

## CamJam EduKit Sensors Worksheet Six

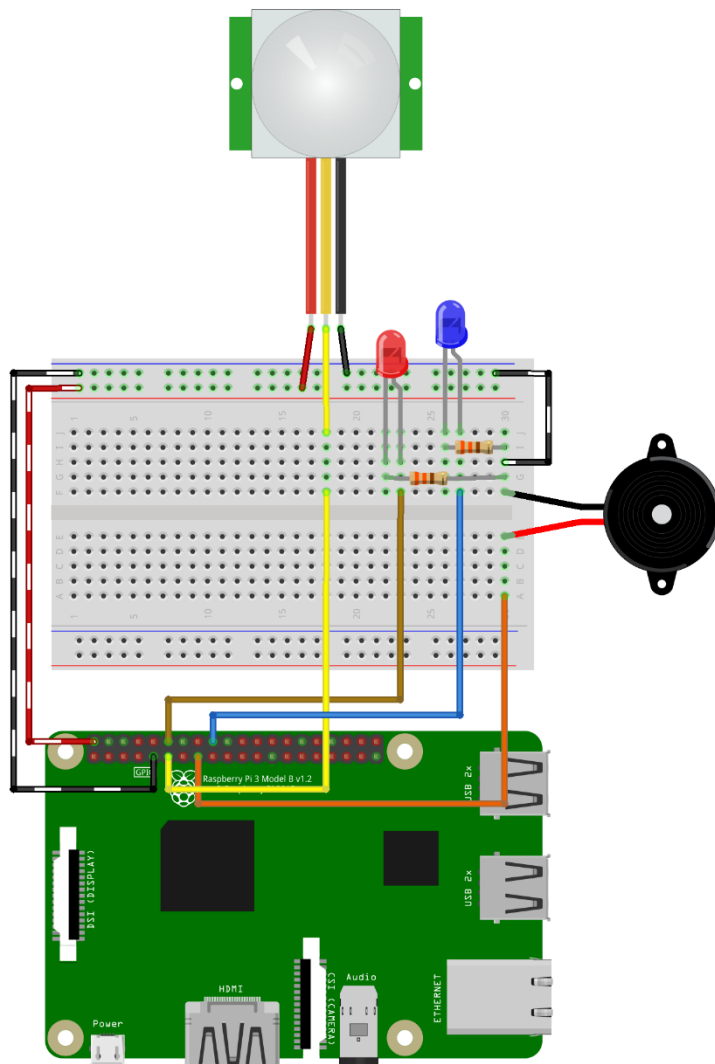
**Project** Intruder Alarm

**Description** In this project, you will use the passive infrared sensor circuit to include lights and sound.

### Equipment Required

- Your Raspberry Pi
- 400 Point Breadboard
- Passive Infrared Sensor
- 2 x 330  $\Omega$  resistors
- 9 x M/F jumper wires
- 1 x M/M jumper wires
- 1 x Red LED
- 1 x Blue LED
- 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.

fritzing

## Code

The code is based on the code in Worksheet Five, so you are going to copy that code instead of retyping it all. Start the IDLE3 editor and open the file 5-PIR.py from the EduKitSensors directory. Save it immediately as 6-Alarm.py.

Edit the code so that it looks like the following.

```
# CamJam Edukit 2 - Sensors
# Worksheet 6 - Alarm

# 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
        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()
```

Save the file as 6-Alarm.py.

## **Running the Code**

Select the Run Module menu option, under the Run menu item. Alternatively, you can just press the F5 key.

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, by using the LDR.