Crop protection of farms from wild- boars using IOT

Unnatha Bharat Abhiyan

KGRCET

Project by:
TARUN KOTAGIRI
19QM1A0557

Guidance By:
Ms. Udaya Sree Kakarla
Mr. Thangamani

Abstract:

Wild boars are one of the major problems for farmers whose land is nearby to the forest. To stop the wild boars entering into the farm, we developed an IoT device that can detect the wild Boars and give loud alert sounds to scare the wild Boars along with iridium lights glowing partially. The main theme of this project is to save the crop from wild animals without harming them.

Introduction:

India is rich with lots of agricultural lands with lots of nutrients and moisture soil. The crop that grows in our country are exported to a lot of various other countries and we are a hub to produce natural food. Lots of other wild animals like snakes, wild rabbits, wild rats and some other Birds which feed on the agricultural crop and cause some kind of damage to it. Other than these concerns, wild boars are another major issue that farmers are facing in recent times. The wild Boars majorly attack the crop at nighttime and a huge number of them enter into the field directly from the forest. It's a challenging situation for farmers to identify when the Wild Boars enter into the field because of less visibility of light at nighttime and they couldn't realize the damage till the next day morning.

Many engineers worked on this problem statement and came up with solutions like solar electrical fencing. The concept of this solar electrical fencing works in such a way that whenever a wild boar enters into the field and touches the fences, a limited amount of electrical shock is given to the wild boar. Satisfy the requirement of the crop, but as a con's this became a difficulty of saving the wild boar as all the wild birds don't have the same threshold limit for the electric shock.

Prototype:

Considering this as a problem statement of saving the crop as well as the wide boar we came up with a solution using the technology of the internet of things. This project will be using a PIR sensor, node MCU module, Raspberry Pi and a speaker.

A solution goes in such a way that whenever a wild Boar is detected by PIR sensor the information is forwarded to the Raspberry Pi 3 which receives the message from the PIR sensor and accordingly place multiple soundtracks using a loudspeaker like gun firing, dog shouting, people shouting and other scary sounds along with multiple iridium lights glowing randomly.

Here, the PR sensors are placed along with the parameter of the field covering the Rangers according to each PIR sensor. The PIR sensor is connected to the node MCU board, which has an inbuilt Wi-Fi module. Whenever the PIR sensor detects some motion before them, it sends an alert message to the Raspberry Pi using the MQTT server.

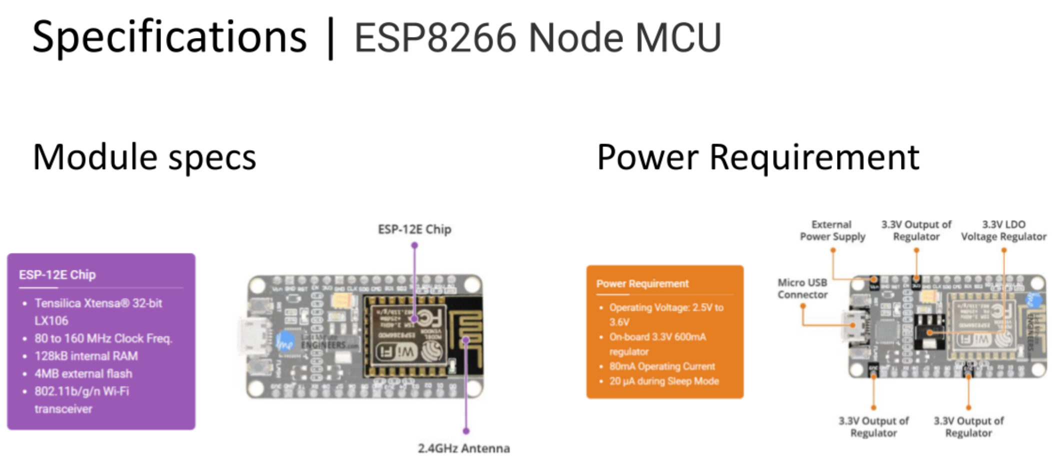
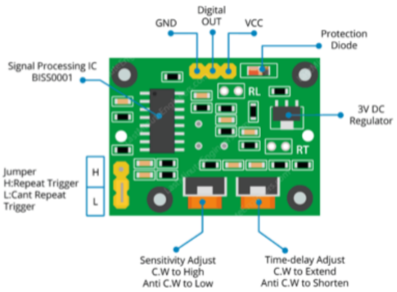


Fig-1: NODE MCU ESP8266

Specifications | PIR Sensor (HC-SR501)

Module specs



Power Requirement

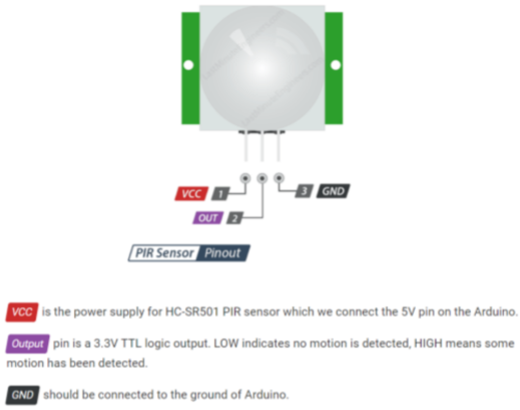
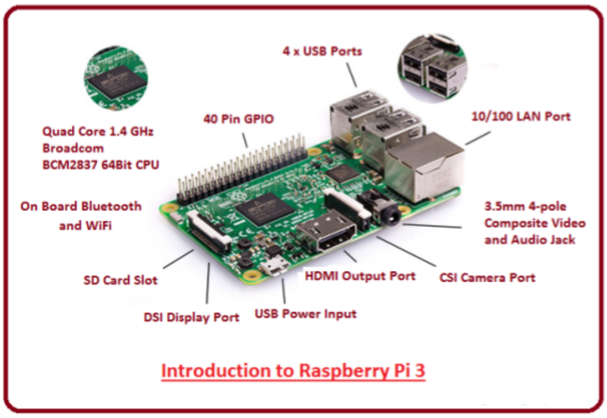


Fig-2: PIR SENSOR

Specifications | Raspberry Pi

Module specs



Power Requirement

	Raspberry Pi 3 Model B
Introduction Date	2/29/2016
SoC	BCM2837
CPU	Quad Cortex A53 @ 1.2GHz
Instruction set	ARMv8-A
GPU	400MHz VideoCore IV
RAM	1GB SDRAM
Storage	micro-SD
Ethernet	10/100
Wireless	802.11n / Bluetooth 4.0
Video Output	HDMI / Composite
Audio Output	HDMI / Headphone
GPIO	40
Price	$35

Fig-3: Raspberry-PI

Procedure:

- **Step 1**: The PIR sensors are allocated at the perimeters of the field according to the radius of the sensor.
- **Step 2:** A minimum of three sensors are connected to each wifi module (ESP8266 Nodemcu Esp8266)
- **Step 3**: All the three sensors run under a loop in the wifi module.
- **Step 4:** If the value of the sensor is high, The wifi module uses a WPA supplicant to connect with the multimedia processor.
- **Step 5:** As soon as the processor receives a wireless connection, It starts implementing the code of the speaker, which produces a massive sound randomly whenever it is established with a WPA connection.
- **Step 6:** Along with the speaker, Iridium lights starts glowing in the form of a human.

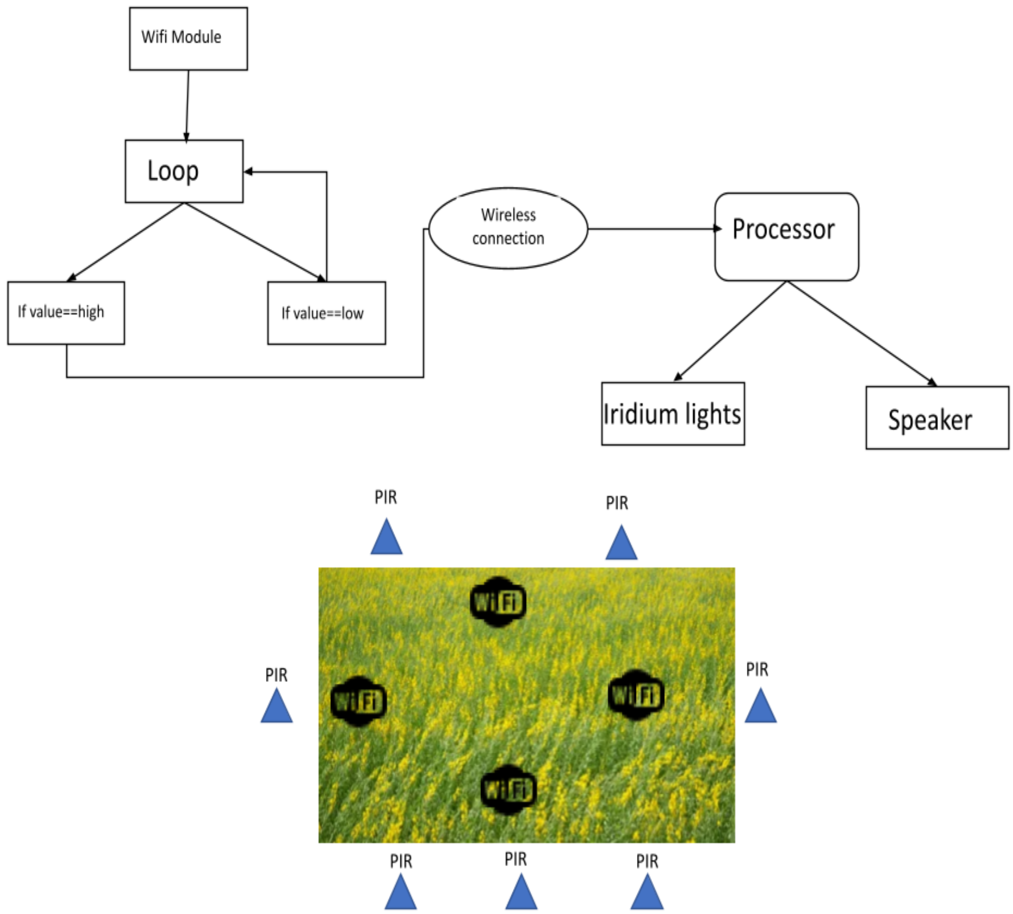


Fig-4: Flow-chart

CODE:

PIR SENSOR:

```
#include "PubSubClient.h" // connect and publish to the mqtt broker
#include "ESP8266WiFi.h" // enables esp to connect to the local wifi
#define PIR_1 D1
#define PIR_2 D2

int pir_value;

// WiFi
const char* ssid = "STUDENT_AFFAIRS";
const char* wifi_password = "**********";


//MQTT

const char* mqtt_server = "192.168.2.103";
const char* alert_topic = "boar/alert";

const char* mqtt_username = "testUser";
const char* mqtt_password = "test";
const char* clientID = "client_farm";

// Initialize the Wifi and MQTT client objects

WiFiClient wifiClient;

// 1883 is the listener port for the Broker
PubSubClient client(mqtt_server, 1883, wifiClient);

// custom function to connect to the mqtt broker via wifi

void connect_MQTT(){
  Serial.print("Connecting to ");
  Serial.println(ssid);

  // connect to the WiFi
  WiFi.begin(ssid, wifi_password);

  while(WiFi.status()!= WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }

  // Debugging - Outputting the ip for testing
  Serial.println("WiFI connection established");
  Serial.println("IP address : ");
  Serial.println(WiFi.localIP());


  // connect to MQTT Broker
```

```
}

void setup() {
  // put your setup code here, to run once:
  pinMode(PIR_1, INPUT);
  pinMode(PIR_2, INPUT);
  Serial.begin(9600);
  connect_MQTT();

}

void loop() {




  Serial.setTimeout(2000);
  pir_value = digitalRead(PIR_1);

  Serial.println(!pir_value);
//  Serial.println("Testing");

  // MQTT can only transmit strings

  String alert_data = String((int)pir_value);

  // publish to mqtt broker
 if(pir_value == LOW){

  if(client.connect(clientID,mqtt_username, mqtt_password)){
    Serial.println("Connected to MQTT Broker!");
  } else{
    Serial.println("Connection to MQTT Broker failed..");
  }

   if(client.publish(alert_topic, String(pir_value).c_str())){
    Serial.println("Alert sent to the RPi");
      } else{
           //Failed to tranmit data. Reconnecting to MQTT broker and trying
      again.
    Serial.println("Reconnecting to MQTT broker");
    client.connect(clientID, mqtt_username, mqtt_password);
     delay(10); // this delay ensures that client.publish doesn't clash with
      the client.connect call
    client.publish(alert_topic, String(pir_value).c_str());
  }
 }

  client.disconnect();
  delay(1000*3);  // execute this loop every 2 seconds.
}
```

RASPBERRY PI:

```
import paho.mqtt.client as mqtt
import os
import random
from pydub import AudioSegment
from pydub.playback import play
import RPi.GPIO as GPIO
import time

# LED FUNCTIONALITY

LED_PIN = 17

MQTT_ADDRESS = '192.168.2.103'
MQTT_USER = 'testUser'
MQTT_PASSWORD = 'test'
MQTT_TOPIC = 'boar/alert'

def on_connect(client, userdata, flags, rc):
    """ The callback for when the client receives a CONNACK response from the
      server."""
    print('Connected with result code ' + str(rc))
    client.subscribe(MQTT_TOPIC)


def on_message(client, userdata, msg):
       """The callback for when a PUBLISH message is received from the
      server."""
    print(msg.topic + ' ' + str(msg.payload))
    if str(msg.payload):
        path = '/home/pi/Desktop/audio'
        audios = os.listdir(path)
        d = random.choice(audios)
        play(AudioSegment.from_mp3(path + '/' + d))
        GPIO.setmode(GPIO.BCM)

        GPIO.setup(LED_PIN, GPIO.OUT)
        i = 0
        while(i < 10):
            GPIO.output(LED_PIN,GPIO.HIGH)
            time.sleep(0.5)
            GPIO.output(LED_PIN,GPIO.LOW)
            time.sleep(0.5)
            i += 1

        GPIO.cleanup()

def main():
    mqtt_client = mqtt.Client()
    mqtt_client.username_pw_set(MQTT_USER, MQTT_PASSWORD)
    mqtt_client.on_connect = on_connect
    mqtt_client.on_message = on_message

    mqtt_client.connect(MQTT_ADDRESS, 1883)
    mqtt_client.loop_forever()
```

```
if __name__ == '__main__':
    print('MQTT to InfluxDB bridge')
    main()
```

Result:

We have successfully demonstrated our prototype there whenever the PR sensor detects a moment before PIR sense the information as one and forwards this to the Raspberry Pi 3 using MQTT server, whenever the Raspberry Pi receives the input as one from the PIR sensor it activates the speaker and also Iridium lights using the random function of the Python library. The maximum range of the PIR sensor was 15 meters, so we placed each PIR sensor for the approximate value of 12 meters.

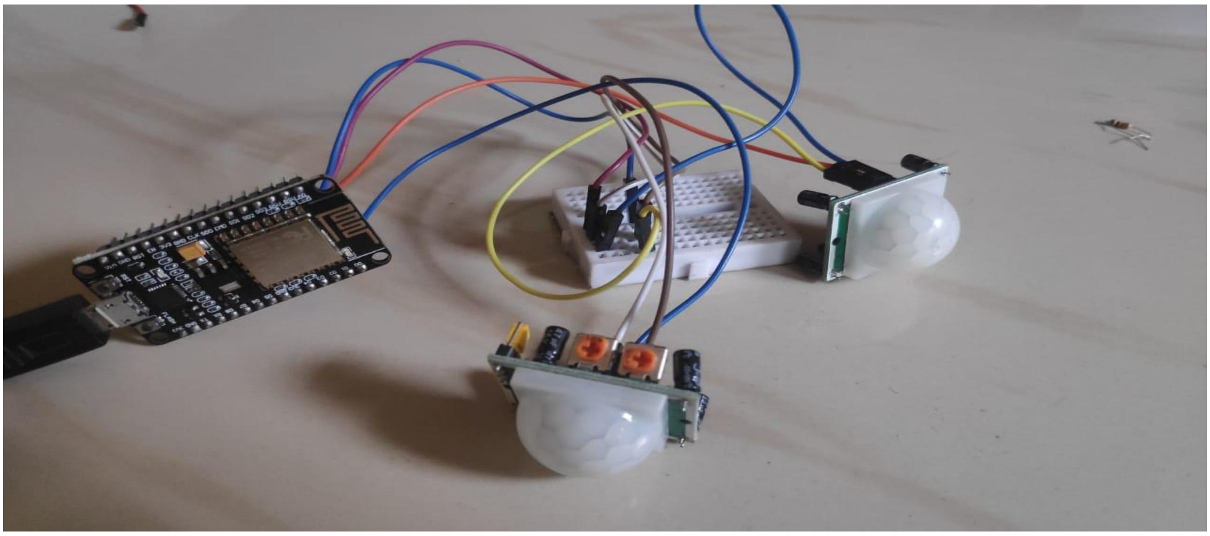


Fig-5: PIR sensor's connected to NODE MCU

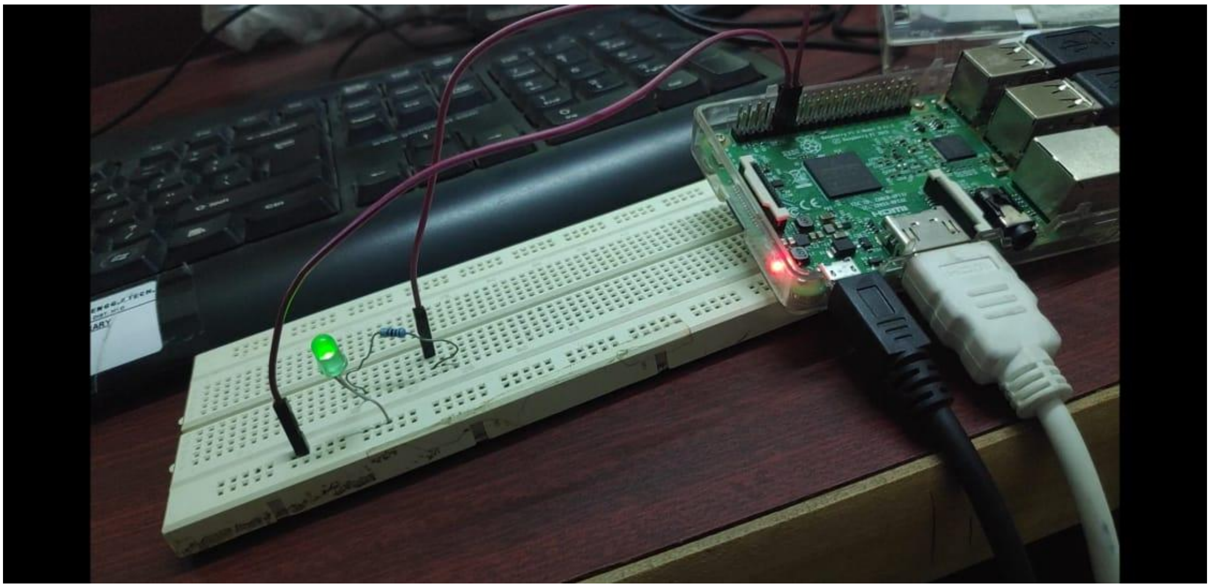


Fig-6: Raspberry Pi Receiving signal from NODE MCU

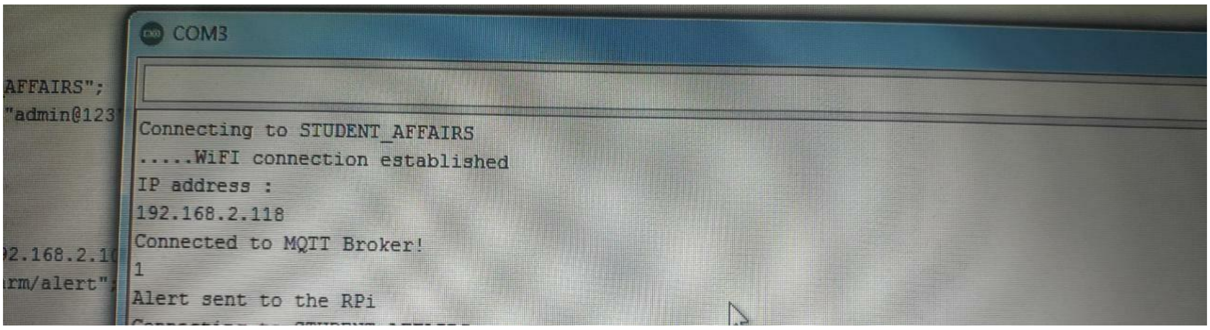


Fig-7: Output of NODE MCU after connecting with Raspberry Pi

Conclusion:

Finally, we achieved the solution for our problem statement of saving both the wild boars and crop, which out harming the animals and environment.