­­­­­Rashid Goshtasbi

CS120B

rgosh001@ucr.edu

861056442

Final Project – Simon Says

YouTube: https://www.youtube.com/watch?v=Wl\_CH6R5Fxg

This quarter I had the opportunity to design, code and implement my own project. For this project, I chose Simon Says. To describe my project, we’ll break it up and describe in detail each portion of the game implemented. First I used the ATMEGA-1284 microprocessor. From the processor, I was able to utilize all ports effectively with a few pins left over. From PORTA, I have last 2 pins from the LCD connected to it, followed by the input lines from my large buttons which I had purchased online from SparkFun. The larger buttons allow for easier presses for the user which makes for a more concrete and comfortable gaming experience. Although they’re larger, they are still flimsy like the small buttons included from our kit. PORTB has its pins connected to my LED 8x8 matrix color pins. PORTC was used to control my ground for me LED 8x8 matrix so I can manipulate the matrix to turn on the specific LED’s I want. PORTD was used to connect the required pins from the LCD so that the LCD.

The following is the functionality of the hardware used in the design. Large Buttons were purchased to enhance the user interface so that they did not have to press small buttons. The LCD was used to display the game’s main menu, score, and game process. It’s also used together with the large buttons to guide the user through the menu and make selections. In the LCD, we start off in the main menu, from there the user can start the game depending what level they choose. The level is defaulted to level 1 (easy) upon power but can be changed by entering the menu selection of “choose level.” This functionality enables the user to increase and decrease the levels from one to five. One being the easier with least button presses, while five is the hardest with multiple button presses from the user. The last option in the menu is the high score. The high score is read from the EEPROM and is kept alive even when the power is turned on and off. The high score is determined by the amount of stages within each level completed. When in the game mode of the LCD, the LCD will display begin game as the blinks are running, after which will prompt for the user input depending on the blink number press required. If the game is waiting for user input for the first blink, the LCD display will prompt “Enter Key Press 1.” If the correct input was selected, the user will be shown a congratulations string with a custom character of a person (aka Simon) which will continue back to the main menu. This process will be repeated for different levels as they progress.

The 8x8 LED matrix was used specifically to light up the 4 different corners of the screen so that the user can use the buttons to select which corners of the led matrix were displayed. Each corner consisted of 4x4 LED’s lit up while the rest of the board was off. If multiple boxes were displayed, the user would have two button presses required to account for each corner that lit up. I was also able to connect a power plug to the bread board so the game wouldn’t have to run off a computer but while plugged into a socket. This concludes my hardware included and description of their functionality.

While each hardware had their specific functionality, I still managed to incorporate build-upons for the project. The first being my 8x8 LED matrix and having the capability of configuring it to work properly to the game’s requirements. I was also able to enable and configure EEPROM to save the high score of the game when the board is un-plugged from a power source. I also incorporated customer characters into the main menu and congratulations display on my LCD when the user passes the level. I was also able to incorporate a soft reset during the game play. Lastly, I was able to make each level a bit different. Level 1 is the basic, 1 blink level. Level 2 has 2 blinks. Level 3 had a blink for each button but increased the period per blink. Level 4 had multiple blinks with an increase in the period. Level 5 was similar to level 4 but incorporated multiple blinks per period which required the user to input multiple key presses at once to duplicate the blinks. Overall the Build-upons was the challenge for this overall project.

The testing period of the design and implementation was rigorous. Through every step of the project, there was multiple trail and errors to determine the correct way of implementing the project. For the 8x8 LED matrix, I spent two days determining how to organize the pins, power and ground to determine the correct LEDs I wanted to display. For the menu, I had to think of a way to incorporate multiple concurrent SM’s to work along each other. After trying to use one SM for the game portion of the design, I ended up making multiple synch SM’s for each level with a flag that would let them run, as well as a flag that would cause the menu screen to pause while the game is running in another SM. For EEPROM, I had to google and find online documentation on how to implement the EEPROM library that the avr program provides. Afterwards I was able to enable the feature. Although there are a few bugs I couldn’t fix. Since I have two flags running for the menu and current game level SM, when I press the last button that ends the game SM, the menu would also react to that button press as if I was in the menu SM. It doesn’t occur all the time but it will occur once in a while. Another bug is the period and the button presses, I didn’t notice it but if the buttons aren’t pressed precisely together when I have multiple button press inputs, the period will read it as once button press. To fix that, I have found a soft spot where the period will be able to wait and read for both button inputs.

Overall, I am very proud of my work and the time I spent the last 3 weeks to design, code, and implement it. It was a great learning experience and since this is the field I want to work in after college, I plan to make many more projects as I take more embedded system classes.