



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Colima
División de Estudios de Posgrado e
Investigación

Tecnológico nacional de México campus
Colima

Maestría en Sistemas Computacionales

Tecnologías de Internet

Emisor y Receptor: Controlador de leds con
resistencias

PRESENTA:

BRANDON JESÚS SANTANA GUDIÑO | G2146006

LUIS MANUEL SALTOS PINEDA | G2246017

DOCENTE:

PATRICIA ELIZABETH FIGUEROA MILLÁN

VILLA DE ÁLVAREZ, COLIMA. NOVIEMBRE 2022.





Emisor y Receptor: Controlador de focos con resistencias

Objetivo.

Que el alumno aprenda, conozca, aplique y realice un circuito en el cual mediante el uso de microcontroladores se pueda realizar la comunicación de un led con una fotorresistencia. Además de hacer el circuito electrónico, el alumno también deberá aplicar los conocimientos adquiridos en las prácticas anteriores para realizar un servidor y uno o varios clientes que funcionen con las librerías utilizadas durante el transcurso de la materia, tales como socket, threading, os, entre otras.

Metodología.

Para el desarrollo de esta práctica se realiza una investigación documental entre las diferentes plataformas, foros y documentación oficial de los microcontroladores, igualmente se hace la investigación sobre la documentación de Python, la cual nos ayuda a comprender mejor cómo funciona la comunicación entre los microcontroladores y el código para desarrollar el circuito electrónico. Una vez investigado el funcionamiento y aplicando los conocimientos adquiridos durante la materia se procede a realizar el circuito electrónico y se hacen las pruebas necesarias para obtener los objetivos planteados.

Materiales.

Para la realización de la práctica se hizo uso de los siguientes materiales:

- Computadora con acceso a internet
- Editor de texto (Visual Studio Code)
- Arduino UNO





- Resistencias eléctricas
- Protoboards
- Fotorresistencias
- Leds

Desarrollo.

A continuación, se describe una pequeña descripción sobre la creación del circuito electrónico, de igual manera se explicará brevemente el funcionamiento de los scripts de Python que realizan la función de servidor y cliente para este circuito electrónico.

En la imagen 1 se puede observar el código fuente del emisor del circuito electrónico cabe mencionar que este código está desarrollado en el lenguaje de Python, lo primero que podemos observar en la imagen 1 es el llamado a las librerías que se utilizarán, posteriormente se declara el comando con el cual el Arduino iniciará, para esto se pone el número de puerto y la velocidad con la que se comunicará. Después se pasan los valores que se mandarán por el socket, continuando con la codificación del mensaje, una vez se tiene lo anterior se crea el socket y se configura el TTL con la opción IP MULTICAST TTL con la finalidad de definir que se utilizara el protocolo IP. Por último, se hace la impresión de los datos que se mandan al receptor y se llama al sistema para enviar los datos por el socket.



```
import serial, time
import socket
import struct

#Comando para abrir puerto
arduino = serial.Serial('COM5', 9600)
time.sleep(2)
#arduino.write(b'1')

#ESTE ES EL VALOR QUE TENGO QUE ENVIAR POR EL SOCKET
valor = arduino.readline()
variable = str(valor)
message = variable[2:5]
print(message)
arduino.close()

# Codifica el mensaje a bytes
bytes_mesaage = str.encode(message)

#multicast_addr = '224.0.0.1'
#port = 10000
grupo_multicast = ('224.0.0.1', 10000)

# Crea el socket de tipo datagrama
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
# TTL (Time To Live), el valor por defecto es 1, significa que el paquete
# no sera redireccionado por el router mas allá del segmento de red actual.
# el valor puede ir de 1 a 255.
ttl = struct.pack('b', 1)
# El socket se configura con ese valor de tiempo de vida para los mensajes.
# El TTL controla cuantas redes (saltos) se realizarán para recibir el mensaje
# la llamada al sistema setsockopt() pasa información al kernel
# Se configura el TTL con la opción IP_MULTICAST_TTL, y se indica que se utilizará el protocolo IP
sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, ttl)

print("Enviando datos a los receptores: ", message)
# Llamada al sistema para enviar los datos por el socket
sock.sendto(bytes_mesaage, grupo_multicast)
sock.close()
```

Imagen 1 - Código emisor

La imagen 2 corresponde al receptor del circuito electrónico, al igual que la imagen anterior se puede ver que el código está desarrollado en el lenguaje de programación Python, la estructura de este script es similar a la anterior pero con las siguientes diferencias primero se define el canal por el cual se hace la comunicación, además de la IP y el puerto, también se tienen las instrucciones para crear el socket y permitir que sea multi-clientes, se crean los protocolos necesarios para una correcta comunicación y por último se capturan y se imprimen los valores que manda el emisor, en este proceso se define el puerto al que está conectado

el microcontrolador y la velocidad de comunicación. Por último, se coloca una comparativa en la cual se compara la cantidad de intensidad de luz obtenida y si es mayor a 200 el led se colocará en estado de encendido de lo contrario permanecerá apagado.

```
# PROGRAMA RECEPTOR EN UNA COMUNICACIÓN MULTICAST
import serial, time
import socket
import struct

multicast_addr = '224.0.0.1'
bind_addr = '0.0.0.0'
port = 10000

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
membership = socket.inet_aton(multicast_addr) + socket.inet_aton(bind_addr)

sock.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, membership)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

sock.bind((bind_addr, port))

while True:
    message, address = sock.recvfrom(255)
    recibido = str(message)
    valor = int(recibido[2:5])
    print(valor)

    arduino = serial.Serial('COM4', 9600)
    time.sleep(2)

    if valor >= 200 :
        arduino.write(b'1')
        print("encendido")
        arduino.close()

    else :
        arduino.write(b'2')
        print("apagado")
        arduino.close()
```

Imagen 2 - Código receptor

En las imágenes 3 y 4 se pueden observar los códigos en el lenguaje de programación C++ los cuales nos sirven para hacer la comunicación entre los microcontroladores y el lenguaje de programación de Python, en estos scripts básicamente se hace mención de las salidas y entradas de información, así como los pines a los cuales se enlaza la comunicación, su estado y las condicionantes que se tienen que cumplir en base a la información que reciba.

```
1  int pin = A5;
2  int valor;
3
4  void setup() {
5      // put your setup code here, to run once:
6      Serial.begin(9600);
7      pinMode(pin, INPUT);
8  }
9
10 void loop() {
11     // put your main code here, to run repeatedly:
12     valor = analogRead(pin);
13     Serial.print(valor);
14     delay(100);
15 }
16
```

Imagen 3 - Código emisor C++

```
const int pinLED = 9;
int estado=0;

void setup()
{
  Serial.begin(9600);
  pinMode(pinLED, OUTPUT);
}

void loop()
{
  char option = Serial.read();
  if (option == 'n' && estado==0)
  {
    digitalWrite(pinLED, HIGH);
    delay(100);
    estado=1;
  }
  if (option == 'd' && estado==1)
  {
    digitalWrite(pinLED, LOW);
    delay(100);
    estado=0;
  }
}
```

Imagen 4 - Código receptor C++

Para la parte física de la práctica se realizan dos circuitos electrónicos, en el primero se pone en led con una resistencia los cuales van conectados a uno de los microcontroladores, en este circuito se podrá observar la salida de los parámetros obtenidos de la fotorresistencia. En el segundo circuito se tiene una fotorresistencia conectada a una resistencia las cuales van conectadas al segundo microcontrolador, este circuito se encargará de capturar y mandar los valores de la intensidad de la luz que se tiene y en caso de ser cierto valor se activará o desactiva el led, en la imagen 5 se puede observar los dos circuitos electrónicos que se utilizaron para esta práctica.

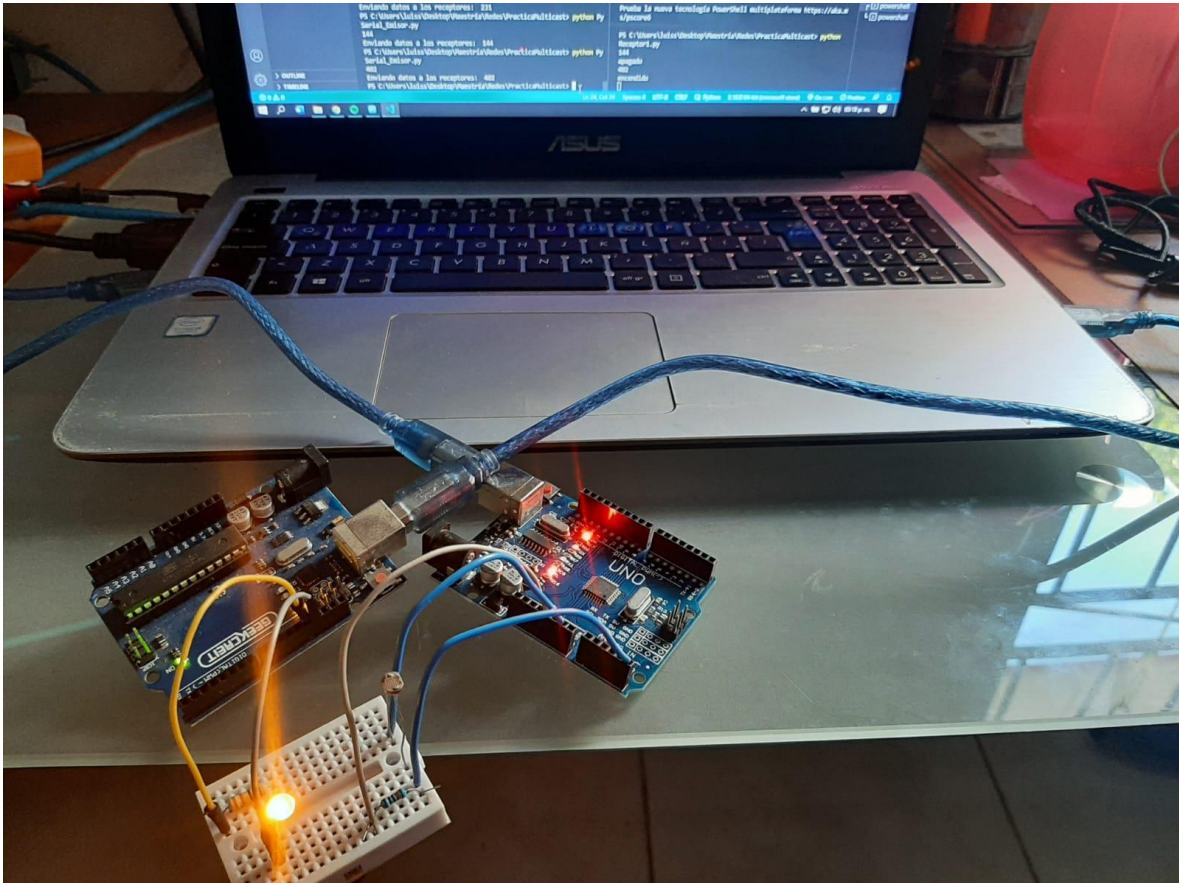


Imagen 5 - Circuito electrónico

Conclusión.

Con la creación de servidores y clientes mediante sockets se pudo hacer la conexión a microcontroladores en los cuales se pudo crear un circuito capaz de hacer que mediante el uso de una fotorresistencia se pudiera encender o apagar un led en diferentes computadores mediante la conexión de internet. Además de eso se aprendió sobre el uso de nuevas librerías de Python que hicieron que esta práctica fuera posible, también se comprende mejor como hacer la conexión entre un microcontrolador con el lenguaje de programación de Python lo cual al principio de la práctica se desconocía, de igual manera se empieza a comprender mejor cómo es el



funcionamiento del ya tan famoso internet de las cosas y cómo es que este es posible.

