive maximum number needed to be around 2,000,000 instead of 127.

I then modified the directories of ports 1 and 2 to use input and output features. Port 1 was set to input on pins 5, 6, and 7. Two of these inputs were used to set the LED state to on or off, while the third pin was intended to set an integer array and the counter that accompanied the array to zero - effectively clearing the display. The inputs this port delivered were processed through a series of logic operators and loops to deliver an output through pins 3 and 4 of port 2. ould send voltage to LED1 and LED2 on the board.

After the prototype worked as expected, I wrote out more complicated details that this stage of the project relied on for the hardware to work with the software as expected.

I switched to a breadboard and 8-bit LED after setting the pins of port 2 to “output”. I scripted functions to communicate with the user. One flash of all LEDs confirms that an input was received, while three flashes of all LEDs conveys that there is no more memory allocated to store more inputs. At this point, only the “Clear” command was allowed to execute upon button press.

IV. Conclusion

This project emphasized the importance of planning an approach to solving a problem that has not been solved before. Additionally, testing and comparing the actual output to the expected output assisted my troubleshooting process in eliminating any observed bugs.

References

[1] F. Xie and G. Zhang, "Microcontroller Data Recorder for Solar Energy Hybird CPV-Thermal System Evaluation," *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Nanchang, Jiangxi, 2012, pp. 118-121, doi: 10.1109/IHMSC.2012.35.