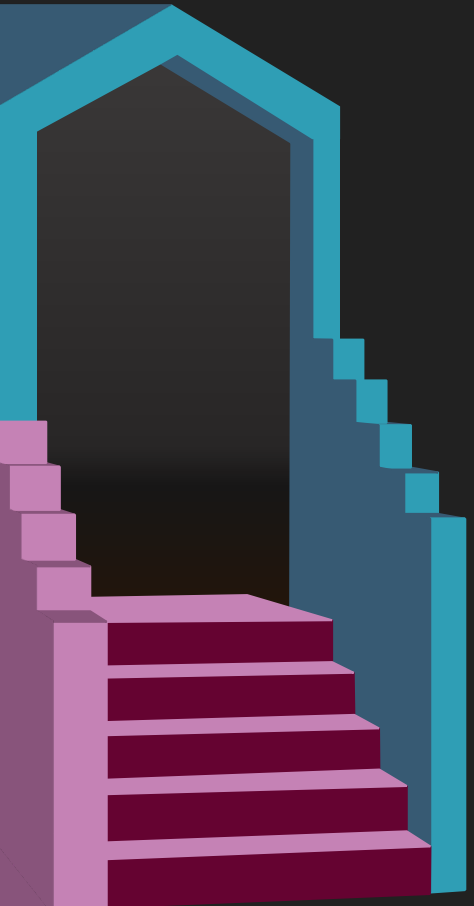
Red Light, Green Light

Squid Game Squad 174

Team Members: Shamal Kumar, Vigomar Kim Algador,

Sam Ho, Dimetree Aburto

CONTENTS



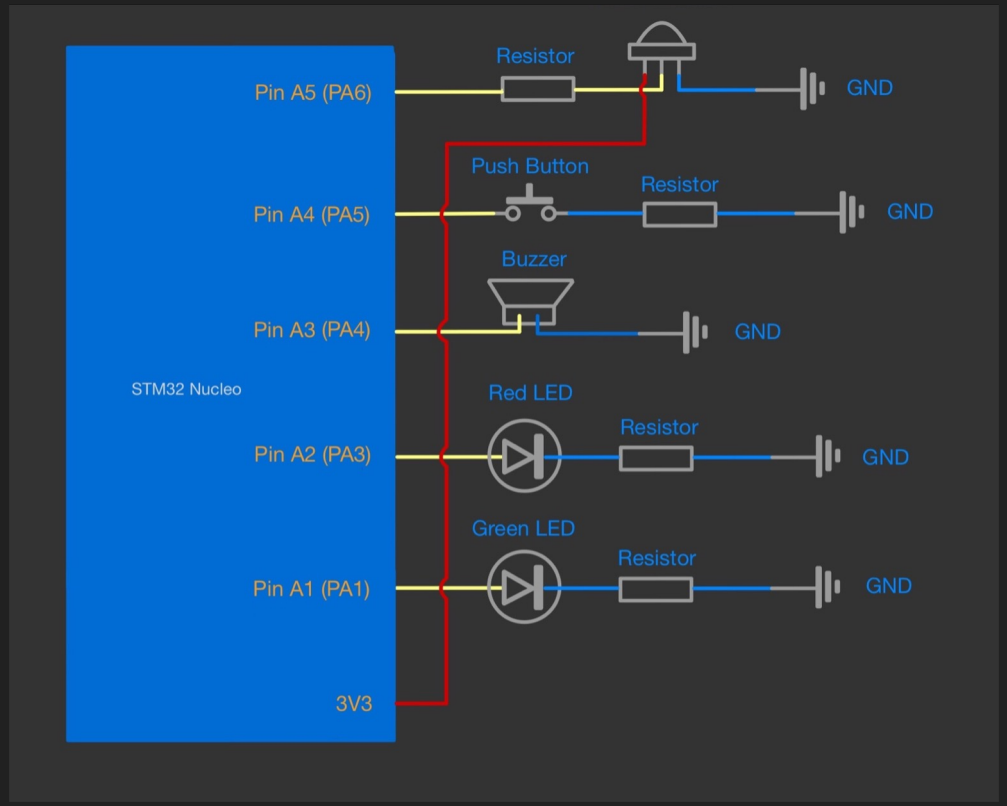
DESCRIPTION

The project imitates “Red Light, Green Light” a minigame based off the show “ Squid Games”, but our game will only be intended for a single player at a time. In this lab, we will be using a STM board, motion sensor, green/red LEDs, and a buzzer. There will be a green LED indicating that movement is allowed and a red LED indicating that movement is not allowed. When the red LED is active and its code is executed, the motion sensor activates and detects any movement. If the motion sensor senses movement, the output buzzer will sound off. After a certain time, the green LED will light up indicating the removal of the player was successful and a new play may start.

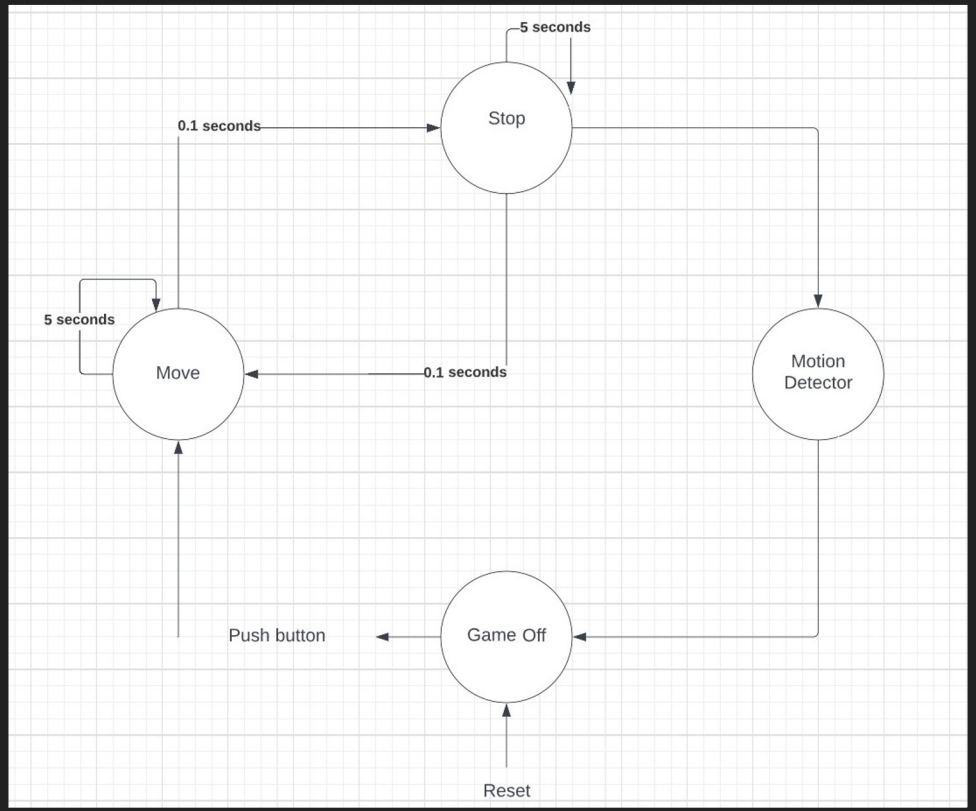
MATERIALS

- STM32 microcontroller
- Breadboard
- Red and Green LED
- AM312 IR Motion sensor (5 meter detection)
- ADA1536 5V Buzzer
- Tactile push button
- Jumper cables
- 1K Ω Resistor (1x)
- 470 Ω Resistor (1x)
- 300 Ω Resistor (2x)

SCHEMATIC


Pin A5 (PA6)
Pin A4 (PA5)
Pin A3 (PA4)
STM32 Nucleo
Pin A2 (PA3)
Pin A1 (PA1)
3V3
Resistor
GND
Push Button
Resistor
GND
Buzzer
GND
Red LED
Resistor
GND
Green LED
Resistor
GND

FSM DIAGRAM

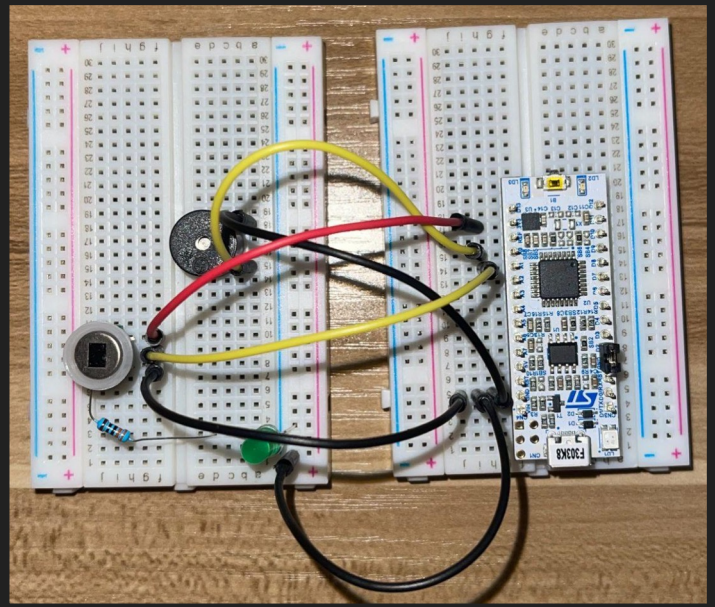

5 seconds
Stop
0.1 seconds
5 seconds
Move
0.1 seconds
Motion Detector
Push button
Game Off
Reset

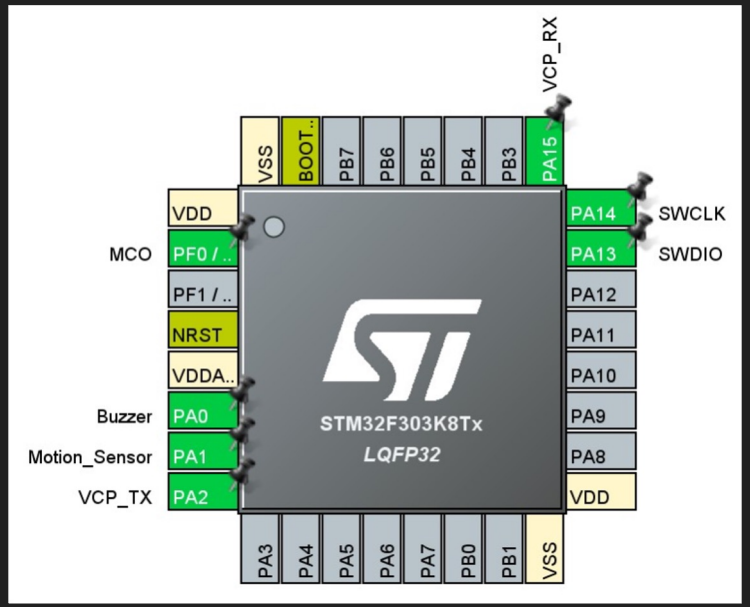
FSM TABLE

Current State	Input	Next State	Output
Move	Time delay 5 secs	Move	Green_LED = 1 Red_LED = 0
	Time delay 0.1 secs	Stop	
Stop	Time delay 5 secs	Stop	Green_LED = 0 Red_LED = 1
	Time delay 0.1 secs	Motion Detection (MD)	
Motion Detection (MD)	Motion Detector (Pin A6)	Motion Detection (MD)	Buzzer sounds off
		Game Off (GO)	
Game Off (GO)	Push Button (Pin A5)	Game off	Green_LED = 0 Red_LED = 0
		Move	

Test Case: Buzzer/LED

Main breadboard test case:
https://youtube.com/shorts/SxclQrjSY0c?feature=share

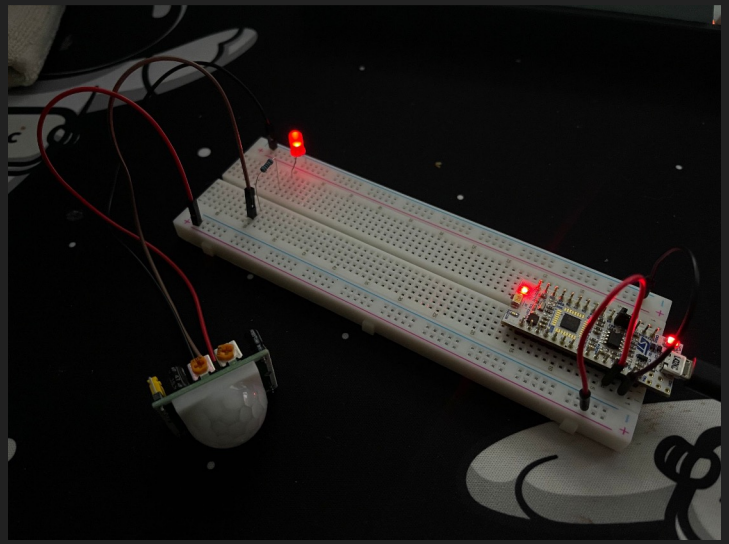


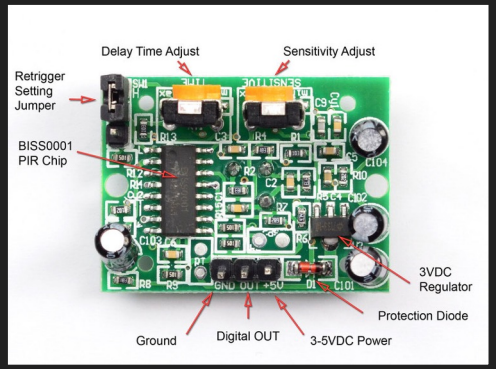


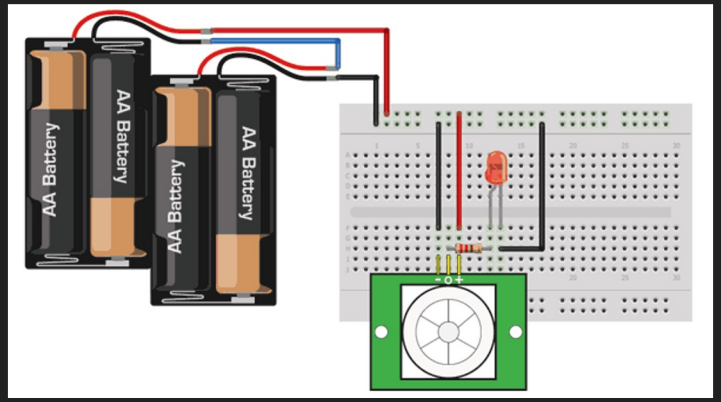
Test Case: PIR Motion Sensor/LED

- Functionality
- Sensitivity

https://www.youtube.com/shorts/rD1VaD2fAQE







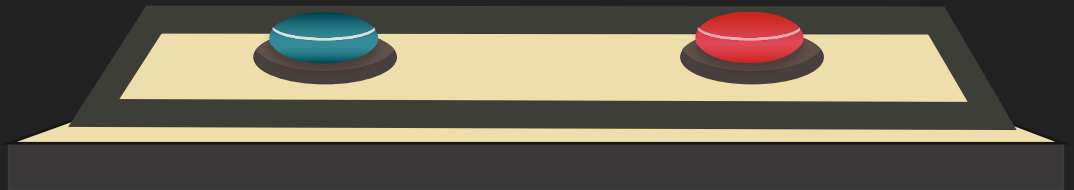
SOURCE: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor

Test Case 3 : Debug

```c
typedef enum
{
    Move,
    Stop,
    MD,
    GO
} eSystemState;

 //Green A1, Red A3, Sensor A6, Buzzer A4, Button A5
eSystemState GreenLight(void) {
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
    HAL_Delay(5000);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
    HAL_Delay(100);
    return Stop;
}
eSystemState RedLight(void) {
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
    HAL_Delay(5000);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
    HAL_Delay(100);
    return Move;
 }
eSystemState Motion(void) {
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
    HAL_Delay(3000);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
    HAL_Delay(1000);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
    HAL_Delay(3000);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
    HAL_Delay(1000);
    return GO;
 }
eSystemState Finish(void) {
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
    HAL_Delay(100);
    return GO;
```

```c
switch(eNextState)
  {
    case Move:
        eNextState = GreenLight();
        break;
    case Stop:
        eNextState = RedLight();
        if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_6))
        {
            eNextState = MD;
        }
        else
        {
            eNextState = Move;
        }
        break;
    case MD:
        eNextState = Motion();
        break;
    case GO:
    eNextState = Finish();
    if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_5))
    {
        eNextState = Move;
    }
    else
    {
        eNextState = GO;
    }
    break;
  }
```



Conclusion

With the knowledge learned from our test cases and the drawn circuit schematic, we were able to successfully have our board run our code. It was a good experience to collaborate as a team to further deepen our knowledge of what microcontrollers can do. By applying the concepts we learned in Lab 2, we were able to get a single player game of “Red Light, Green Light” to work as we intended.