

INSTITUTO TECNOLÓGICO DE OAXACA

INGENIERIA EN SISTEMAS COMPUTACIONALES

MATERIA: SISTEMAS PROGRAMABLES GRUPO: ISA

Practica 3

ENVIAR DATOS POR MEDIO DE INFRARROJOS

ALUMNOS:

CHÁVEZ PÉREZ MIGUEL EDUARDO

OROCIO SANTIAGO IVAN

Contenido

Índice de ilustraciones.....	3
Definición	4
Componentes	4
Protoboard	4
Sensor PIR HC-SR501.....	5
Especificaciones	6
Sensor DHT11	7
Especificaciones	7
Sensor MQ2.....	8
Especificaciones	8
Led IR 383	9
Especificaciones	9
Fototransistor.....	10
Especificaciones	10
Resistencia 100 Ω	10
Especificaciones	11
Resistencia 10k Ω	11
Especificaciones	11
RS232 USB TTL STC-ISP	12
Especificaciones	12
Arduino Nano	13
Especificaciones	13
Transistor BC547	14
Características	14
Cables Jumper	15
Diseño.....	15
Emisor.....	15
Receptor	17
Código	19
Comprobación	24
Pruebas.....	24
Resultados	25

Mediciones	27
Tensión	27
Corriente	28
Conclusiones	30

Índice de ilustraciones

Ilustración 1 Protoboard	4
Ilustración 2 Sensor PIR.....	5
Ilustración 3 Rango de detección del Sensor PIR HC-SR501	6
Ilustración 4 Sensor DTH11	7
Ilustración 5 Sensor MQ-2.....	8
Ilustración 6 Led IR 383	9
Ilustración 7 Fototransistor	10
Ilustración 8 Resistencia 100 Ohm	10
Ilustración 9 Resistencia 10k Ohm	11
Ilustración 10 RS232 USB TTL STC-ISP	12
Ilustración 11 Arduino Nano v3.....	13
Ilustración 12 Transistor BC547	14
Ilustración 13 Cables Jumper	15
Ilustración 14 Diagrama de conexión Emisor	16
Ilustración 15 Diagrama esquemático emisor.....	17
Ilustración 16 Diagrama del Receptor	18
Ilustración 17 Diagrama esquemático Receptor	18
Ilustración 18 Prueba en distancia mínima	24
Ilustración 19 Prueba en distancia máxima	25
Ilustración 20 Grafica PIR	25
Ilustración 21 Grafica de la temperatura	26
Ilustración 22 Grafica de la Humedad	26
Ilustración 23 Detección de Gas.....	27

Definición

Censar datos presencia con el sensor PIR, humedad y temperatura con el sensor DTH11, y gas por medio del sensor MQ2, posteriormente enviar los datos a una pc por medio de un led infrarrojo emisor y un receptor conectado a un RS232.

Componentes

Protoboard

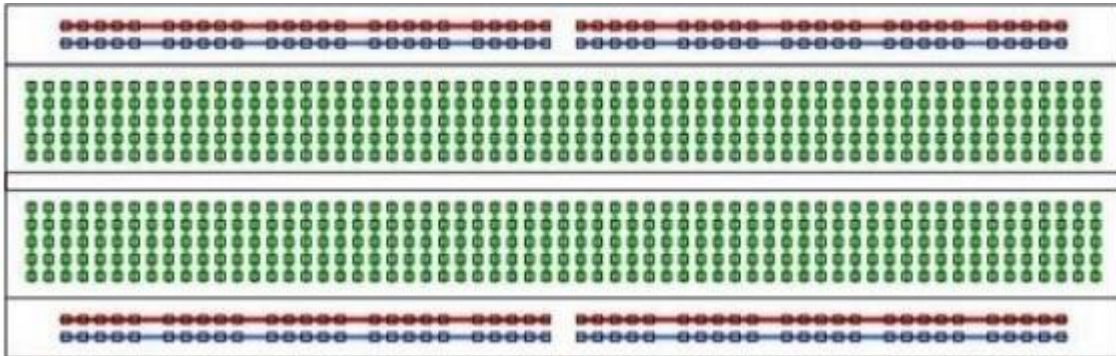


Ilustración 1 Protoboard

El protoboard está dividido en dos áreas principales que son los buses y las pistas. Los buses tienen conexión y por ende conducen a todo lo largo (aunque algunos fabricantes dividen ese largo en dos partes).

Las líneas rojas y azules te indican como conducen los buses. No existe conexión física entre ellos, es decir, no hay conducción entre las líneas rojas y azules. En los buses se acostumbra a conectar la fuente de poder que usan los circuitos o las señales que quieres inyectarles a ellos desde un equipo externo. Por su parte, las pistas (en morado) te proveen puntos de contacto para los pines o terminales de los componentes que colocas en el protoboard siguiendo el esquemático de tu circuito, y conducen como están dibujadas. Son iguales en todo el protoboard. Las líneas moradas no tienen conexión física entre ellas.

Sensor PIR HC-SR501

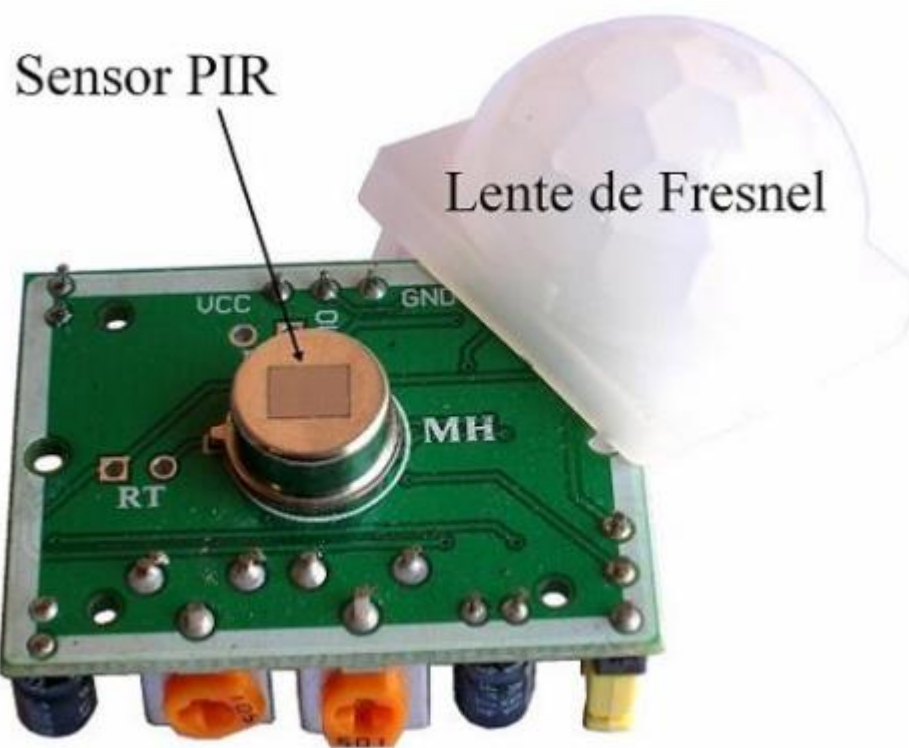


Ilustración 2 Sensor PIR

En los sensores de movimiento, el sensor PIR consta en realidad de 2 elementos detectores separados, siendo la señal diferencial entre ambos la que permite activar la alarma de movimiento. En el caso del HC-SR501, la señal generada por el sensor ingresa al circuito integrado BISS0001, el cual contiene amplificadores operacionales e interfaces electrónicas adicionales. Las funciones y ajustes complementarios del sensor de movimiento son:

- Ajuste de parámetros: mediante 2 potenciómetros, el usuario puede modificar tanto la sensibilidad como la distancia de detección del PIR.
- Detección automática de luz (esta función no está disponible al adquirir el sensor de fábrica): por medio de una foto resistencia CdS (Sulfuro de Cadmio), se deshabilita la operación del sensor en caso que exista suficiente luz visible en el área. Esta función es utilizada en caso de sensores que enciendan lámparas en lugares poco iluminados durante la noche, y especialmente en corredores ó escaleras.

Rango de detección de los sensores PIR: Como se indicó anteriormente, el rango de detección de movimiento de los PIR es ajustable y generalmente funcionan con alcances de hasta 7 metros, y con aperturas de 90° a 110°, como se muestra en la figura. El montaje del PIR puede realizarse tanto en piso, muro o techo, según convenga a la aplicación.

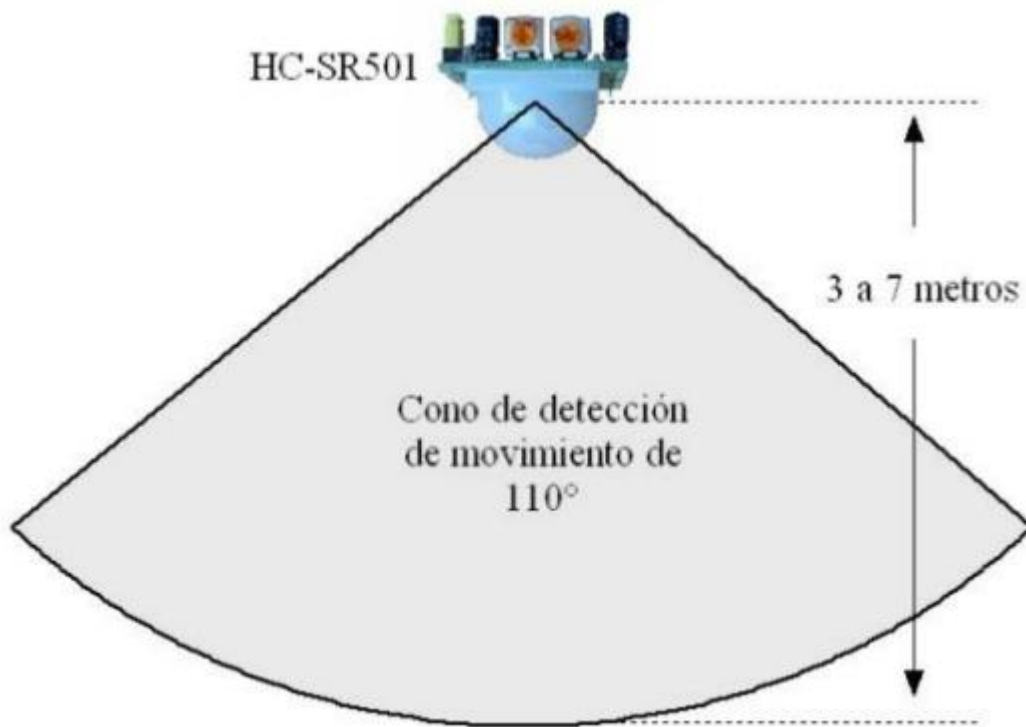


Ilustración 3 Rango de detección del Sensor PIR HC-SR501

Especificaciones

- Usa el PIR LHI778 y el controlador BISS0001.
- Voltaje de alimentación: de 5 a 12 VDC.
- Consumo promedio: <1 mA.
- Rango de distancia: 3 a 7 metros ajustable.
- Angulo de detección. Cono de 110°
- Ajustes: 2 potenciómetros para ajuste de rango de detección y tiempo de alarma activa.
- Jumper para configurar la salida de alarma en modo mono-disparo o disparo repetitivo.
- Salida de alarma de movimiento con ajuste de tiempo entre 3 segundos a 5 minutos.
- Salida de alarma activa Vo con nivel alto de 3.3 V y 5mA, lista para conexión de un led, ó un transistor y relevador.
- Tiempo de inicialización: después de alimentar el módulo HC-SR05, debe transcurrir 1 minuto antes de que inicie su operación normal. Durante ese tiempo, es posible que el módulo active 2 ó 3 veces su salida.
- Tiempo de salida inactiva: cada vez que la salida pase de activa a inactiva, permanecerá en ese estado los siguientes 3 segundos. Cualquier evento que ocurra durante ese lapso es ignorado.
- Temperatura de operación: -15° a +70° C.
- Dimensiones: 3.2 x 2.4 x 1.8 cms.

Sensor DHT11

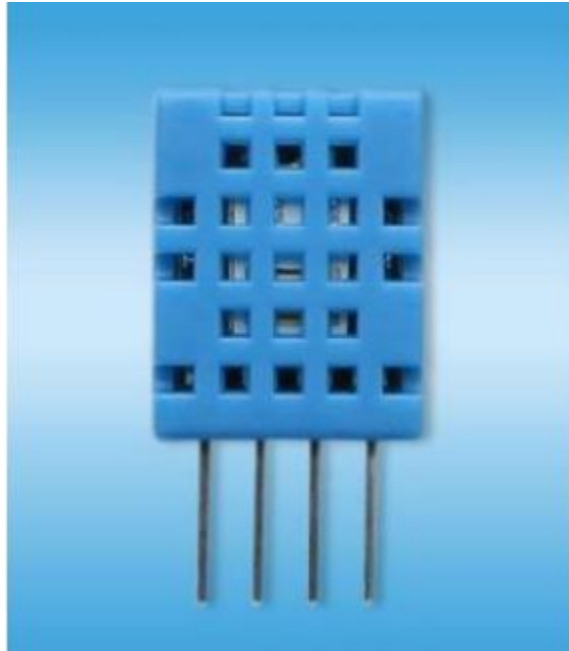


Ilustración 4 Sensor DHT11

El sensor digital de temperatura y humedad es un sensor compuesto que contiene una salida de señal digital calibrada de la temperatura y la humedad. Aplicación de una tecnología de recolección de módulos digitales dedicada y la tecnología de detección de temperatura y humedad, para garantizar que el producto tenga una alta confiabilidad y una excelente estabilidad a largo plazo. El sensor incluye una sensación sensorial de componentes húmedos y dispositivos de medición de temperatura NTC, y está conectado con un microcontrolador de alto rendimiento de 8 bits.

Especificaciones

Humedad relativa

- Resolución: 16 bit
- Repetibilidad: $\pm 1\%$ de HR
- Precisión: a $25^{\circ}\text{C} \pm 5\%$ de HR
- Intercambiabilidad: completamente intercambiable
- Tiempo de respuesta: 1 / e (63%) de 25°C 6s
- 1 m / s de aire 6s
- Histéresis: $< \pm 0.3\%$ RH
- Estabilidad a largo plazo: $< \pm 0.5\%$ HR

Temperatura

- Resolución: 16 bits
- Repetibilidad: $\pm 0.2^{\circ}\text{C}$
- Rango: a $25^{\circ}\text{C} \pm 2^{\circ}\text{C}$

- Tiempo de respuesta: 1 / e (63%) 10S

Características eléctricas

- Fuente de alimentación: DC 3.5 ~ 5.5V
- Corriente de suministro: medida 0.3mA en espera 60μA

Sensor MQ2

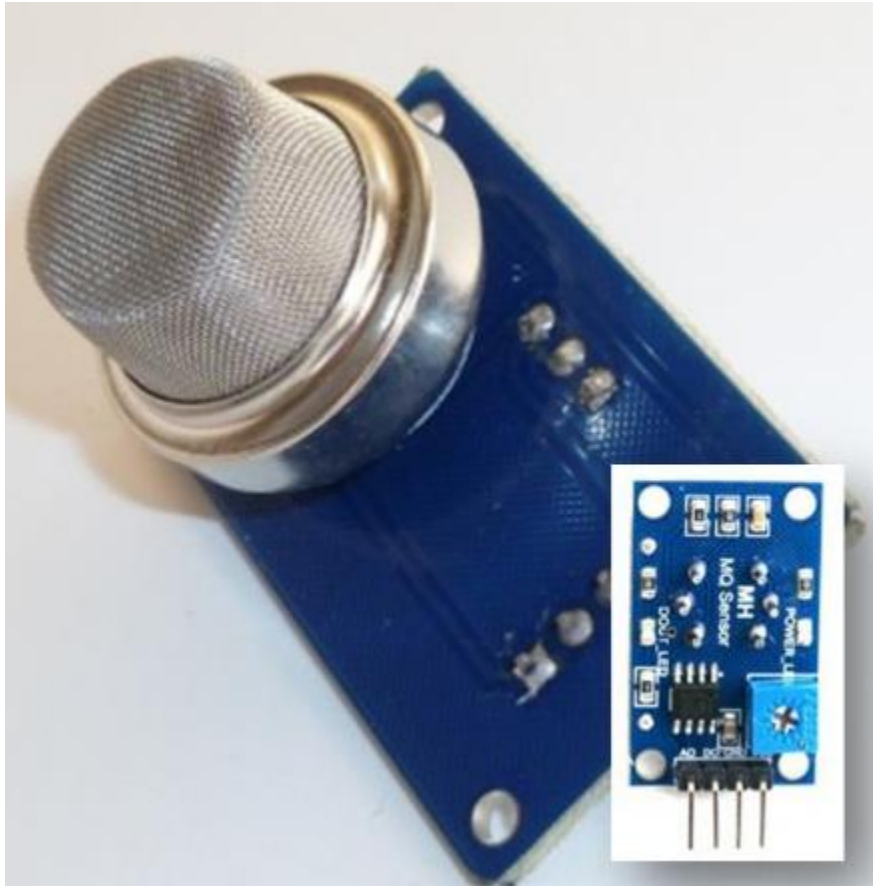


Ilustración 5 Sensor MQ-2

El sensor de Gas MQ-2 puede detectar o medir gases como LPG, Alcohol, Propano, Hidrógeno, CO e incluso metano. La versión del módulo de este sensor viene con un Pin Digital que hace que este sensor funcione incluso sin un microcontrolador y es muy útil cuando solo está tratando de detectar un gas en particular. Cuando se trata de medir el gas en ppm, se debe utilizar el pin analógico, el pin analógico también TTL y funciona en 5V y, por lo tanto, se puede usar con la mayoría de los microcontroladores comunes.

Especificaciones

- Voltaje de funcionamiento es 5V.
- Se puede usar para medir o detectar GLP, alcohol, propano, hidrógeno, CO e incluso metano.
- Tensión de salida analógica: 0V a 5V
- Voltaje de salida digital: 0 V o 5 V (TTL Lógico)

- Duración de precalentamiento 20 segundos
- Se puede usar como un sensor digital o analógico
- La sensibilidad del pin digital se puede variar usando el potenciómetro
- Consumo: menos de 800mW

Led IR 383



Ilustración 6 Led IR 383

Por parte del emisor tenemos un diodos infrarrojo (IR) que funciona convirtiendo la corriente eléctrica en luz infrarroja. Lo cual emiten una longitud de onda de luz que el ojo humano no es capaz de ver, pero que puede ser detectado por un receptor como un fotodiodo o fototransistor.

Especificaciones

- Tipo de encapsulado: LED
- Tipo de LED: 5 mm
- Voltaje invertido: 5 V
- Potencia de disipación: 150 mW
- Longitud de onda: 940 nm
- Sin plomo
- Modelo: IR383
- Corriente: 100mA

Fototransistor



Ilustración 7 Fototransistor

Son sensibles a la luz, normalmente a los infrarrojos. La luz incide sobre la región de base, generando portadores en ella. Esta carga de base lleva el transistor al estado de conducción.

Un fototransistor es igual a un transistor común, con la diferencia que el primero puede trabajar de 2 formas:

- Como transistor normal con la corriente de base I_b (modo común).
- Como fototransistor, cuando la luz que incide en este elemento hace las veces de corriente de base. I_p (modo de iluminación).

Puede utilizarse de las dos en formas simultáneamente, aunque el fototransistor se utiliza principalmente con el pin de la base sin conectar.

Especificaciones

- Longitud de onda: 940nm
- Diámetro: 5mm
- Voltaje de emisor a colector: 5 V
- Corriente colector: 20 mA
- Tiempo de recuperación: 5 μ S
- Contiene filtro de luz de día
- 2 terminales
- Alcance: 9metros sin obstáculos.

Resistencia 100 Ω



Ilustración 8 Resistencia 100 Ohm

Una resistencia es un elemento pasivo que disipa energía en forma de calor según la ley de Joule. También establece una relación de proporcionalidad entre la intensidad de corriente que la atraviesa y la tensión medible entre sus extremos, relación conocida como ley de Ohm. En general,

una resistencia podrá tener diferente comportamiento en función del tipo de corriente que circule por ella.

Esta resistencia es la que se conectará al ánodo del led IR, elegimos esta de acuerdo a los cálculos con la ley de ohm.

$$R = \frac{V}{I}$$

$$R = \frac{5}{0.1} = 50$$

Como no existe una resistencia comercial de 50 Ω , la más cercana es de 100 Ω .

Especificaciones

- Valor: 100 ohm
- Potencia: ¼ W
- Tolerancia: $\pm 1\%$

Resistencia 10k Ω



Ilustración 9 Resistencia 10k Ohm

Se usan resistencias de 10 k Ω para hacer una configuración pull up en el circuito del fototransistor, pues cuando el circuito está en reposo, la caída de tensión es de 5V (HIGH), interpretando un uno lógico.

Especificaciones

- Valor: 10K ohms
- Tolerancia: 5%
- Potencia: 2W

RS232 USB TTL STC-ISP

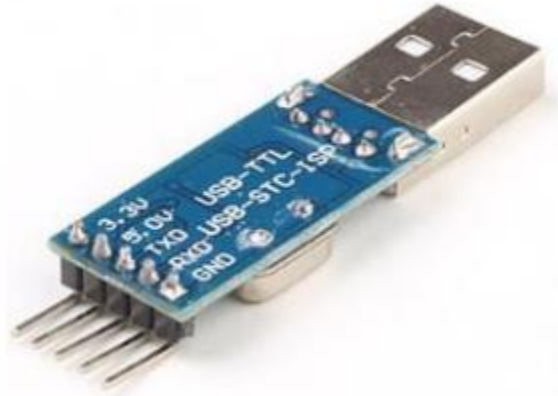


Ilustración 10 RS232 USB TTL STC-ISP

El conversor que un microcontrolador y una PC se comuniquen utilizando el protocolo USB de forma sencilla. Es compatible con cualquier microcontrolador como PIC, Atmel AVR, Arduino y ESP8266.

Especificaciones

- USB 2.0 de velocidad completa de 12 Mbps
- Conector USB: USB tipo A
- Pines salida (TTL): +3.3V, RST, TXD, RXD, GND y + 5V
- Cristal oscilador integrado
- Regulador de voltaje de 3.3V interno
- Buffer de recepción de 576 Bytes
- Buffer de transmisión de 640 Bytes
- Temperatura de trabajo: -40° a 80°C
- Sistemas Operativos soportados: Windows 10, 8, Vista, 7, XP, 2000, 98SE y Linux 2.40(en adelante)
- Dimensiones: 21 mm x 16 mm

Arduino Nano



Ilustración 11 Arduino Nano v3

El Arduino Nano es un tablero pequeño, completo y fácil de usar basado en el ATmega328P (Arduino Nano 3.x). Tiene más o menos la misma funcionalidad del Arduino Duemilanove, pero en un paquete diferente. Le falta solo un conector de alimentación de CC, y funciona con un cable USB Mini-B en lugar de uno estándar.

Especificaciones

- Microcontrolador: Atmel ATmega328 (ATmega168 versiones anteriores)
- Tensión de Operación (nivel lógico): 5 V.
- Tensión de Entrada (recomendado): 7-12 V.
- Tensión de Entrada (límites): 6-20 V.
- Pines E/S Digitales: 14 (de los cuales 6 proveen de salida PWM).

Transistor BC547



Ilustración 12 Transistor BC547

Es un transistor amplificador de audio y VHF Freq. Driver con una corriente máxima de colector de 0.6 A, en su composición posee una placa de semiconductor con tres regiones consecutivas de diferente conductibilidad eléctrica los cuales forman dos uniones n-p-n, las dos regiones extremas tienen un mismo tipo de conductibilidad, la intermedia, conductibilidad de otro tipo, estas son llamadas emisor, colector y base.

Características

- Polaridad (N-P-N)
- Amplificador, audio to VHF Freq. Driver
- Corriente máxima de colector (I_c) 0.6 A
- De colector a base (CBO) 75 Voltios
- De colector a emisor (CEO) 40 Voltios
- De emisor a base (EBO) 6 Voltios
- Ganancia típica de la corriente directa (h_{fe}) 200 Min
- Máxima disipación de potencia en colector (P_d) 0.625 (Watts)
- Frecuencia en (MHz) 300 Min

Cables Jumper



Ilustración 13 Cables Jumper

Es un cable con un conector en cada punta hembra o macho, que se usa normalmente para interconectar entre sí los componentes en una placa de pruebas, se utilizan de forma general para transferir señales eléctricas de cualquier parte de la placa de prototipos a los pines de entrada/salida de un microcontrolador.

Diseño

En total se utilizarán 3 pines del arduino nano para los datos procedentes de los sensores y uno para el led IR, sin contar los pines de 5v y GND.

Emisor

PIR

Pin Pir	Pin Arduino nano
VCC	5v
GND	GND
OUT	D3

DHT11

Pin DHT11	Pin Arduino nano
-----------	------------------

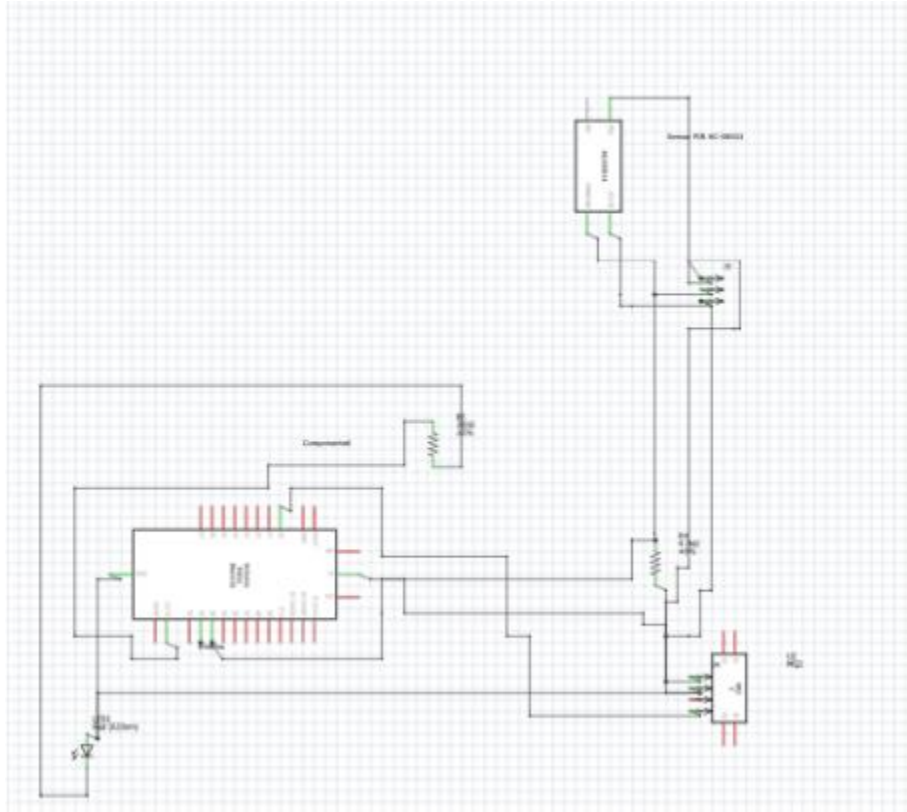


Ilustración 15 Diagrama esquemático emisor

Receptor

Fototransistor

Pin RSC232	Fototransistor
	Colector
GND	Emisor

Transistor BC547

Pin RSC232	Pines Transistor BC547
RDX -> R 10kOhm	Colector
	Base
GND	Emisor

Diagrama del Receptor

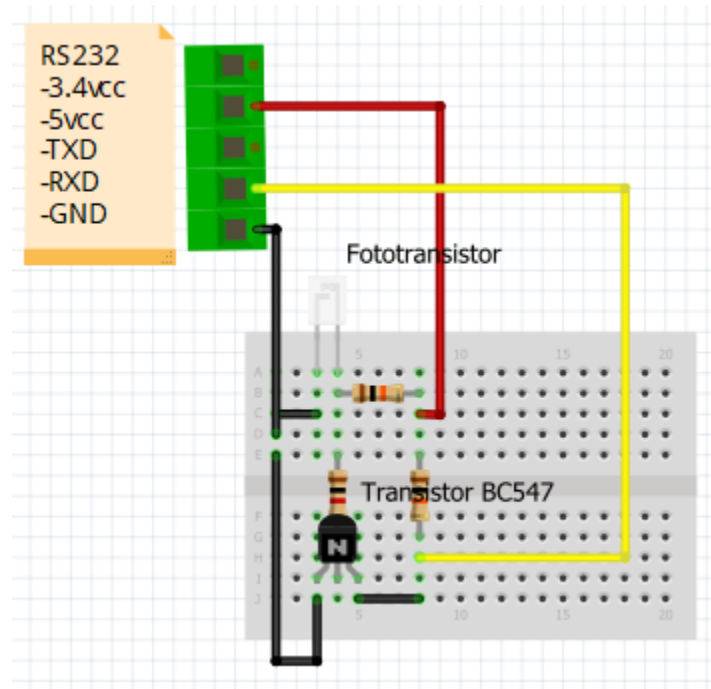


Ilustración 16 Diagrama del Receptor

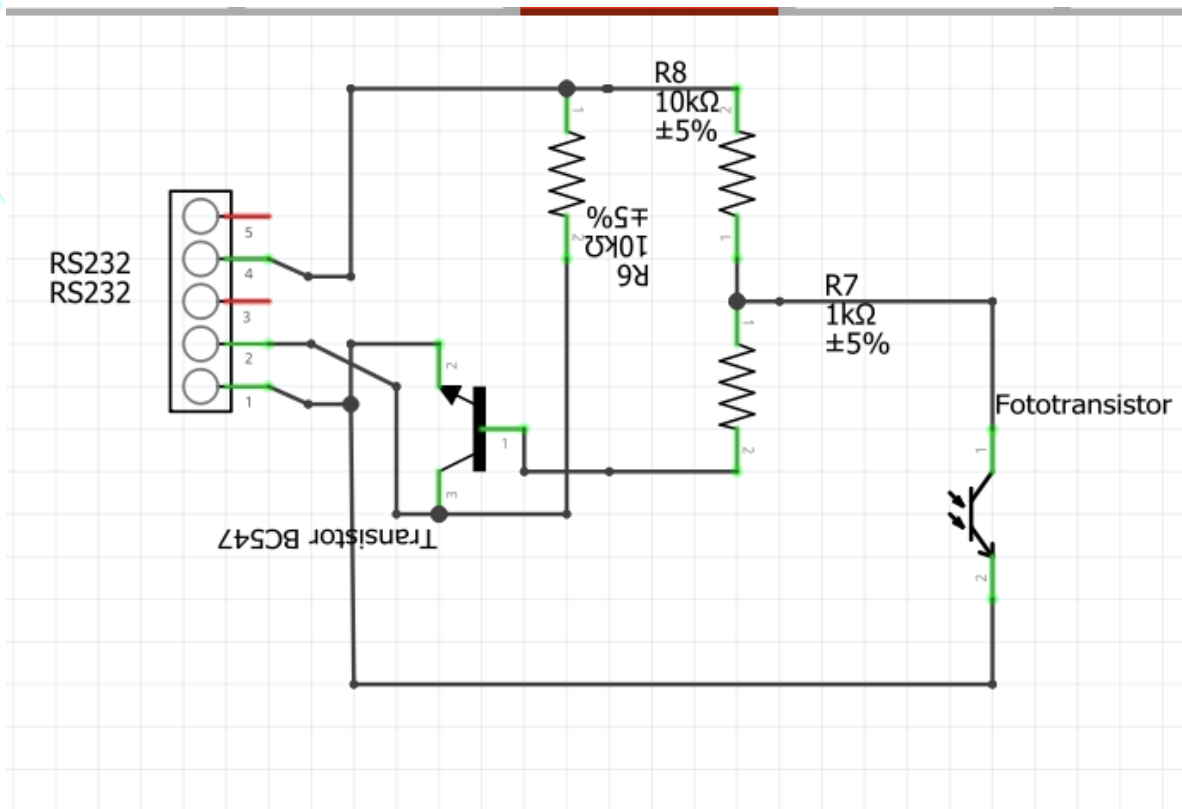


Ilustración 17 Diagrama esquemático Receptor

Código

Emisor

Enviamos la información por medio del pin serial TX del arduino. se usa una interrupción para saber el momento en el que un pir entrega una señal en alto, y con un correspondiente método que notifica al receptor, también en el setup se hace el censado de los sensores para inicializar variables, después de ello, en el loop se censan nuevamente y envían los datos cuando existe una un cambio de algún sensor.

```
#include <DHT.h>
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
int mq2 = A0;
int pir = 2;
int h =0, t=0, gas=0;
void setup() {
  Serial.begin(9600);
  dht.begin();
  pinMode(mq2,INPUT);
  attachInterrupt(0, manejoPir, RISING);
  sensarMQ2();
  sensarDTH();
  Serial.print("m:");
  Serial.println(mq2);
  Serial.print("h:");
  Serial.println(h);
  Serial.print("t:");
  Serial.println(t);
}
void loop(){
  int temporalM = gas;
  int temporalH = h;
  int temporalT = t;
  if(digitalRead(pir==LOW)){
    Serial.println("npir");
  }
  sensarMQ2();
  sensarDTH();
  if(temporalM != gas){
    Serial.print("m:");
    Serial.println(gas);
  }
  if(temporalH != h){
    Serial.print("h:");
    Serial.println(h);
  }
  if(temporalT != t){
    Serial.print("t:");
    Serial.println(t);
  }
}
```

```

}
delay(2000);
}
void sensorMQ2(){
  gas = analogRead(mq2);
}
void sensorDTH() {
  h = dht.readHumidity();
  t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("DTH11 no detectado");
  }
}
void manejoPir(){
  Serial.println("pir");
}
Receptor Servidor NODE.js
var express = require('express');
var app = express();
var server = require('http').Server(app);
var io = require('socket.io')(server);
app.use(express.static("public"));
app.get('/', function(req, res){
  res.status(200).send("hola mundo");
});
const SerialPort = require('serialport');
const ReadLine = SerialPort.parsers.Readline;
const mySerial = new SerialPort("COM4", { //modificar puerto segun su equipo
  baudRate: 9600
});
const parser = mySerial.pipe(new ReadLine({ delimiter: '\n' }));
parser.on('open', function(){
  console.log('puerto abierto');
});
parser.on('data', function(data){
  var datos = data;
  console.log(datos.toString());
  if(data.charAt(0) == 'm'){
    io.emit('mq2',data.match(/(\d+)/g));
  }
  if(data.charAt(0) == 'h'){
    io.emit('humedad',data.match(/(\d+)/g));
  }
  if(data.charAt(0) == 't'){
    io.emit('temperatura',data.match(/(\d+)/g));
  }
  if(data.charAt(0) == 'p'){
    io.emit('pir',"detectando presencia");
  }
}

```

```

    }
    if(data.charAt(0) == 'n'){
        io.emit('npir',"no detectando presencia");
    }
    else {
        io.emit('error',"error en recepcion de datos");
    }
}

});
server.listen(8080, function(){
    console.log("running server");
});
mySerial.on('err', function(err){
    console.log(err.message);
});

```

Receptor

Servidor Node.js encargado de recibir los datos por medio de la librería Serial Port y emitirlos a un cliente html por medio de sockets.

```

var express = require('express');
var app = express();
var server = require('http').Server(app);
var io = require('socket.io')(server);
app.use(express.static("public"));

app.get('/', function(req, res){
    res.status(200).send("hola mundo");
});
const SerialPort = require('serialport');
const ReadLine = SerialPort.parsers.Readline;

const mySerial = new SerialPort("COM8", {
    baudRate: 9600
});

const parser = mySerial.pipe(new ReadLine({ delimiter: '\n' }));

if (!parser) {
    console.log("Leyendo...");
}
else console.log("No leyendo...");

```

```

parser.on('open', function(){
  console.log('puerto abierto');
});

parser.on('data', function(data){
  var datos = data;

  console.log(datos.toString());
  if(data.charAt(0) == 'm'){
    io.emit('mq2',data.match(/(\d+)/g));
  }
  if(data.charAt(0) == 'h'){
    io.emit('humedad',data.match(/(\d+)/g));
  }
  if(data.charAt(0) == 't'){
    io.emit('temperatura',data.match(/(\d+)/g));
  }
  if(data.charAt(0) == 'p'){
    io.emit('pir',data.match(/(\d+)/g));
  }

  else {
    io.emit('error',"error en recepcion de datos");
  }
});

server.listen(8080, function(){
  console.log("running server");
});

mySerial.on('err', function(err){
  console.log(err.message);
});

```

Cliente html

Encargado de graficar los datos de forma pública

<!DOCTYPE html>

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Practica 3</title>
  <script src="socket.io.js"></script>
</head>
<body>
  <h1 style="text-align: center">Practica 3</h1>
  <div id="datos" style="background:#DAF7A6; text-align: center"></div>

  <script>
    var humedad = 0;
    var temperatura = 0;
    var gas = 0;
    var pir = "no detectando presencia";

    function render(){
      var html= `<strong>PIR: ${pir}</strong><br>
        <strong>Temperatura: ${temperatura} °C</strong><br>
        <strong>Humedad: ${humedad} %</strong><br>
        <strong>MQ2: ${gas}</strong>`;
      document.getElementById('datos').innerHTML = html;
    }

    var socket = io.connect('http://localhost:8080', {'forceNew': true});
    socket.on('pir', function(data){//cuando recibe
      console.log(data);
      pir = data;
      render();
    });
    socket.on('npir', function(data){//cuando recibe
      console.log(data);
      pir = data;
      render();
    });

    socket.on('mq2', function(data){//cuando recibe
      console.log(data);
      gas = data;
      render();
    });

    socket.on('temperatura', function(data){//cuando recibe

```



```
console.log(data);  
temperatura = data;  
render();  
});  
socket.on('humedad', function(data){//cuando recibe  
console.log(data);  
humedad = data;  
render();  
});  
</script>  
  
</body>  
  
</html>
```

Comprobación

Pruebas

Pruebas con una distancia(mínima)de 0cm entre emisor y receptor.

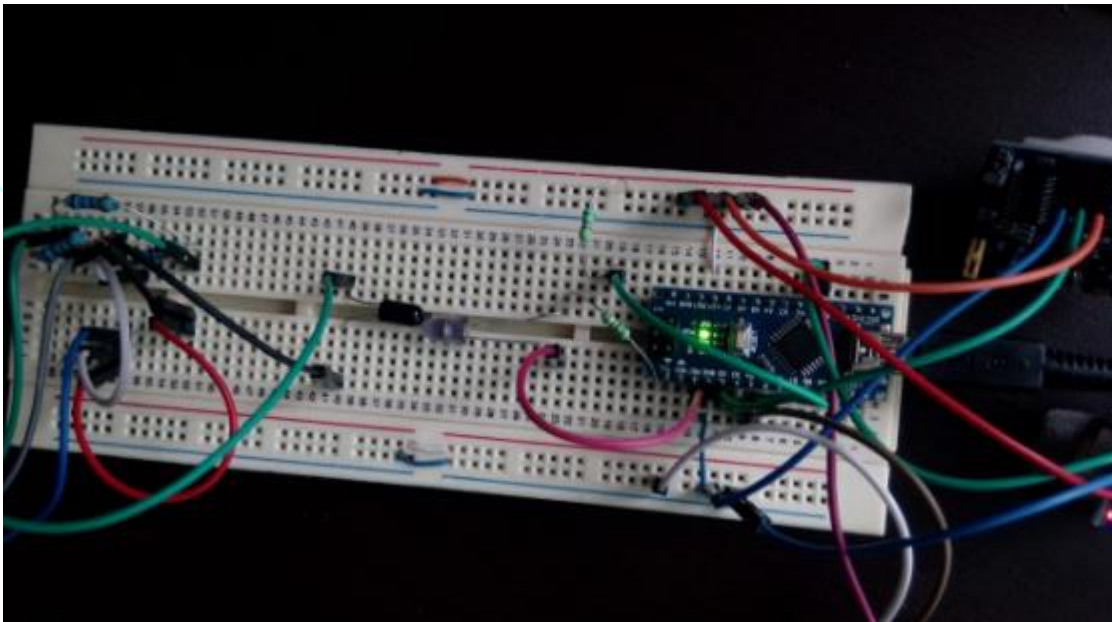


Ilustración 18 Prueba en distancia mínima

Prueba con una distancia de 6 metros (Máxima) entre emisor y receptor.

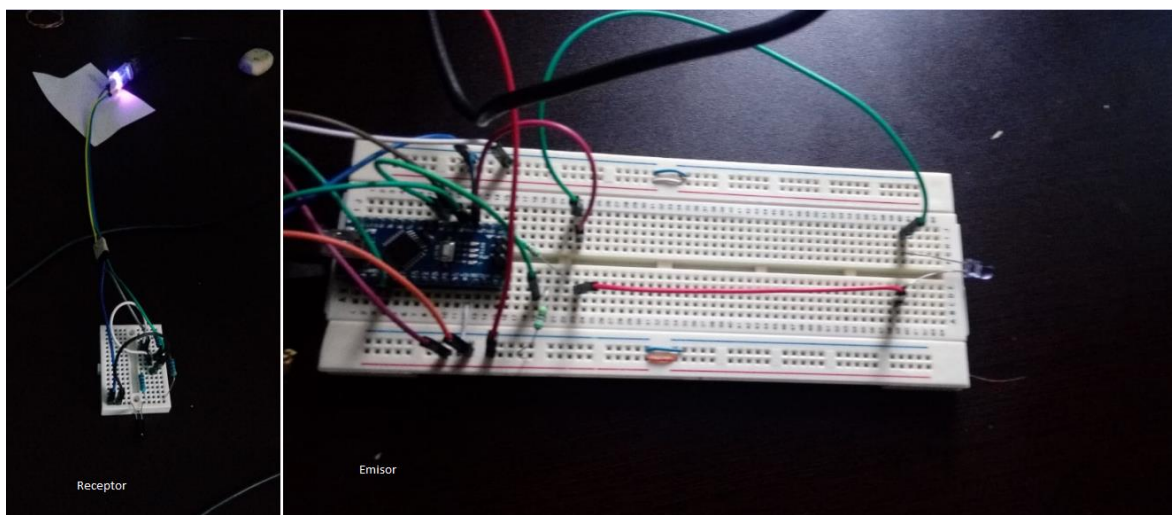


Ilustración 19 Prueba en distancia máxima

La distancia máxima alcanzada fue de 6.35 m entre emisor y receptor, al alejarse más se obtenían datos incompletos como lo muestra la siguiente imagen.

```

pir:0
t:37
pir:0
p:1
pir:0
m:11

```

Resultados

La siguiente gráfica representa el comportamiento del PIR, cada que detecta un movimiento se representa como 1 en el eje y, mientras que en x se toma la hora que fue la intervención.

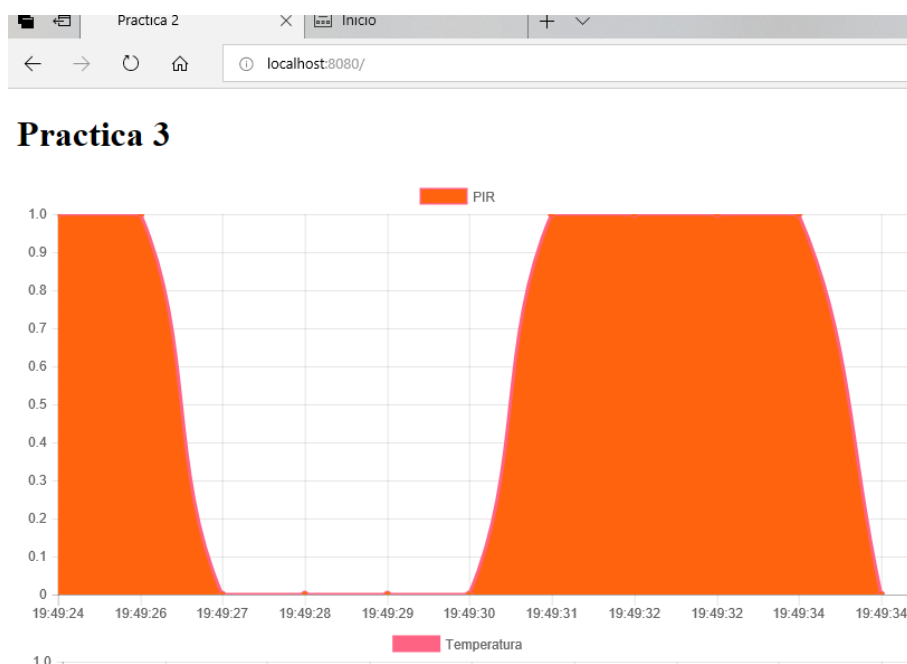


Ilustración 20 Grafica PIR

La siguiente gráfica muestra los resultados de la temperatura brindados por el Sensor DTH11.

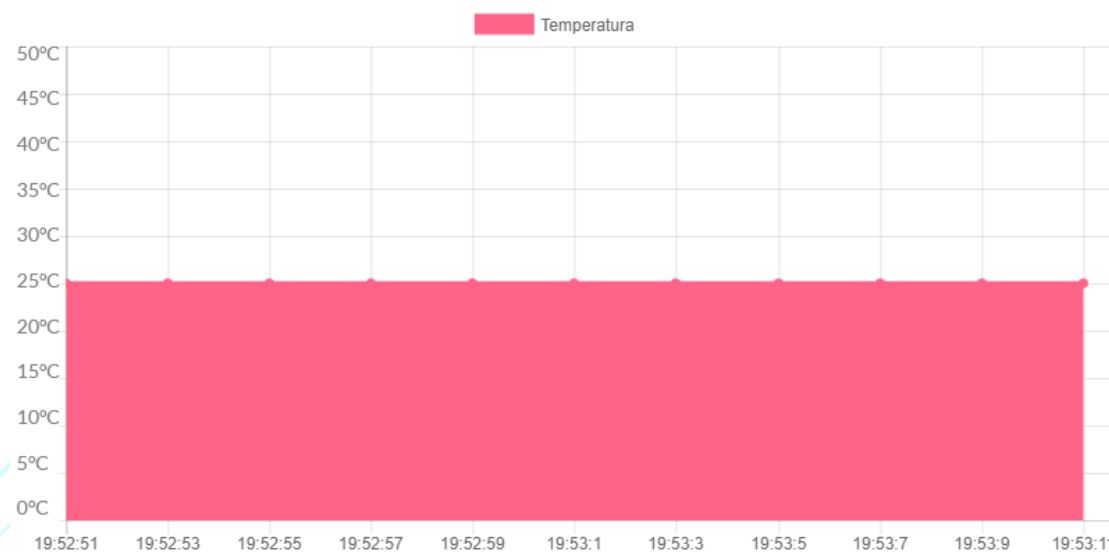


Ilustración 21 Gráfica de la temperatura

La siguiente gráfica muestra el porcentaje de humedad en tiempo real

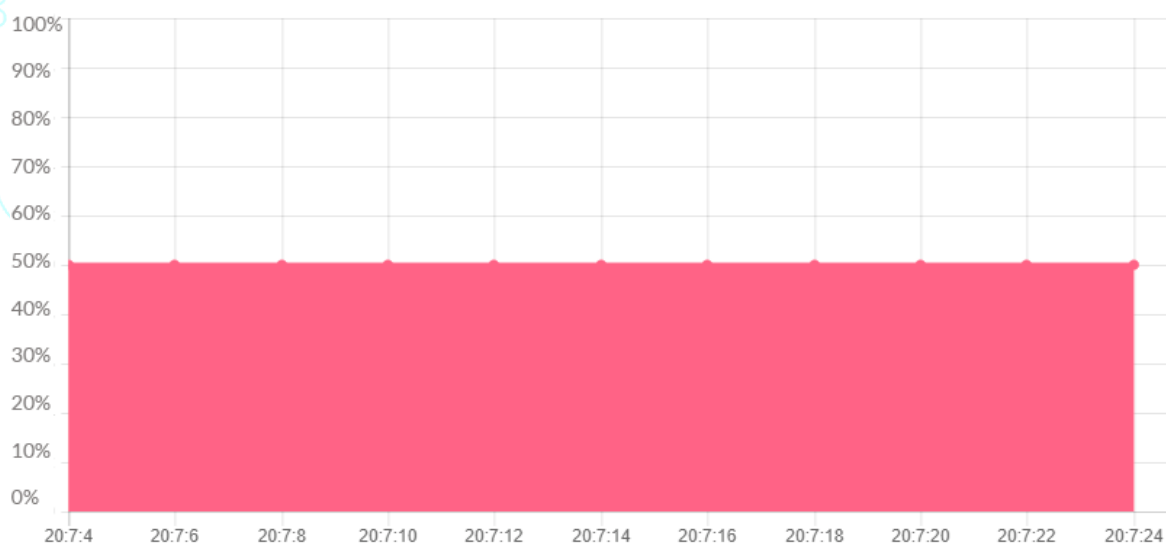


Ilustración 22 Gráfica de la Humedad

Resultados al detectar Gas con el sensor MQ-2, cuando el cuadro es de color verde, no se detecta gas, pero si cambia a rojo, significa que se está detectando Gas.

No detectando Gas




Detectando Gas





Ilustración 23 Detección de Gas

Mediciones



Tensión

Sensor	Valor esperado	Valor obtenido
MQ-2	5V	

PIR	5V	
DTH11	5.5V	

Corriente

Sensor	Valor Esperado	Valor Obtenido
MQ-2	No encontrado en la hoja de datos	

DHT11	60 μ A	
PIR	<1 mA	

Conclusiones

Después de las correcciones del PIR, nos dimos cuenta que antes del manejo de interrupciones estábamos enviando datos censados todo el tiempo y sin sentido, después de usar las interrupciones pudimos darnos cuenta que podemos optimizar recursos enviando solo los datos necesarios, para que el servidor sea optimizado.

Así también se hizo con los demás sensores comparando las medidas anteriores con las recién recolectadas, y cuando eran diferentes se enviaban, cuando eran iguales no se enviaban.

Por la parte de la comprobación, el voltaje que variaba entre cada sensor, lo mismo pasaba en con la corriente, pues en el caso del sensor DHT11 en la hoja de datos mencionaba que el consumo es menor de $60\mu\text{A}$, cuando en realidad el multímetro marcaba 1.4 mA , mucho más que el doble, en el sensor PIR es todo lo contrario pues los valores coinciden con la hoja de datos, pues los valores que nos brindó el multímetro era de 0.2 mA , el esperado era menor a 1 mA .