CamJam EduKit Sensors Worksheet Six

Project	Intruder Alarm
Description	In this project, you will the passive inferred sensor circuit to include lights and sound.

Equipment Required

□ Raspberry Pi & SD card

□ Keyboard & Mouse

□ Monitor & HDMI Cable

□ Power supply

□ 400 Point Breadboard

□ Passive Infrared Sensor

□ 1 x Red LED

□ 1 x Blue LED

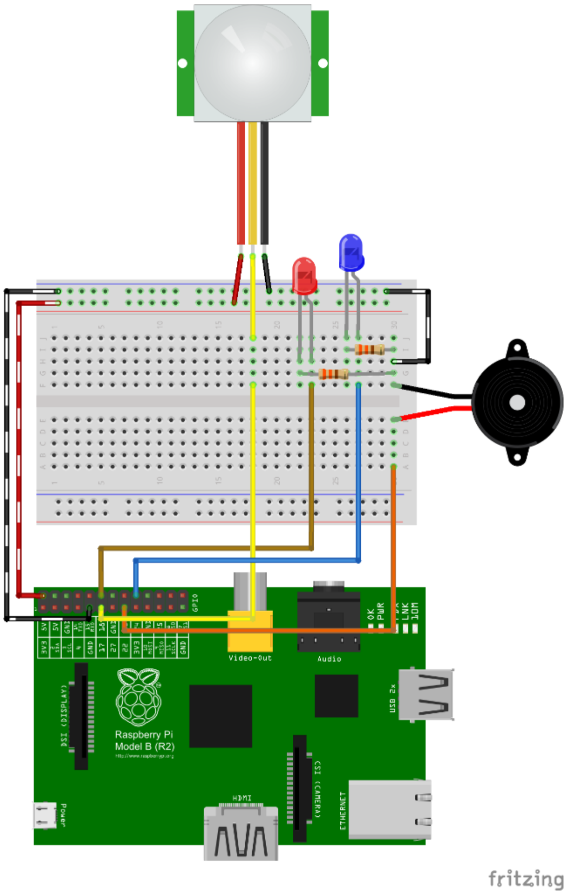
□ 2 x 330 Ω resistors

□ 9 x M/F jumper wires

□ 1 x M/M jumper wires

□ 1 x Buzzer

Building the Circuit



The Alarm circuit combines the PIR circuit and the LED/Buzzer circuit from Worksheet Two. You will be using this to create a simple movement alarm.

Build the circuit as shown.

Code

Follow the instructions in Worksheet One to turn on your Pi and open the terminal window. The code is based on the code in Worksheet Five, you are going to copy that code instead of retyping it all.

In the terminal window:

1. Change directory to the directory you created in Worksheet One using:

   ```
   cd ~/EduKitSensors/
   ```

2. Copy the code from the previous worksheet:

   ```
   cp 5-PIR.py 6-Alarm.py
   ```

3. Edit the file "6-Alarm.py" by typing the following:

   ```
   nano 6-Alarm.py
   ```

4. Edit the code so that it looks like the following.

   ```
   # Import Python header files
   import RPi.GPIO as GPIO
   import time

   # Set the GPIO naming convention
   GPIO.setmode(GPIO.BCM)
   GPIO.setwarnings(False)

   PinPIR = 17
   PinRedLED = 18
   PinBlueLED = 24
   PinBuzzer = 22

   print "PIR Module Test (CTRL-C to exit)"

   # Set pins as input/output
   GPIO.setup(PinPIR, GPIO.IN)
   GPIO.setup(PinRedLED, GPIO.OUT)
   GPIO.setup(PinBlueLED, GPIO.OUT)
   GPIO.setup(PinBuzzer, GPIO.OUT)

   # Variables to hold the current and last states
   Current_State  = 0
   Previous_State = 0

   try:
     print "Waiting for PIR to settle ..."
     # Loop until PIR output is 0
     while GPIO.input(PinPIR)==1:
       Current_State  = 0

     print "  Ready"
     # Loop until users quits with CTRL-C
     while True :
       # Read PIR state
   ```

Code

```
    Current_State = GPIO.input(PinPIR)

    if Current_State==1 and Previous_State==0:
      # PIR is triggered
      print "  Motion detected!"
      # Flash lights and sound buzzer
      for x in range(0,3):
         GPIO.output(PinBuzzer, GPIO.HIGH)
         GPIO.output(PinRedLED, GPIO.HIGH)
         time.sleep(0.5)
         GPIO.output(PinRedLED, GPIO.LOW)
         GPIO.output(PinBlueLED, GPIO.HIGH)
         time.sleep(0.5)
         GPIO.output(PinBlueLED, GPIO.LOW)
         GPIO.output(PinBuzzer, GPIO.LOW)
         time.sleep(0.5)

      # Record previous state
      Previous_State=1

    elif Current_State==0 and Previous_State==1:
      # PIR has returned to ready state
      print "  Ready"
      Previous_State=0

    # Wait for 10 milliseconds
    time.sleep(0.01)

except KeyboardInterrupt:
  print "  Quit"
  # Reset GPIO settings
  GPIO.cleanup()
```

Once complete use “Ctrl + x” then “y” then “enter” to save the file.

Running the Code

To run the code, type the following into the terminal window:

```
sudo python 6-Alarm.py
```

If you find that the code does not run correctly there may be an error in the code you have typed. You can re-edit the code by using the nano editor, typing `nano 6-Alarm.py`.

When the PIR detects movement, it will flash the LEDs and sound the buzzer three times.

Challenge

Alter the circuit and code so that the alarm is only active when it is dark.