

Whac-A-Mole

We started the project by building the basic circuit provided alongside the starter code. The first upgrade implemented was elaborating the interrupt triggered when the button is pressed which allowed us to check that the button press coincided with one of the three green/red LEDS lighting up checking if a point has been scored, to light up the white LED in response. Adding and tracking a score to the game was just a matter of initializing a variable at the start of the code and increasing this variable every time the button press interrupt was called since that meant a point was scored.

To flash all LED lights once the score reached ten, we implemented a condition that splits the main void loop of the Arduino into two states: a “game state” that randomly flashes one of the three LEDS while checking for correct button presses; and a “winning state” responsible for flashing all the LEDS once the player has won. Therefore, at the start of each loop the score of the player is compared to 10 and then the appropriate state’s code is executed

The next step was to develop two-player mode, by recreating the same circuit on the opposite half of the breadboard, programming wise this meant implementing a second interrupt that monitors the button corresponding to player 2 and have the conditional at the state of the loop check if either of the players have sent the circuit into the “winning state”. To further the two player functionalities, we added a servo that compares the score of players one and two and then point to the side of the currently winning player or point to the middle if the score is tied.

We then added another two extra features a difficulty dial and a game reset button. The difficulty dial was done with a potentiometer, the analog values measured with it were then mapped to a delay time for the LEDS using the map function to allow the players to choose how fast the game goes. The game reset button is used to start the game again once either player has won; the button is checked in the loop of the “winning state”, if it is read as HIGH both player’s scores will be reset and a variable which works as another option to enter the “game state” instead of the “winning state” will be set to true, effectively resetting the game. An upgrade to this feature we considered implementing was a led for each player that kept count of how many wins each player has, however we didn’t build it since we had run out of pins to control said LEDS. A possible solution to this is issue would have been to use shift registers.

