The background of the slide features a white desk lamp on the left, which is turned on, casting a bright yellow light. On the right, a hand is holding a smartphone with a yellow screen. The entire scene is set against a white background with a wooden floor at the bottom.

# IOT SMART LAMP PRESENTATION

BY ROBERT SOLOMON

GITHUB: [HTTPS://GITHUB.COM/EAZYROB97](https://github.com/EazyRob97)



**WHY  
SMARTLAMP!**

---







# WHY SMARTLAMP!

---

- CAN BE CONTROLLED FROM ANYWHERE



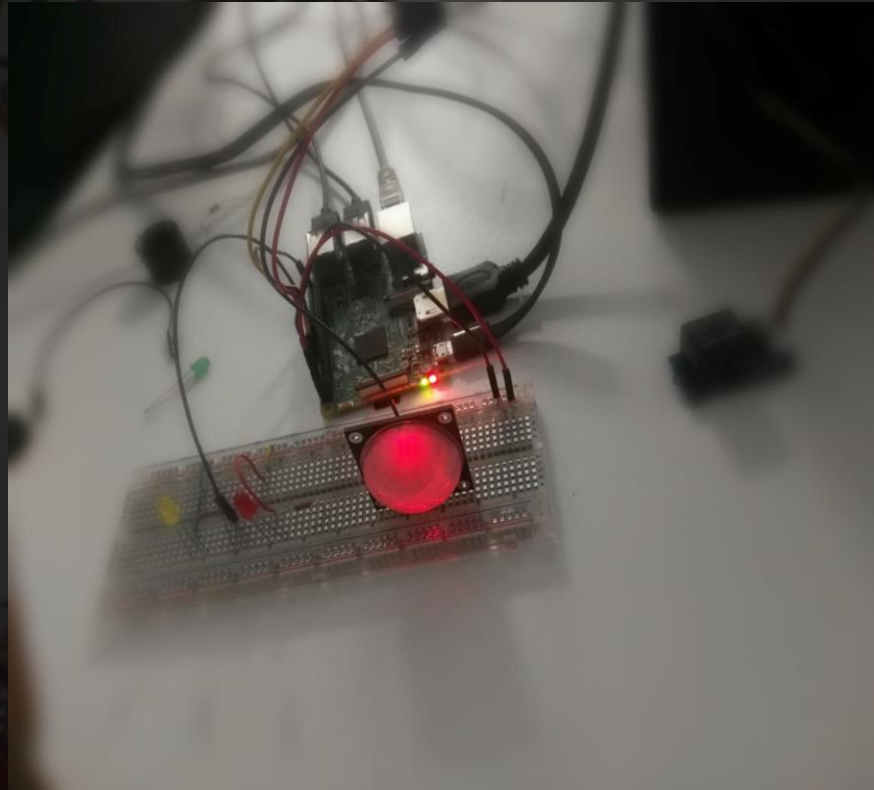
- 
- A photograph of a couple sleeping peacefully in a bed. The room is dimly lit, with a warm, yellowish light emanating from a smart light bulb on a bedside table. The couple is covered with a light-colored blanket. The overall mood is calm and cozy.
- SOME CAN HELP YOU SLEEP BETTER



# DESIGN/TECHNOLOGY USED!

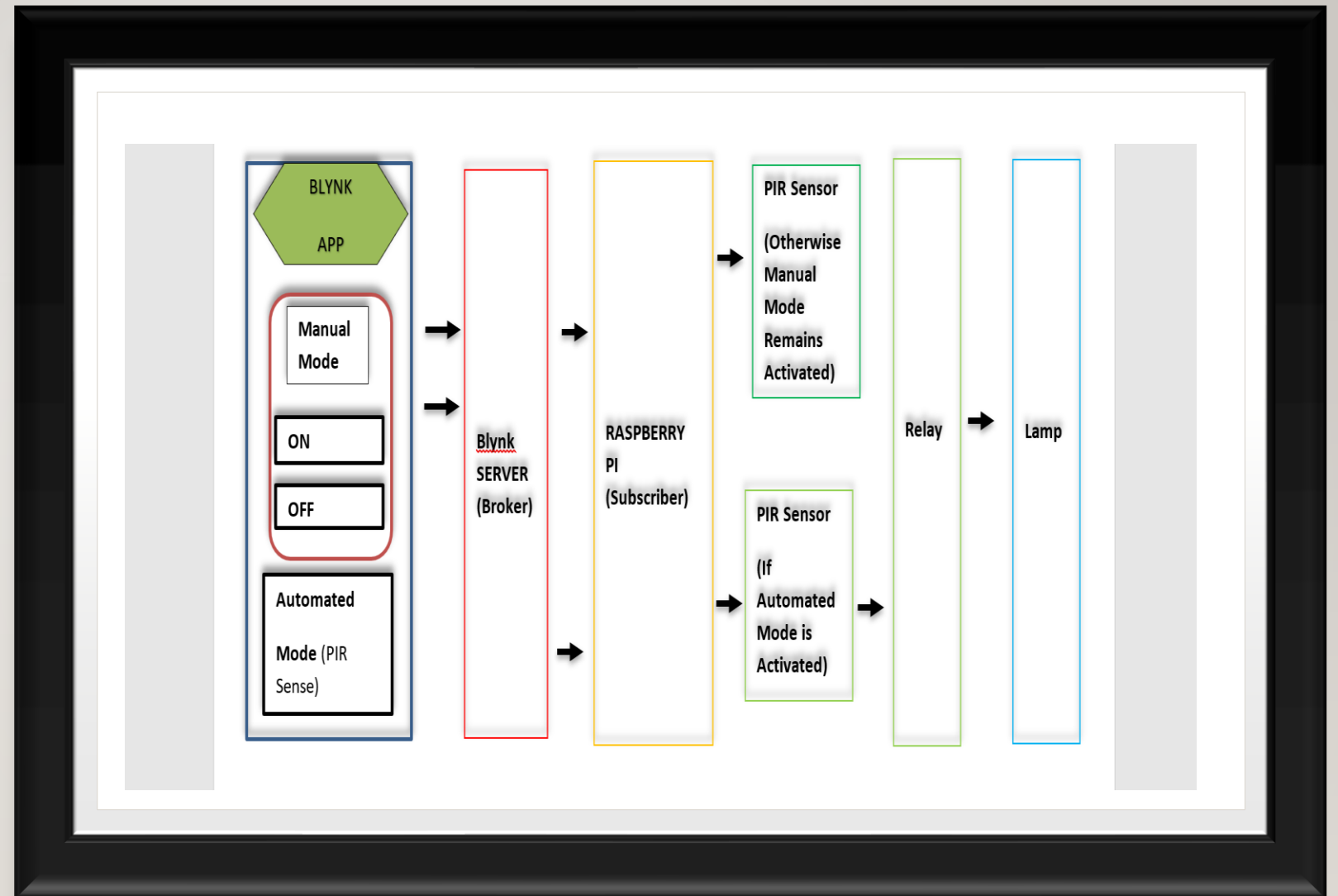
---

- Smartphone & App
- SD Card
- Raspberry Pi3
- Lamp
- 5V Relay
- 2x Resistors (Optional)
- PIR Sensor
- Blynk Application
- Breadboard (Optional)
- 2x LED's (Optional)



# FUNCTIONALITY (HOW IT WORKS!)

---





```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40, GPIO.IN) #Read input from PIR motion sensor
GPIO.setup(22, GPIO.OUT) #YELLOW LED
GPIO.setup(7, GPIO.OUT) #RELAY
try:
    while True:
        input=GPIO.input(40)
        if input==0:
            #When output from motion sensor is LOW
            print("No intruders", input)
            GPIO.output(22, 1) #YELLOW LED IS ON
            GPIO.output(7, 0) #RELAY IS OFF
            time.sleep(0.1)
        elif input==1:
            #When output from motion sensor is HIGH
            print("Intruder detected", input)
            GPIO.output(22, 0) #YELLOW LED IS OFF
            GPIO.output(7, 1) #RELAY IS LIVE
            time.sleep(0.1)
finally:
    GPIO.cleanup()
```

# PYTHON CODE FOR PIR SENSOR

# PYTHON CODE

---

- Sensor reads in Motion Data
- While process is true, turn on Relay
- Python Based Program

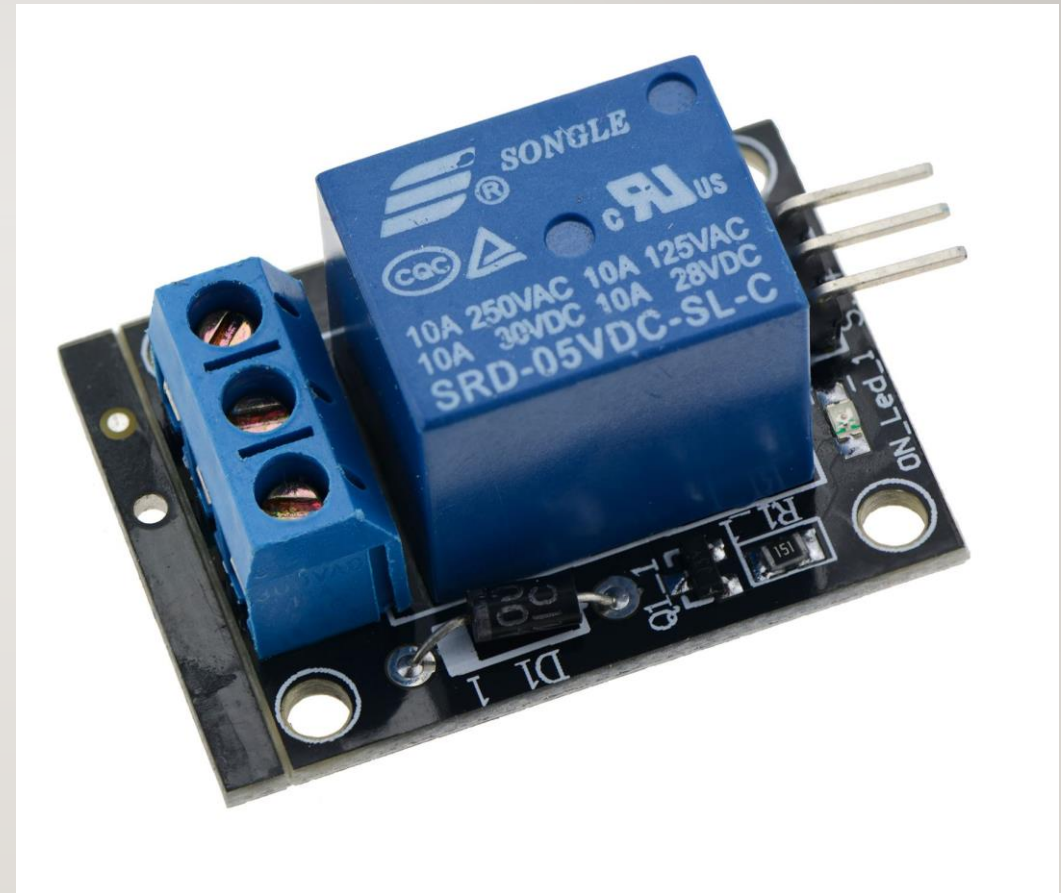
```
GNU nano 2.7.4
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40,GPIO.IN) #Read input from PIR motion sensor
GPIO.setup(22,GPIO.OUT) #YELLOW LED
GPIO.setup(7,GPIO.OUT) #RELAY
try:
    while True:
        input=GPIO.input(40)
        if input==0:
            #When output from motion sensor is LOW
            print("No intruders",input)
            GPIO.output(22,1) #YELLOW LED IS ON
            GPIO.output(7,0) #RELAY IS OFF
            time.sleep(0.1)
        elif input==1:
            #When output from motion sensor is HIGH
            print("Intruder detected",input)
            GPIO.output(22,0) #YELLOW LED IS OFF
            GPIO.output(7,1) #RELAY IS LIVE
            time.sleep(0.1)
finally:
    GPIO.cleanup()
```



# RELAY

---

- Electrically Operated Switch
- Many operate using an electromagnetic switch
- Other use solid-state relays





```
var spawn = require('child_process').spawn;
//var process = spawn('python', ["pirTest.py"], {detached: true});
var Gpio = require('onoff').Gpio; //include onoff to interact with the GPIO
var RELAY = new Gpio(4, 'out'); //use GPIO pin 7 (ON BOARD) and specify that it is this pin
var SENSORSTATE = new Gpio(21, 'out'); //sets the state of the PIR Sensor to 0 ... only need
var Blynk = require("blynk-library");
var AUTH = '388dee48785e45498fcde4458c72c39a';
var blynk = new Blynk.Blynk(AUTH);
var v0 = new blynk.VirtualPin(0);
var v1 = new blynk.VirtualPin(1);

v0.on('write', function(param){
  if(param[0] === '1')
    RELAY.writeSync(1); //1 (turn LAMP ON)
  else RELAY.writeSync(0); //0 (turn LAMP OFF)
  console.log('v0:', param[0]);
})

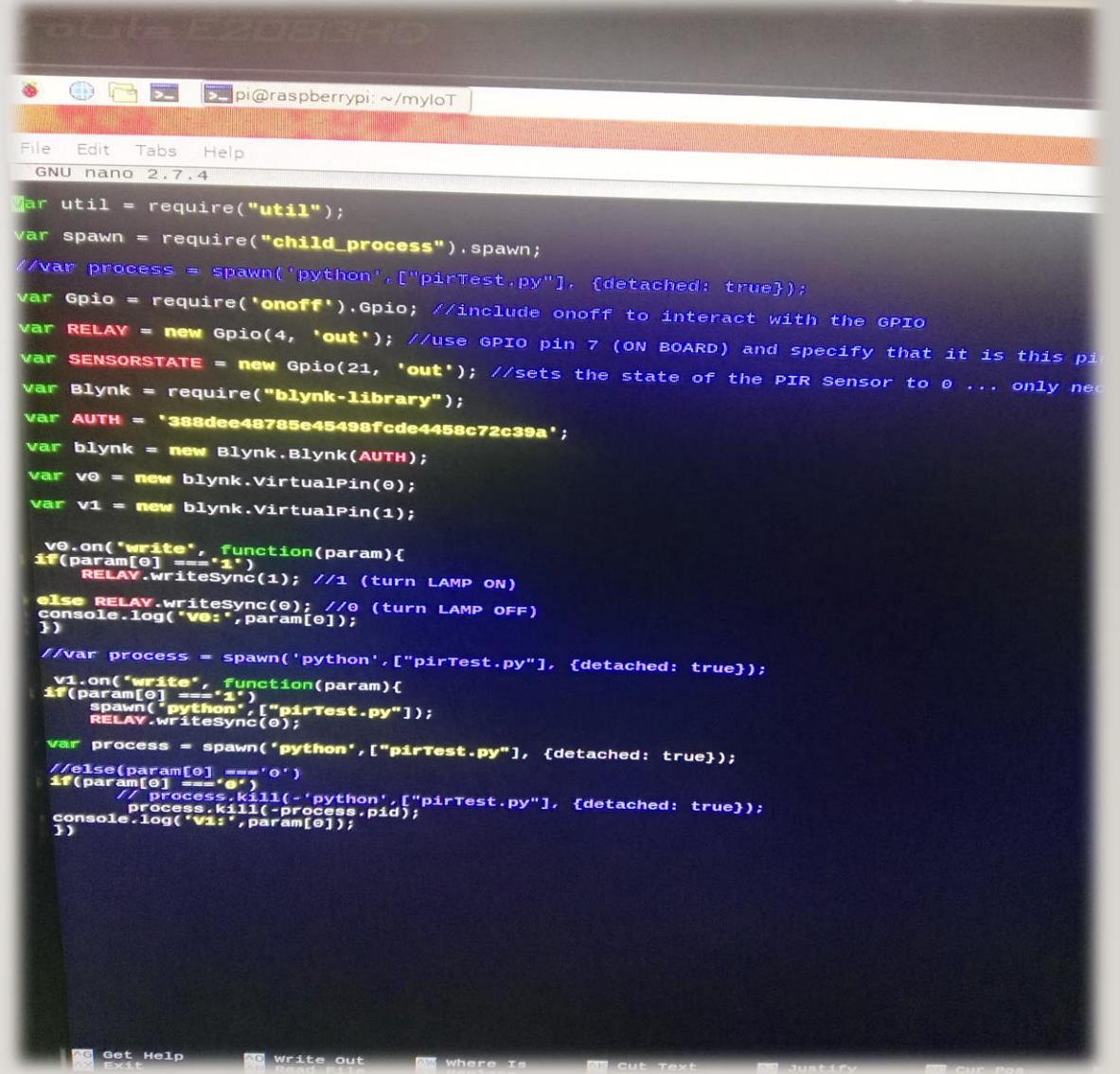
//var process = spawn('python', ["pirTest.py"], {detached: true});
v1.on('write', function(param){
  if(param[0] === '1')
```

## NODE.JS CODE FOR RELAY SWITCH

# NODE.JS CODE FOR RELAY SWITCH & PYTHON SCRIPT LAUNCH

---

- ON/OFF controls the Relay
- Activate/Deactivate controls the PIR Sensor
- Node.js Based Program
- Running Child process module to launch Python Script (PIR Sensor)



```
var util = require("util");
var spawn = require("child_process").spawn;
//var process = spawn('python', ["pirTest.py"], {detached: true});
var Gpio = require('onoff').Gpio; //include onoff to interact with the GPIO
var RELAY = new Gpio(4, 'out'); //use GPIO pin 7 (ON BOARD) and specify that it is this pin
var SENSORSTATE = new Gpio(21, 'out'); //sets the state of the PIR Sensor to 0 ... only need
var Blynk = require("blynk-library");
var AUTH = '388dee48785e45498fcde4458c72c39a';
var blynk = new Blynk.Blynk(AUTH);
var v0 = new blynk.VirtualPin(0);
var v1 = new blynk.VirtualPin(1);

v0.on('write', function(param){
  if(param[0] === '1')
    RELAY.writeSync(1); //1 (turn LAMP ON)
  else RELAY.writeSync(0); //0 (turn LAMP OFF)
  console.log('v0:', param[0]);
});

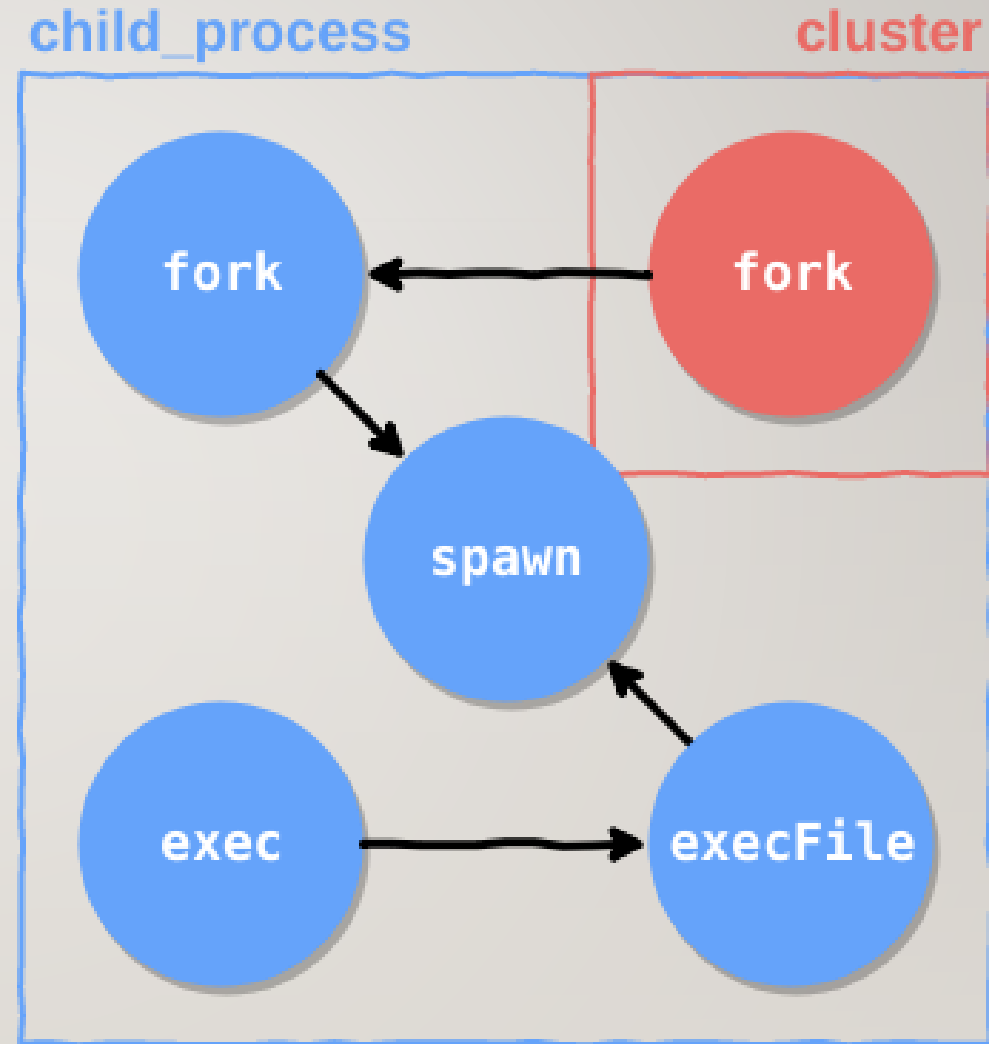
//var process = spawn('python', ["pirTest.py"], {detached: true});
v1.on('write', function(param){
  if(param[0] === '1')
    spawn('python', ["pirTest.py"]);
    RELAY.writeSync(0);
  var process = spawn('python', ["pirTest.py"], {detached: true});
  //else(param[0] === '0')
  if(param[0] === '0')
    // process.kill('-python', ["pirTest.py"], {detached: true});
    process.kill(-process.pid);
  console.log('v1:', param[0]);
});
```



# NODE.JS CHILD PROCESS

---

- Process created by another process
- Allows for ease communication between two different programs
- Pre-built-in Node.js module

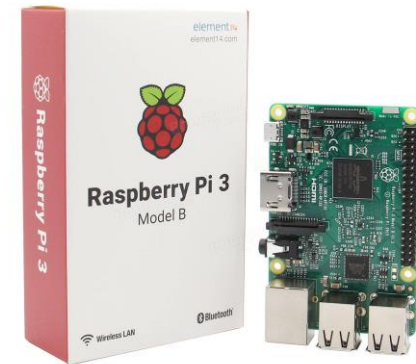
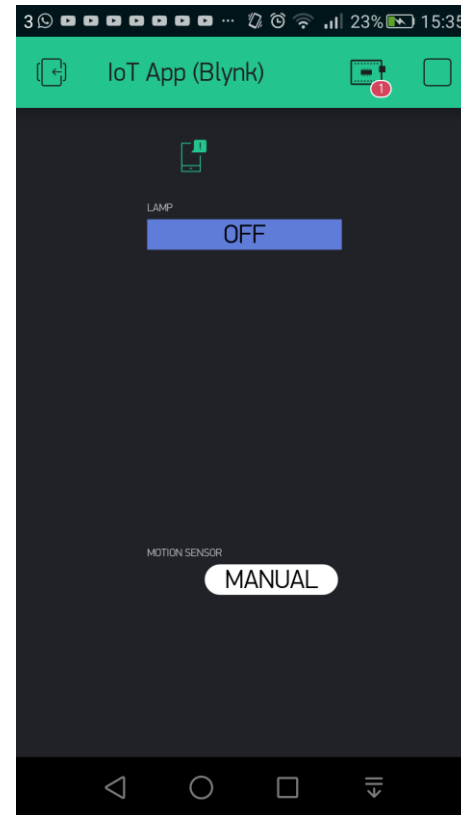




# MAJOR COMPONENTS IN USE

---

- Blynk Application
- Raspberry Pi





# CHALLENGES ENCOUNTERED

---

- Establishing communication between Node.js & Python Scripts
- Disabling the ON/OFF Button when PIR Sensor is active

# THINGS I LEARNED!

---



- Using python & node.js
- Patience
- Critical Thinking
- Source Control

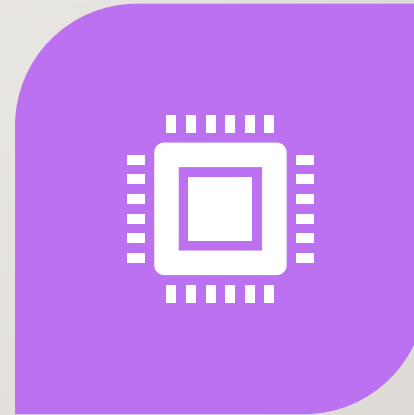


# SUMMARY (FUTURE IMPROVEMENTS)

---



ADD AN OPTION TO USE VOICE  
RECOGNITION TO TURN ON/OFF THE LAMP  
(FOR SATISFACTION & EFFICIENT USE)



RESEARCH INTO ADDING MORE SENSORS  
(FOR MORE SECURE FEATURES)

A yellow lamp with a wide, conical shade. The base of the lamp is a yellow sphere featuring a laughing emoji face with closed, curved eyes and a wide, open-mouthed smile showing teeth. Two grey, teardrop-shaped accents are positioned on either side of the mouth, suggesting tears of joy. The background is a plain, light grey.

THANK YOU!