ECEN 2310 Final Project Report

Stefan Stroessner and David Wade

For our final project, we re-created the classic childhood game “Whack-a-Mole.” In this project, we wanted to not only display the capabilities of utilizing MATLAB to program a microcontroller, but also create a fun and interactive project that was enjoyable to work on. We used an Arduino microcontroller, breadboard, three buttons, sound buzzer, potentiometer, and four light emitting diodes to create the gaming platform. In short, the game is single player with the player’s objective being to hit the correct button to the corresponding lit up LED. If the correct button is pushed, a distinct sound is played from the buzzer and a point is added to the player’s score, however a point is deducted when the incorrect button is pushed. When incorrect, the red LED will light up along with another distinct sound from the buzzer. After the player’s choice, another LED is randomly selected to light up, continuing until the player reaches a score of ten. In order to properly code the microcontroller, we had to install an Arduino support package for MATLAB that included many useful functions for communicating with the Arduino. This add on along with many others is a clear display of MATLAB’s adverse range of utility. The majority of the code for this game lies within a single while loop that will continue to loop until the player reaches a score of ten. A variable called “hit” is initialized to zero and begins the loop since “hit” is equal to zero. All of the LEDs are set to be turned off, then using the randi function, MATLAB generates a random integer between one and three, indicating which LED will be turned on for the player. The chosen LED is saved through a variable and the value of the hit variable is then set equal to one. Following the LED choice, the while loop is then split into three separate if statements, each depending on which LED has been chosen by the randi function. For example, if the first LED is chosen, then the code runs through to check which button the player has pressed in response. If the player presses the correct corresponding button to the lit up LED, then the musical note A is played and one point is awarded to the player's score. If the incorrect button is pressed, then the red LED is lit up and a very high frequency sound is played in the buzzer. A point is then subtracted from the player's score, which is also displayed on the command console after each round of the game. After a point is either added or subtracted, the value of "hit" is then set back to zero which will cause the while loop to perform another iteration. This process again will continue forever until the player reaches the maximum score of ten points. Once the maximum score is reached, then the code will exit the while loop, followed by the simultaneous lighting of all four LEDs. We were able to program the game to run exactly how we desired, however there are definitely still improvements we can make in order to condense the code. For example, the three sections of the while loop can likely be turned into a single called function. This is due to them being identical in nature with the exception of different pin numbers for the conditionals. In addition we can likely reduce the b variables and the speaker status variables to a single variable to avoid redundancy in variable assignments. Overall, we were able to complete our goal of utilizing MATLAB to design an interactive project and show how MATLAB can be used as a tool for programming an electronic device.