

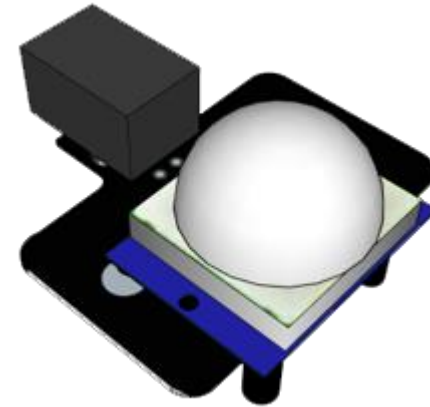
# TScratch Basics

## Coding with mBlock (Software)

# Learning Objective

In this lesson you will learn:

## TScratch (TSense PIR)



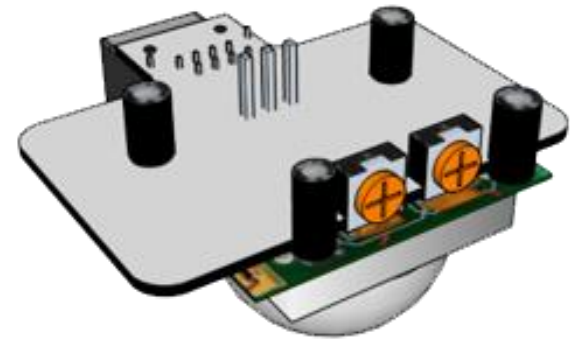
- Include a temperature-sensitive sensor to your project!
- Coding another Digital input with Arduino

# What is a PIR?

- PIR is also known as the Proximity InFra-red sensor.
- It is a “switch” that works on motion/InFra-red .
- COOL(InFra-red absent) → Low Resistance (not working)
- HOT(InFra-red present) → High Resistance (working)
- They are used commonly in security system, door chimes, etc.
- What is InFra-red?  
A spectrum of light that is not visible by naked eyes and commonly used with to detect heat signatures.  
(i.e. temperature sensing)

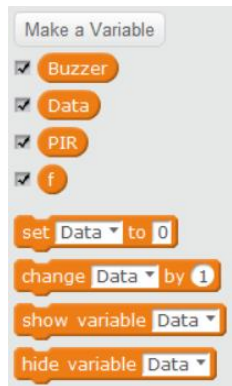
# What is a PIR?

- The PIR should be tuned with the dome facing **DOWN**.
- Check that the distance and delay time knobs (orange) are tuned to the **LEFT**.
- Flat side of the orange knobs should be pointing in the **southeast** direction.



# Write your PIR control program

- In this program, we will have a PIR **digital input (port 2)** and buzzer **digital output (port 13)**. We will use a variable **f** (frequency) to set the notes for the buzzer, and variable **Data** to track the value of PIR.
- Create the respective variables in the Data&Blocks tab, then define them in the scripting area:

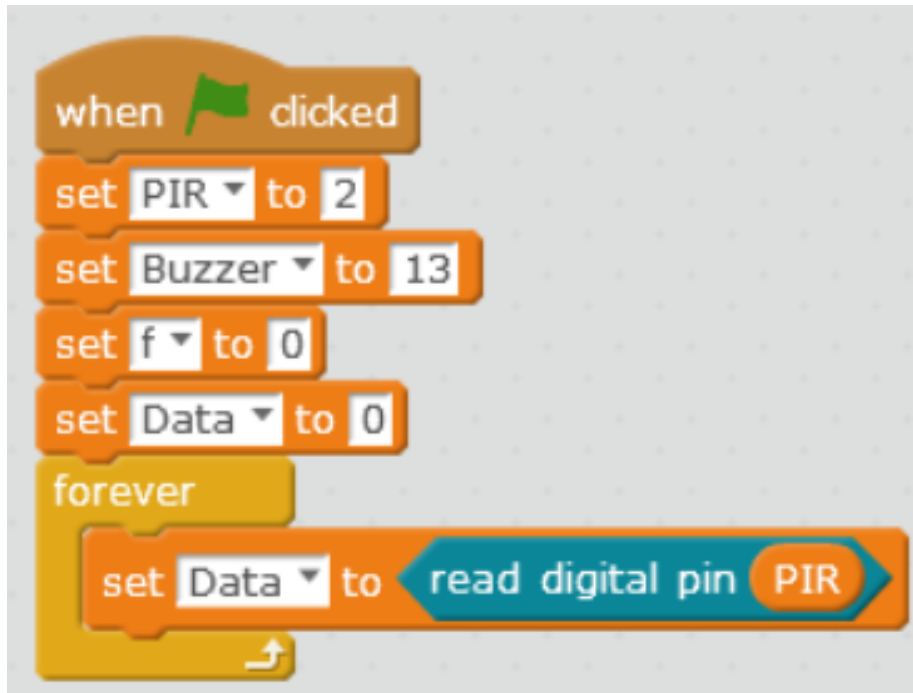


# Write your PIR control program

- Within a forever loop, set your conditions.
- When the PIR sensor detects heat, play tone “f” on buzzer at beat 50, increasing **f** by 35 until it is more than 15000.
- When the PIR sensor doesn’t detect any heat, do nothing.
- PIR sensor detects → PIR = 1 (HIGH)
- Set buzzer play tone “f” on buzzer at beat 50, increasing **f** by 50 until it is more than 15000
- PIR sensor doesn’t detect → PIR = 0 (LOW)
- Buzzer does nothing

# Write your PIR Program

- Write a simple program to track and display the value of the PIR sensor using Data.



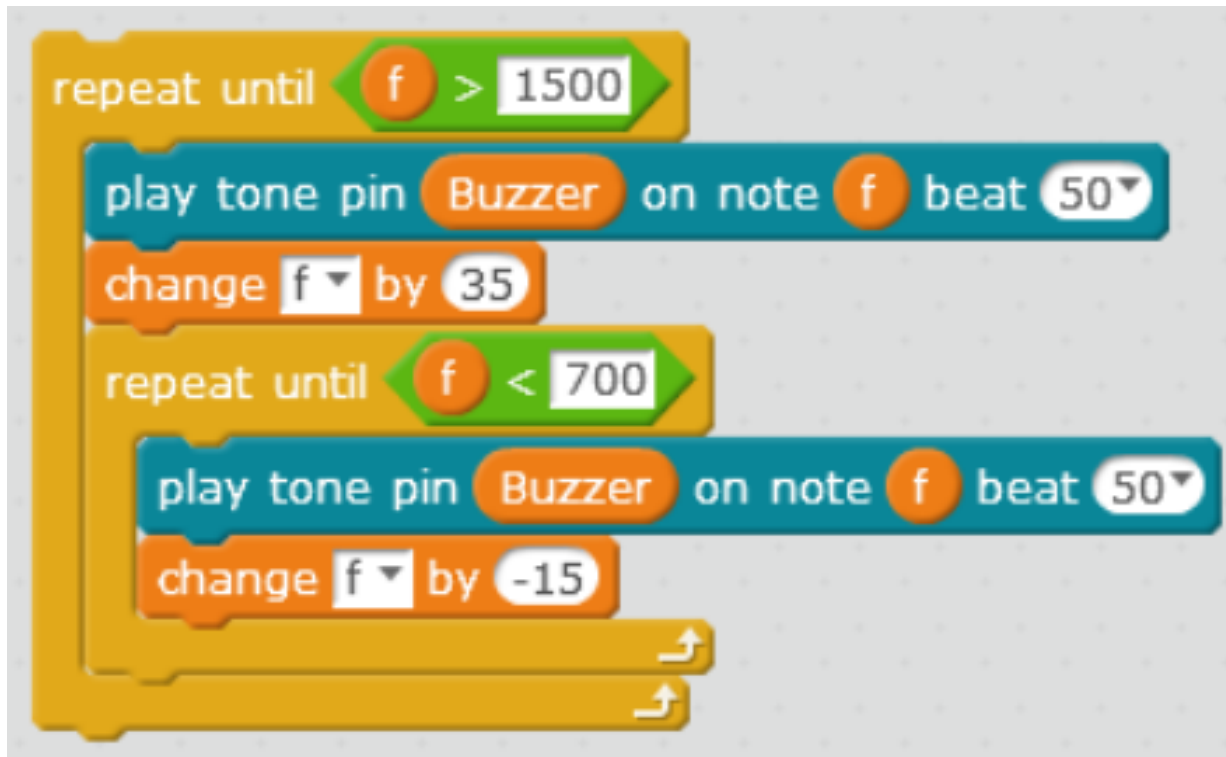
```
when green flag clicked
  set PIR to 2
  set Buzzer to 13
  set f to 0
  set Data to 0
  forever loop
    set Data to read digital pin PIR
```



Note: By putting Data in a forever loop, the value of Data will constantly be updated

# Write your PIR Program

- The alarm sound is created using the following series of tones:



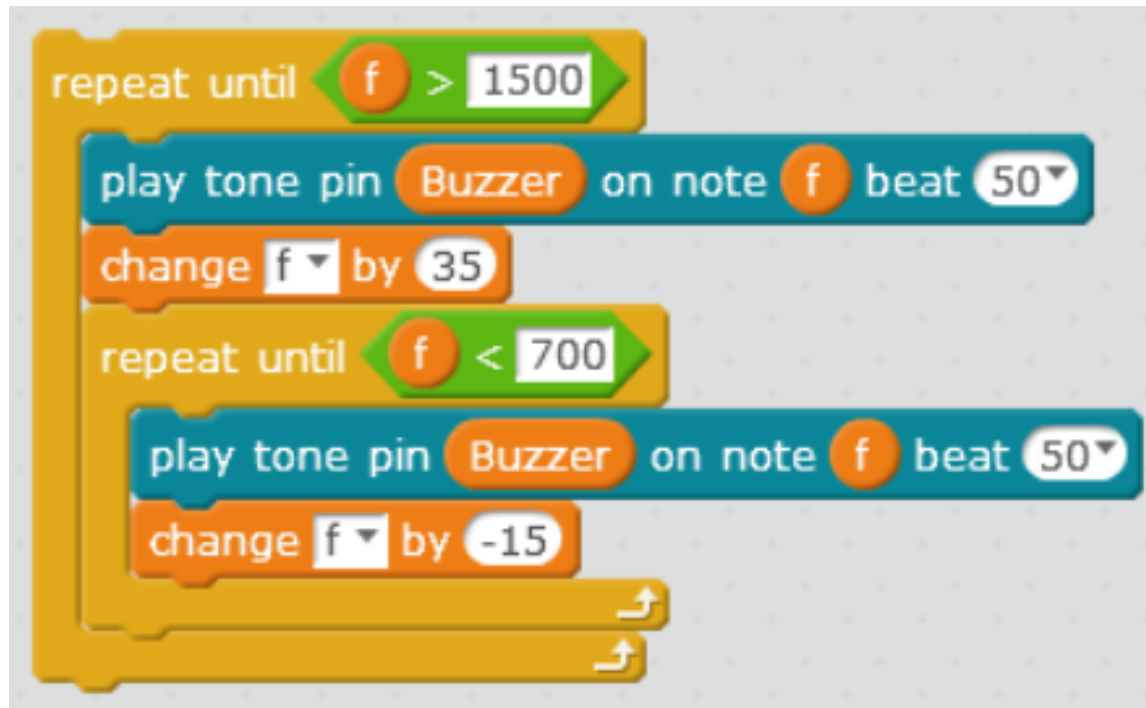
```
repeat until f > 1500
  play tone pin Buzzer on note f beat 50
  change f by 35
  repeat until f < 700
    play tone pin Buzzer on note f beat 50
    change f by -15
```

The image shows a Scratch script for generating an alarm sound. It consists of a large yellow 'repeat until' loop with a condition 'f > 1500'. Inside this loop, there are three blocks: a blue 'play tone' block with pin 'Buzzer', note 'f', and beat '50'; an orange 'change f by 35' block; and a smaller yellow 'repeat until' loop with condition 'f < 700'. Inside this inner loop, there are two blocks: a blue 'play tone' block with pin 'Buzzer', note 'f', and beat '50'; and an orange 'change f by -15' block. The script ends with two upward-pointing arrows indicating the end of the code.



# Write your PIR Program

- The alarm sound is created using the following series of tones:



```
repeat until f > 1500
  play tone pin Buzzer on note f beat 50
  change f by 35
repeat until f < 700
  play tone pin Buzzer on note f beat 50
  change f by -15
```

Note: to change the beat to 50, you have to click on the beat and manually type in 50

# Write your PIR Program

- First, PIR sensor detects → Data = 1 (HIGH)
- Set the buzzer to play the alarm tone.
- Check if the PIR sensor is sensing any presence with the **if** block

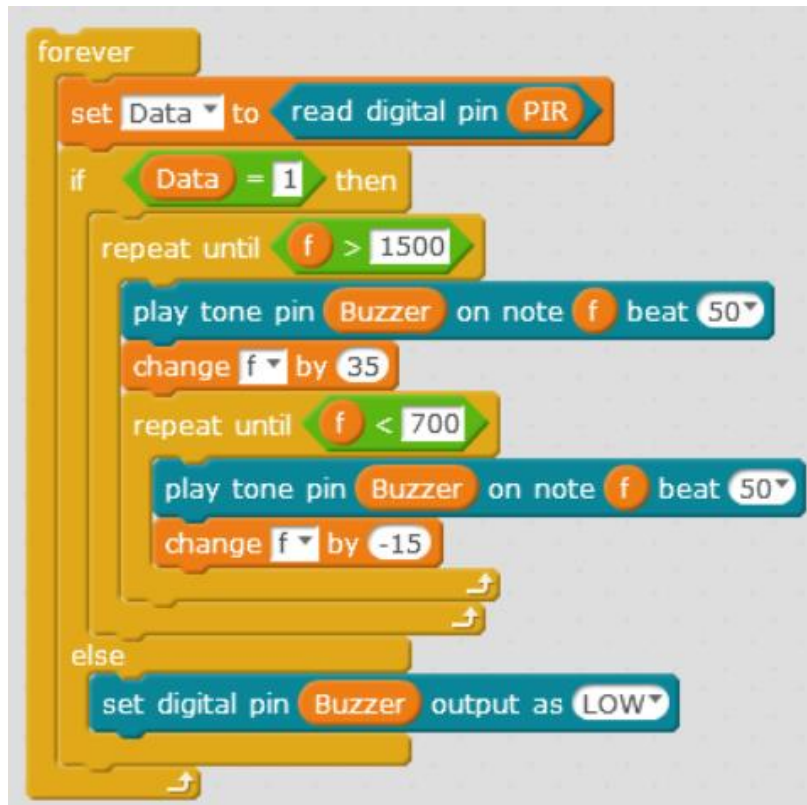
```

forever
  set Data to read digital pin PIR
  if Data = 1 then
    repeat until f > 1500
      play tone pin Buzzer on note f beat 50
      change f by 35
    repeat until f < 700
      play tone pin Buzzer on note f beat 50
      change f by -15
  else
  
```

The image shows a Scratch script for a PIR sensor alarm. It starts with a 'forever' loop. Inside the loop, the first block is 'set Data to read digital pin PIR'. This is followed by an 'if Data = 1 then' block. Inside the 'if' block, there are two 'repeat until' loops. The first 'repeat until' loop has the condition 'f > 1500' and contains two blocks: 'play tone pin Buzzer on note f beat 50' and 'change f by 35'. The second 'repeat until' loop has the condition 'f < 700' and contains two blocks: 'play tone pin Buzzer on note f beat 50' and 'change f by -15'. After the 'if' block, there is an 'else' block which is currently empty. The script ends with a 'forever' loop arrow.

# Write your PIR Program

- Second, PIR sensor doesn't detect → Data = 0 (LOW)
- Switch the buzzer OFF.



```
forever
  set Data to read digital pin PIR
  if Data = 1 then
    repeat until f > 1500
      play tone pin Buzzer on note f beat 50
      change f by 35
    repeat until f < 700
      play tone pin Buzzer on note f beat 50
      change f by -15
  else
    set digital pin Buzzer output as LOW
```

The image shows a Scratch script for a PIR sensor program. It starts with a 'forever' loop. Inside the loop, the first block is 'set Data to read digital pin PIR'. This is followed by an 'if Data = 1 then' block. Inside the 'if' block, there are two 'repeat until' loops. The first 'repeat until' loop has the condition 'f > 1500' and contains two blocks: 'play tone pin Buzzer on note f beat 50' and 'change f by 35'. The second 'repeat until' loop has the condition 'f < 700' and contains two blocks: 'play tone pin Buzzer on note f beat 50' and 'change f by -15'. After the 'if' block, there is an 'else' block with the block 'set digital pin Buzzer output as LOW'. The 'forever' loop ends with a return arrow.

# Write your PIR Program

- Complete program

Note: You can use a cloth or cardboard box to shield the PIR sensor from any surrounding heat sources. Use Data to monitor the PIR detection status (Data = 1 means the sensor is detecting heat).

```

when clicked
  set PIR to 2
  set Buzzer to 13
  set f to 0
  set Data to 0
  forever
    set Data to read digital pin PIR
    if Data = 1 then
      repeat until f > 1500
        play tone pin Buzzer on note f beat 50
        change f by 35
      repeat until f < 700
        play tone pin Buzzer on note f beat 50
        change f by -15
    else
      set digital pin Buzzer output as LOW
  
```