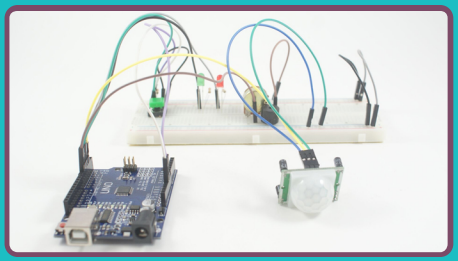
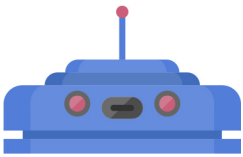
INFRARED SECURITY SYSTEM



Creation Crate

BUILDING THE MAKERS OF TOMORROW

INDEX



INTRODUCTION

Welcome to Month 13!

What are we creating?

Detect intruders coming into your room!
Introducing this month's build: **Infrared Security System!**

With this project, you will learn how to use a passive infrared (PIR) sensor to sound an alarm when somebody walks by.

How do we make it?

In two steps:

1. Build the hardware:

We will attach the output of the PIR to an interrupt pin of the Uno R3 to tell the code when there is an intruder.

2. Programming it:

The program will keep track of whether or not the system is armed. It will only sound the alarm when it is armed.

Support Page

https://mycreationcrate.com/month-13

GHJE78

PART LIST

X1

UNO R3

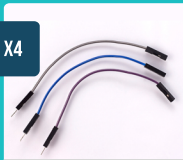
X1

USB CABLE

X1

BREADBOARD

X4



MALE TO FEMALE JUMERS

20cm X 8

10cm X 4

MALE TO MALE JUMPERS

X1

PIR SENSOR

X1

BUTTON

X1

BUTTON CAP

X1

GREEN LED

PART LIST

X1

RED LED

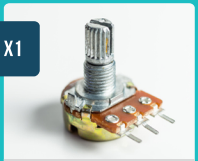
X1

10K OHM RESISTOR

X2

220 OHM RESISTOR

X1



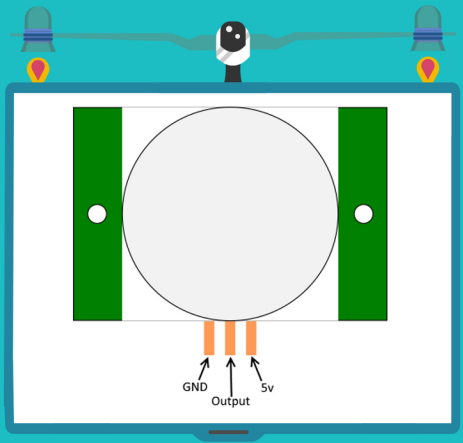
POTENTIOMETER

X1

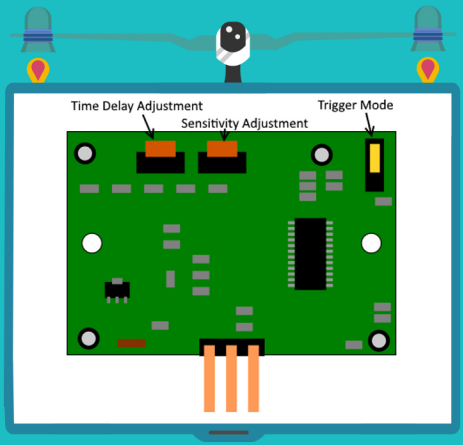
BUZZER

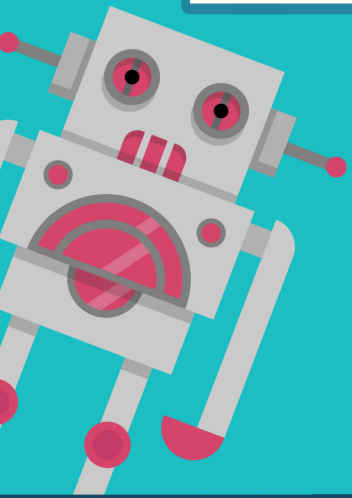
PIR SENSOR

This project utilizes a PIR sensor. PIR stands for Passive InfraRed. Infrared is a type of light outside of the visible light spectrum. This means you cannot see it with your eyes. However, this sensor is built to be able to detect differences in infrared light. Anything that emits heat emits infrared light, even humans! When the PIR sensor detects a change in infrared light where it is pointed, it will turn on the output pin. Below are the diagrams showing the pinout and the various adjustment options. We won't need to change the Time Delay Adjustment or the Trigger Mode, but feel free to play with the Sensitivity Adjustment.



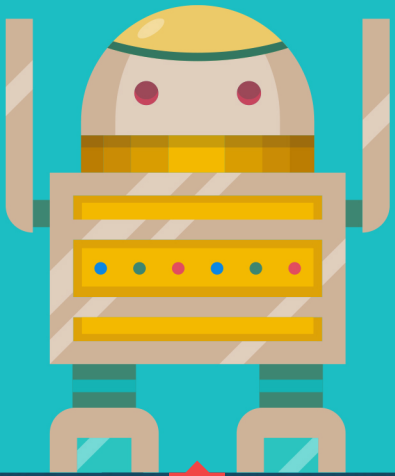
PIR SENSOR


Time Delay Adjustment
Sensitivity Adjustment
Trigger Mode



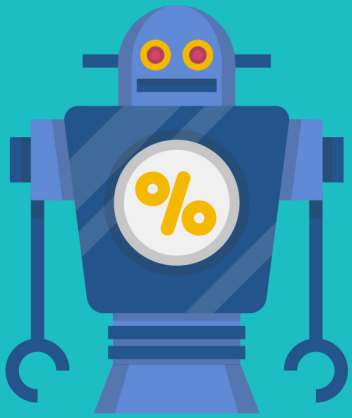
ACCURACY

- Because the PIR sensor is triggered by heat, sometimes it will trigger without actually sensing an intruder. Things like sunlight or even a draft of air could set the sensor off. This is called a false positive, and occurs because the sensor thinks it has sensed something when in reality it didn't. However, there are a few ways of making the sensor more reliable. First, you can adjust the sensitivity potentiometer (shown in the PIR sensor diagram: clockwise is more sensitive, counterclockwise is less sensitive). We'll go into another solution in the Exercises section.



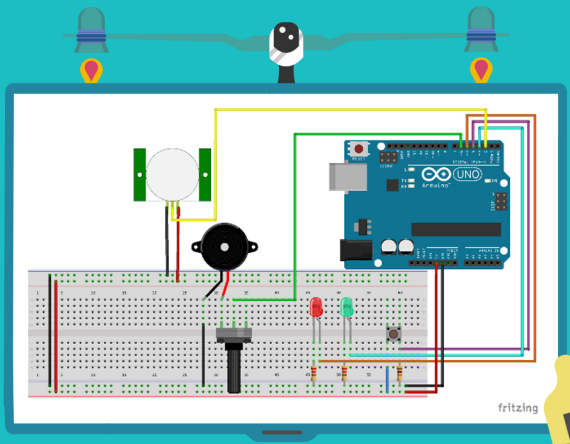
THE MODULO OPERATOR

- In the code for this project, we will be using the modulo operator. In most computer languages, this is a % sign. Modulo returns the remainder of a number divided by another number. For instance, 5 % 3 equals 2, because 5 goes into 3 once with 2 left over. In the code we use millis() % 1000, which will give us a number between 0 and 999, depending on the remainder after the number millis() returns is divided by 1000. We now have a number that counts up from 0 to 999, and then resets back to 0. Using this, we will play one note when the remainder is less than 500 and a different note when this is greater than 500.

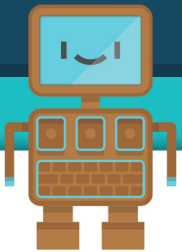


HARDWARE

- Connect the hardware as shown.

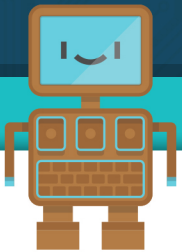


PROGRAM



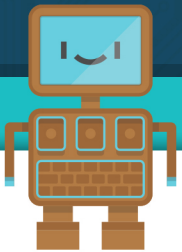
```
//Month 13: Infrared Security System

//Pin definitions
int infraredPin = 0;
int readyLED = 3;
int buttonPin = 4;
int armedLED = 5;
int buzzerPin = 6;

//Security system variables
long readyTime = 0; //Time button was last pressed
boolean readyToArm = false;
boolean armed = false;
boolean buzzer = false;
boolean buttonPushed = false;

void setup()
{
  Serial.begin(9600);
  attachInterrupt(infraredPin, detection, RISING); //Sets up
the PIR pin as an interrupt, runs detection()

  pinMode(readyLED, OUTPUT);
  pinMode(armedLED, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
}
```

PROGRAM



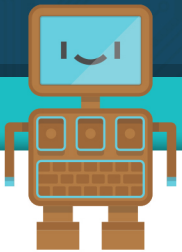
```
void detection() //Runs whenever the PIR
sensor detects movement
{
  Serial.println("Warning! Infrared Sensor detected
movement!");
  if(armed) //Only sounds buzzer if the system is armed
  {
    buzzer = true;
  }
}
void loop()
{
  if(digitalRead(buttonPin) && !buttonPushed) //If the button
is pushed
  {
    buttonPushed = true; //This is used so the "ready" state
is only toggled once while the button is on
    buzzer = false; //Turn off buzzer if it is on
    armed = false;
    readyToArm = !readyToArm; //Toggles the "ready" state
    readyTime = millis(); //Stores the time the button was
pressed
  }
  else if(!digitalRead(buttonPin))
  {
    buttonPushed = false; //When the button is released, allow
it to be pressed again
  }
```

PROGRAM



```
if(readyToArm)
 {
   digitalWrite(readyLED, HIGH); //Turn on the "ready" LED
   if(millis() >= (readyTime + 5000)) //Wait 5 seconds
   {
     armed = true;
   }
 }
 else
 {
   digitalWrite(readyLED, LOW);
   armed = false;
 }

 if(armed)//Turn "armed" LED on or off
 {
   digitalWrite(armedLED, HIGH);
 }
 else
 {
   digitalWrite(armedLED, LOW);
 }

 if(buzzer) //Turn on the buzzer
 {
   if(millis()%1000 < 500) //Between 0-499 milliseconds
   {
     tone(buzzerPin, 880); //Plays an 'A'
   }
```

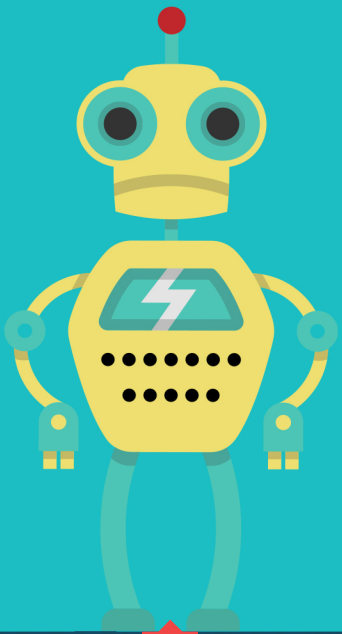
PROGRAM



```
else //Between 500-999 milliseconds
     {
       tone(buzzerPin, 659); //Plays an 'E'
     }
   }
   else
     noTone(buzzerPin); //Play no sound
}
```

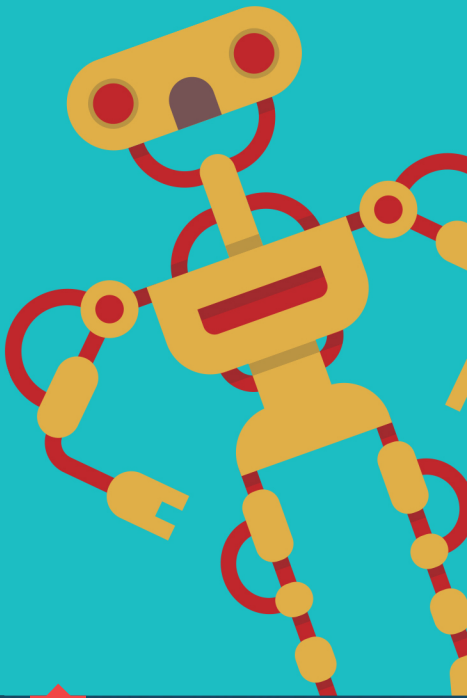
WHAT YOU SHOULD SEE

- Before arming the security system, you need to point the PIR sensor in the direction you wish to monitor motion. To arm the system, press the button. The green “ready to arm” LED should activate. After 5 seconds, the red LED will turn on, indicating that the system is armed. Once the PIR detects motion, the alarm will sound. Simply press the button to deactivate the alarm. If the alarm is too loud, turn the potentiometer to lower the volume.



MONTHLY CHALLENGE

- **Make the alarm pattern more than two notes! You can try random frequencies, or if you want to find the frequency of specific musical notes reference the frequency chart on the project's webpage.**



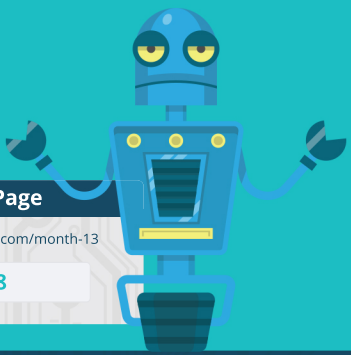
EXERCISES

Solve these problems and write the answers below.

1.) Put the time the system waits before arming into a variable and make it wait longer.
Answer:

2.) Have the LEDs flash when the alarm is sounding to give visual feedback.
Answer:

3.) Help prevent false positives by requiring the sensor to trigger twice in 15 seconds for the alarm to sound.
Answer:



Support Page

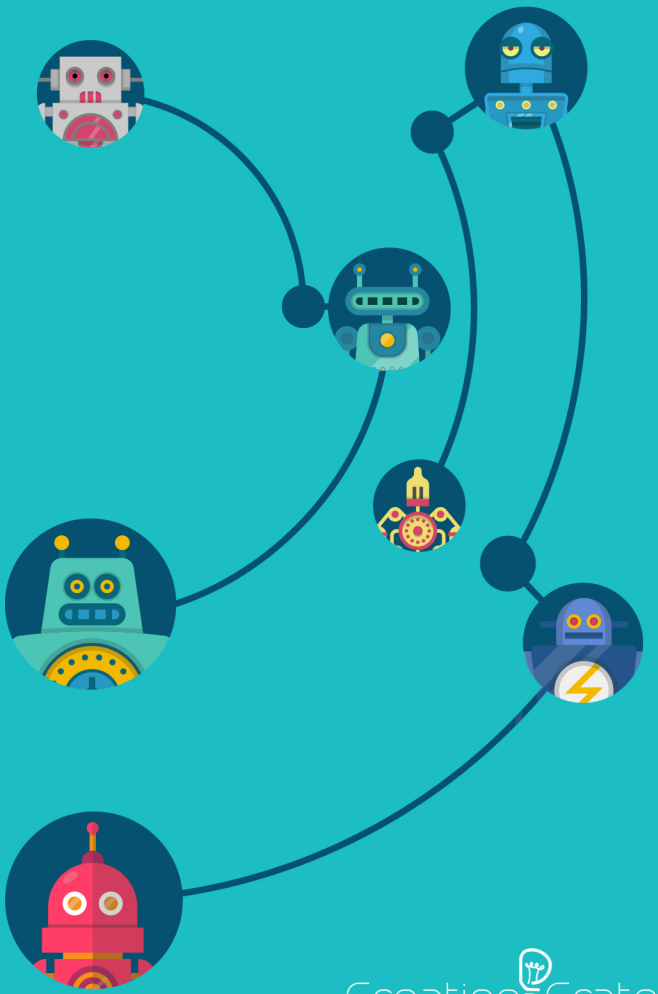
https://mycreationcrate.com/month-13

GHJE78

SNEAK PEEK

Here's a sneak peek at next month's project!

Next month's project will build upon this month's project, so make sure you keep all the parts together until next month! Can you guess what it is?



Creation Crate

BUILDING THE MAKERS OF TOMORROW