



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO[®]

Tecnológico Nacional de México

Instituto Tecnológico de Colima

Materia:

Tecnologías de Internet

Tema:

Reporte de práctica: comunicación multicast con Arduino

Docente:

D en C. Patricia Elizabeth Figueroa Millán

Estudiantes:

Roberto Tapia Torres

Jesús Emmanuel Martínez Torres

Villa de Álvarez, Col. 18 noviembre de 2022

Reporte de Práctica: Comunicación multicast con Arduino

1. Objetivo

Implementar un sistema de conexión entre dos placas de Arduino a través de un servidor multicast. Se encenderá un led en un Arduino. Con la señal que el otro le mandará con la información obtenida de una fotorresistencia

2. Metodología

Primero se programó en Arduino el programa para leer los valores la fotorresistencia. Se hizo la conexión correspondiente del módulo al Arduino y se ajustaron los parámetros de las lecturas que nos daba.

Se programó el otro Arduino para que encendiera un led con un pulso en uno de sus conectores de salida.

En Python se programó el código para el trasmisor de datos, al cual se le adaptó la lectura del puerto serial, de esta manera comenzaría a leer los parámetros entregados por la fotorresistencia

En otro archivo de Python se programó el receptor, al cual se le habilitó la lectura en el puerto serial, este programa es el que se comunicaría con el Arduino que tiene el LED

3. Materiales

Para llevar a cabo esta práctica fue necesario el uso de los siguientes elementos:

- Computadora con Windows 10 de 64 bits

- IDE Arduino 2.0.1
- Python 3 (librerías “socket”, “serial”, “time”, “struct”)
- Editor de texto (Atom 1.60.0 x64)
- 1 Tarjeta Arduino Uno
- Tarjeta Arduino Yún
- Módulo foto resistor keyes KY-018
- Resistencia de 330 ohms
- Diodo LED
- Jumpers
- 2 protoboards

4. Desarrollo

Para la elaboración de la práctica fueron necesarios aprender algunos conceptos, así como adaptarse a los entornos de desarrollo que fueron utilizados, así como lo fue el multicast, Arduino, Python, Atom, el módulo keyes KY-018 y en el caso de Arduino una librería extra la cual tuvo que ser instalada, llamada pySerial.

Multicast: El tráfico IP Multicast, o también conocido como multidifusión IP, es un método para transmitir información a un grupo de receptores (clientes) que están configurados para tal fin. Los equipos que no están configurados específicamente, no recibirán este tráfico de red y podrán dedicarse a enviar y recibir otro tipo de tráfico. [1]

Módulo KY-018: Este es un módulo utilizado para medir la intensidad de la luz. La resistencia decrecerá en presencia de la luz y decrecerá con la ausencia de la misma. La salida es análoga y esta determina la intensidad de la luz. [2]

Arduino: Arduino es una placa electrónica con un microcontrolador programable, es de código abierto lo cual permite que toda la gente alrededor del mundo tenga acceso fácil a tecnologías avanzadas, que les permiten interactuar con el mundo físico. [3]

Python: Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo. [4]

pySerial: Esta es una librería utilizada para Python con la particularidad de poder crear conexiones seriales, así como lo son UTP y UDP

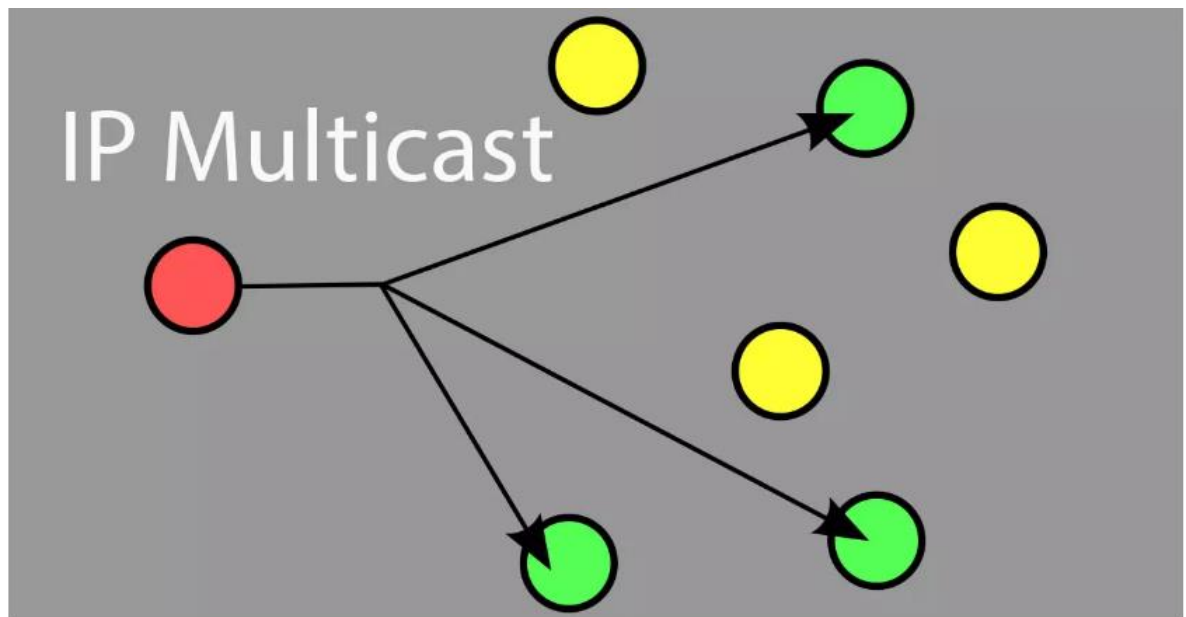


Figura 1. Ip multicast

El desarrollo de la práctica consistió en los siguientes pasos:

1. EL primer paso fue basarse en un diagrama de conexión para el modulo foto resistor keyes KY-018 obtenidos desde la pagina de módulos de Arduino. De esta manera se hicieron las conexiones necesarias

Connect the Power line (middle) and ground (-) to +5 and GND respectively. Connect signal (S) to pin A2 on the Arduino.

KY-018	Arduino
S	Pin A2
middle	+5V
-	GND

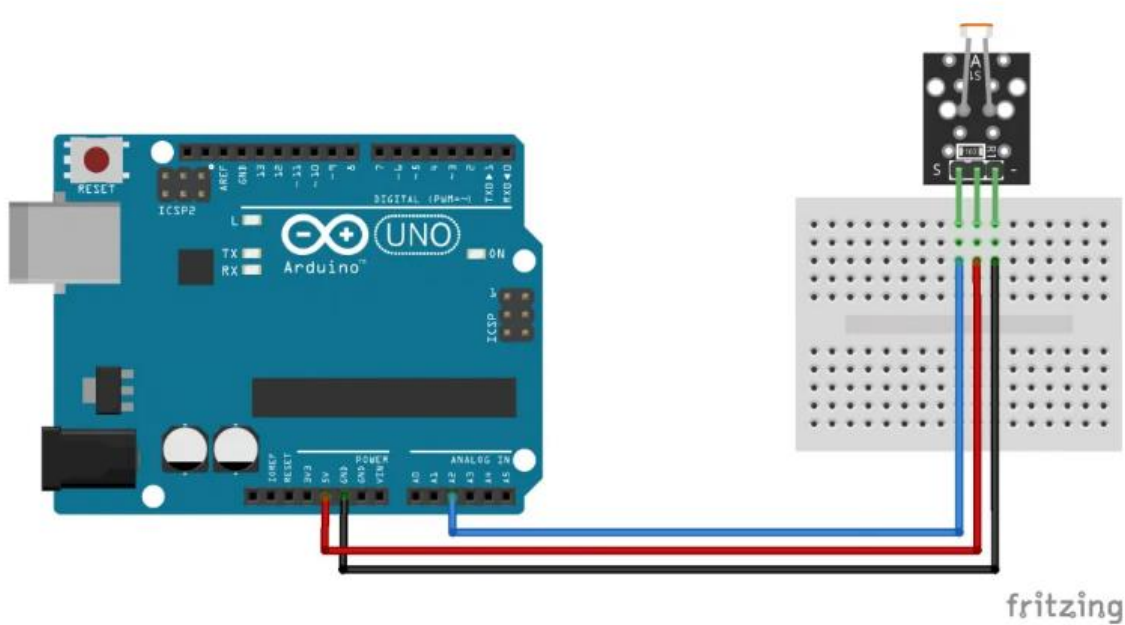


Figura 2. Diagrama conexión módulo KY-018 con Arduino

2. Se comenzó a hacer la programación para la IDE de Arduino el cual sería utilizado solo para la lectura de valores del foto resistor de nuestro modulo, los valores de esta manera serían impresos en el puerto serial de nuestro dispositivo

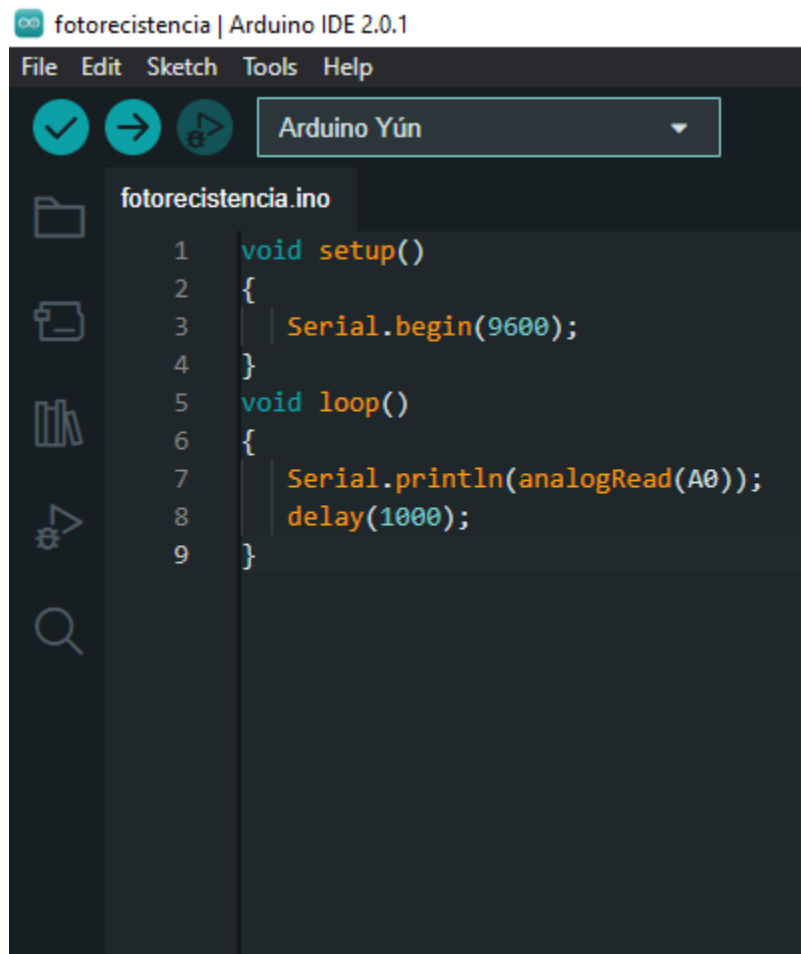


Figura 3. Código Arduino lectura foto resistor

3. En la siguiente parte se comenzó con el programa el Python el cual nos serviría para la trasmisión de los datos con nuestro otro Arduino y para la lectura del valor serial del Arduino con la foto resistor, para este paso era necesario una librería llamada pySerial, por lo cual se instalo previamente utilizando el comando “pip install pySerial” en la ventana de cmd de Windows.

En esta sección se determinó que la conexión sería del tipo UTP y de manera multihilo, de esta manera podríamos hacer un grupo multicast para enviar la

información a varios dispositivos que estén en el mismo grupo al que deseamos mandarlos. En esta parte del programa determinamos los parámetros que elegiremos arbitrariamente según las lecturas que son leídas del puerto serial, siendo que a partir de una menor a 400 se enviará una señal y arriba de esta cifra se mandará otra hacia el programa receptor.

```
import socket, serial, time
group = '224.0.0.1'
port = 5004
arduino=serial.Serial("COM6",9600)
time.sleep(1)
# 2-hop restriction in network
ttl = 2
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM,socket.IPPROTO_UDP)
sock.setsockopt(socket.IPPROTO_IP,socket.IP_MULTICAST_TTL,ttl)
while True:
    valor=int(arduino.readline().decode("ascii"))
    if(valor<400):
        message="P"
        print("es de noche")
        sock.sendto(message.encode("utf-8"), (group, port))
    else:
        print("es de día")
        message="N"
        sock.sendto("N".encode("utf-8"), (group, port))
```

Figura 4. Código python emisor

4. Por otra parte, comenzaremos con el siguiente programa para Arduino el cual será para recibir los comandos de Python por su puerto serial, y este terminará teniendo la respuesta del receptor el cual terminará por encender o apagar un LED según sea la luminosidad recolectada en el ambiente por la fotorresistencia del que está conectada al primer Arduino. En este mismo paso conectaremos el Arduino con el led y la resistencia de 330ohms.


```
encender led serial.ino
1  int option;
2
3  void setup(){
4      Serial.begin(9600);
5      pinMode(led1, OUTPUT);
6  }
7
8  void loop(){
9      if (Serial.available() > 0){
10         option = Serial.read();
11         Serial.println(option);
12
13         if(option == 'P'){
14             digitalWrite(13, HIGH);
15         }
16
17         if(option == 'N'){
18             digitalWrite(13, LOW);
19         }
20     }
21 }
```

Figura 5. Control de la luz LED en Arduino

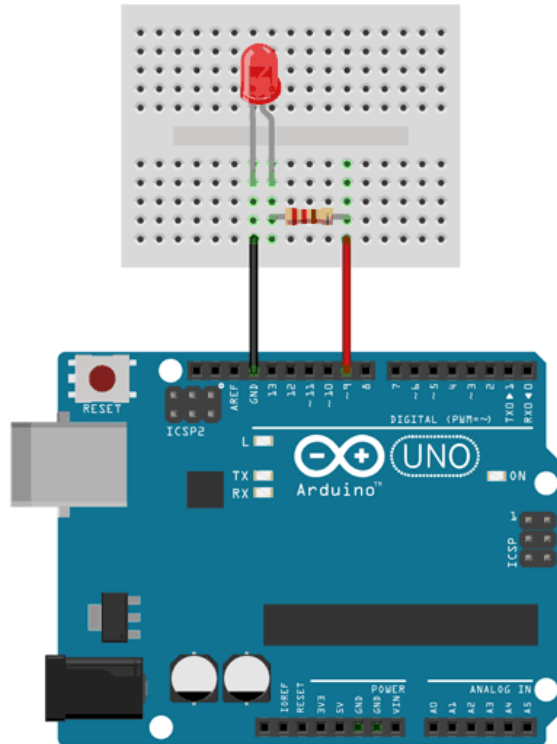


Figura 6. Diagrama de conexión Arduino a LED.

5. En la siguiente parte se programará en Python el receptor de la comunicación, el cual tiene el trabajo de recibir los datos, decodificarlos y después comunicarlos por el puerto serial al Arduino previamente programado para el control del LED. Debido a que la conexión es multicast podrá haber múltiples receptores y todos obtendrán el mismo mensaje enviado por el emisor.

```

import socket, serial, time
import struct
ser = serial.Serial("COM5", 9600, timeout=1)

MCAST_GRP = '224.0.0.1'
MCAST_PORT = 5004

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, socket.IPPROTO_UDP)
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

sock.bind(('', MCAST_PORT))
mreq = struct.pack("4sl", socket.inet_aton(MCAST_GRP), socket.INADDR_ANY)

sock.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, mreq)

while True:
    print(sock.recv(255).decode("utf-8"))
    ser.write(sock.recv(255))

```

Figura 7. Receptor multicast en Python

5. Resultados

Los resultados fueron favorables, se podía ver todo el flujo de mensajes desde el primer Arduino, hacia el programa emisor en Python, después al programa receptor de Python, por último, al programa cargado en el segunda Arduino de esta manera podíamos observar perfectamente como al obstruir la luz ambiental el LED se encendía mientras que si dejábamos pasar la luz ambiental el LED se apagaba

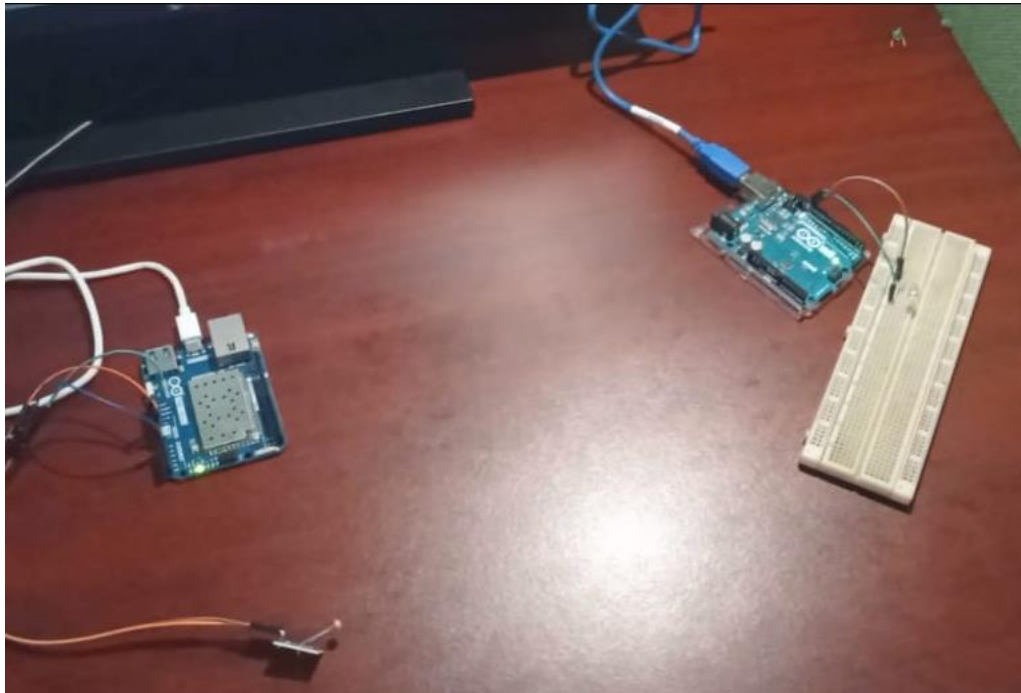


Figura 8. Prueba sin obstrucción de luz ambiental

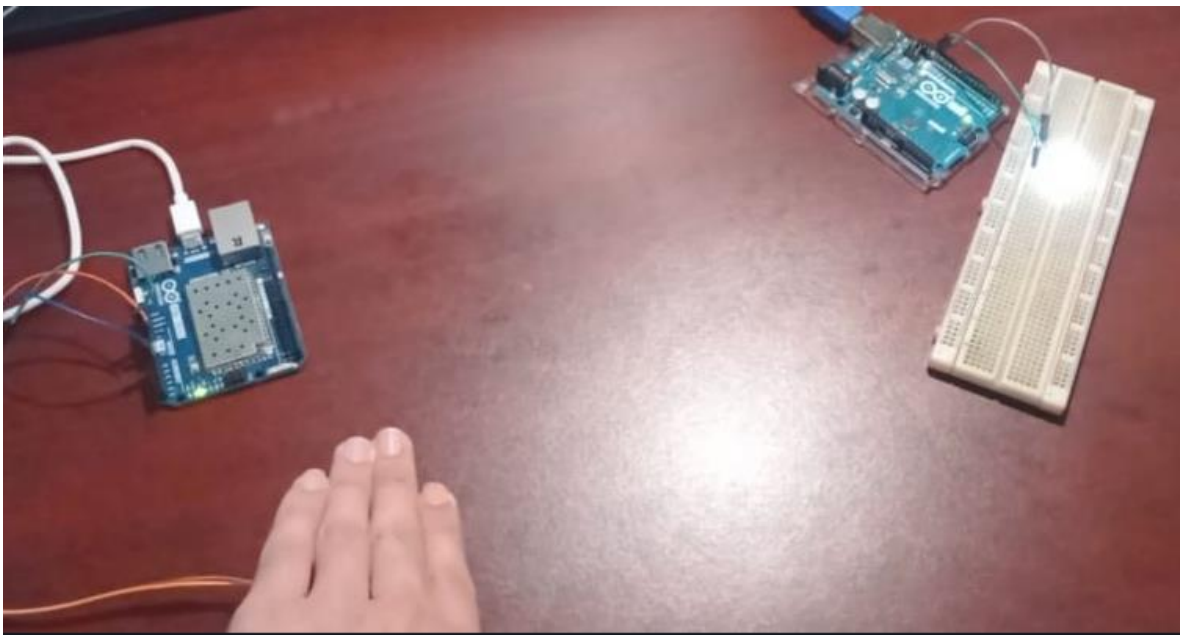


Figura 9. Prueba con obstrucción de luz ambiental

6. Conclusiones

La programación en Arduino siempre puede ser muy útil para el comportamiento de sensores y actuadores. Este tipo de programas siempre suele hacerse en una misma tarjeta limitando la distancia a la que este puede funcionar, Aplicar un servidor para la comunicación con múltiples tarjetas puede presentar una gran ventaja en campos como son la domótica debido a que un mismo emisor le comunicará a todos los receptores las condiciones ambientales a las que nos encontramos con un único sensor que capte estas variables, y expandiendo los límites de la señal a los límites que tenga una misma red local.

7. Bibliografía

[1] «Qué es el tráfico IP Multicast, cómo funciona y aplicaciones que lo usan», RedesZone. <https://www.redeszone.net/tutoriales/internet/que-es-trafico-multicast/> (accedido 18 de noviembre de 2022).

[2] ArduinoModules, «KY-018 Photoresistor Module», *ArduinoModulesInfo*, 16 de febrero de 2017. <https://arduinomodules.info/ky-018-photoresistor-module/> (accedido 18 de noviembre de 2022).

[3] «About Arduino». <https://www.arduino.cc/en/about> (accedido 18 de noviembre de 2022).

[4] «Welcome to Python.org», *Python.org*. <https://www.python.org/> (accedido 18 de noviembre de 2022).

[5] «pyserial: Python Serial Port Extension». Accedido: 18 de noviembre de 2022. [MacOS :: MacOS X, Microsoft :: Windows, POSIX]. Disponible en: <https://github.com/pyserial/pyserial>