

UNIVERSIDAD DE ORIENTE
NÚCLEO ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN
LABORATORIO DE PROYECTOS DIGITALES AVANZADOS



Implementación Protocolo de comunicación UART con múltiples raspberry pi pico

Prof. Rene Cabrera

Elaborado por:

Brazon Eulises

Barcelona, 22 de febrero del 2023

ABSTRACCION

El presente informe está basado en un informe hecho anteriormente donde se puede encontrar más información referente, denominado “*Implementación Protocolo de comunicación UART y Almacenamiento de Información*”, que se puede conseguir a través de del siguiente enlace: https://github.com/EulisesBrazon/sensors_and_display_UART. En él se logra establece una comunicación 1 a 1 entre dos raspberry pi pico, a continuación, se utilizará esta base para establecer comunicación entre distintos raspberry pi pico.

OBJETIVOS

Objetivo General:

- Entender el funcionamiento del protocolo UART

Objetivos Específicos:

- Establecer la comunicación con distritos raspberry pi pico a través de un protocolo de comunicación serial.

MATERIALES

- Potenciómetro (en este caso se usó uno de $10K\Omega$ / analógico).
- LM35 (sensor de temperatura / analógico).
- Fotorresistencia (sensor de luz / analógico).
- HC-SR04 (sensor de distancia por ultrasonido / digital).
- Pantalla LCD TFT 1,8 pulgadas chip de control ST7735, de 128X160px.
- Cuatro Raspberry pi pico.
- Resistencia $1K\Omega$ (1unidad) y 220Ω (2unidades).
- Transistor 2N2222.
- Protoboard (dos mediano como mínimo, 4 recomendable).
- Jumpers para la conexión.
- Dos tiras led tira led rgb modelo Ws2812.
- Alimentación 5V.
- Herramientas para la manipulación y testeo (multímetro, pinzas, entre otros).

DESCRIPCION DEL PROYECTO

Muchas veces el protocolo de comunicación **UART** es visto como un protocolo con el que se puede establecer una comunicación 1 a 1, si bien esto es cierto, con la correcta identificación de la información que se está transmitiendo, es posible entablar una comunicación entre distintos raspberry pi pico, expandiendo así las posibilidades de transmisión de información, a través de una misma línea de comunicación.

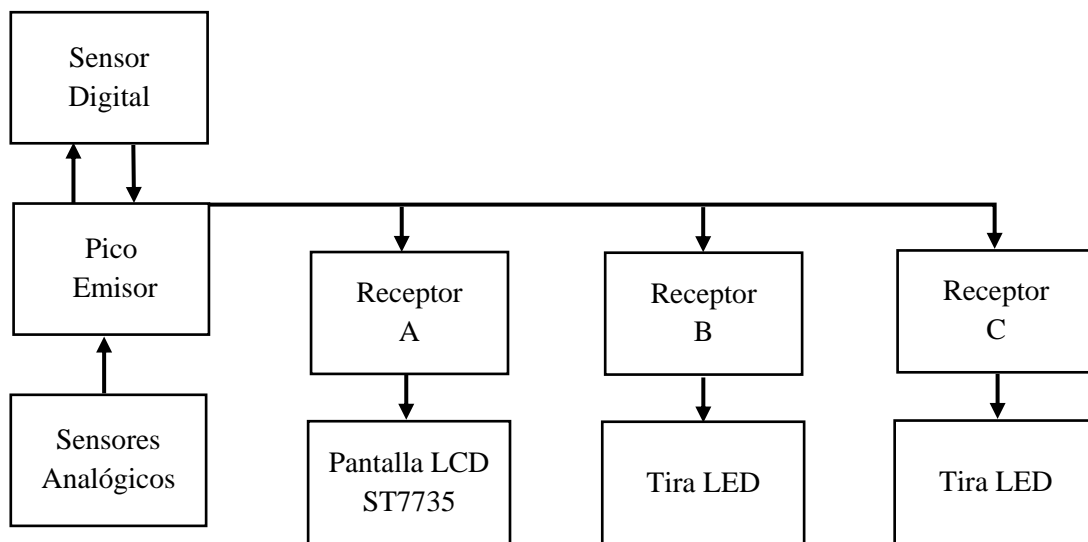
Diagrama de bloque: En el presente proyecto se tomó un enfoque en el que uno de los raspberry pi pico cuenta con las lecturas de distintos sensores, estas lecturas reciben su procesamiento adecuado, y son enviados a distintos raspberry pi pico, cada uno con su respectivo actuador que varía su comportamiento en función de estas lecturas.

El raspberry pi pico con los sensores lo denominaremos **pico emisor**, ya que se encargará de enviar la información a través del pin Tx.

El raspberry pi pico **receptor A**, Cuenta con una pequeña pantalla, donde se estarán visualizando las lecturas de los sensores percibido a través del pin Rx.

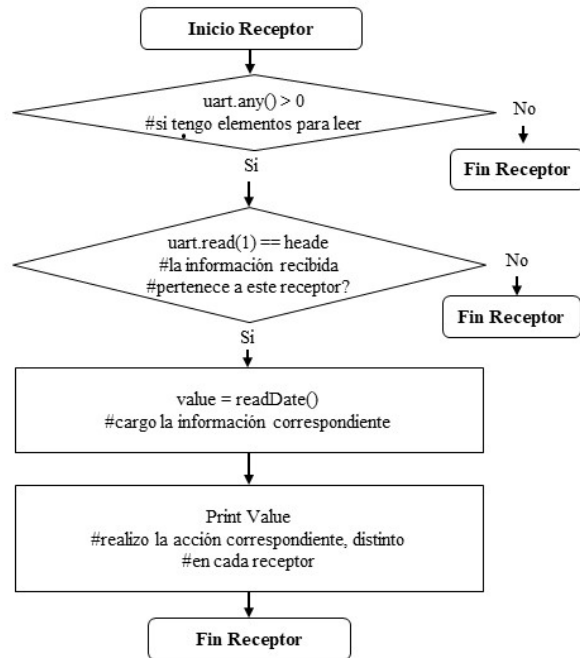
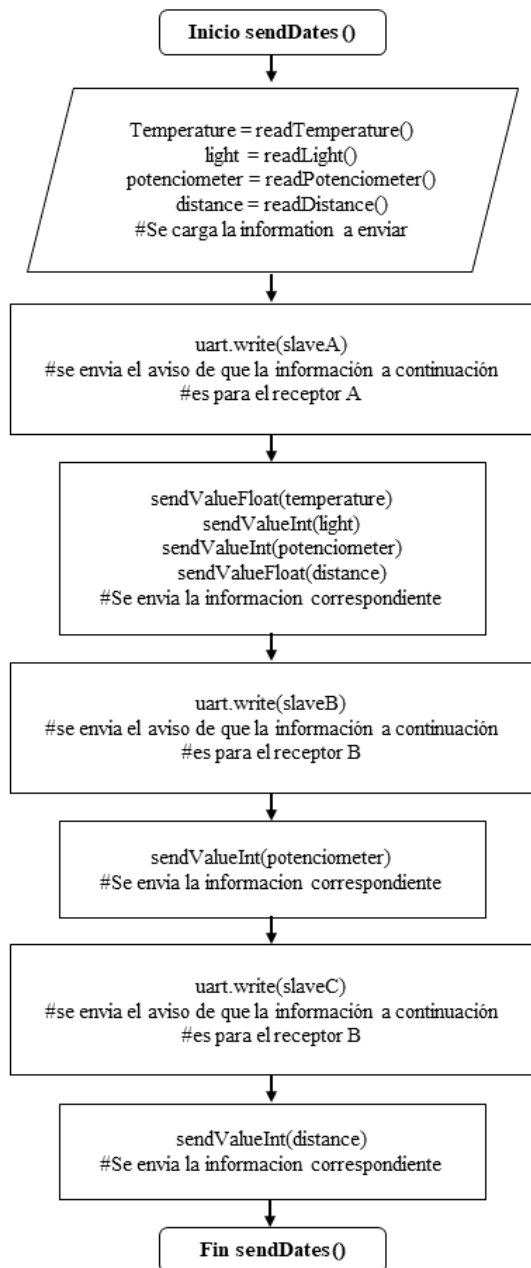
El raspberry pi pico **receptor B**, Cuenta con una pequeña tira led (controlada con la librería neopixel), el cual estará variando el nivel de intensidad, dependiendo del valor percibido por el potenciómetro, esta lectura se obtendrá a través del pin Rx.

El raspberry pi pico **receptor C**, es similar al receptor B, con la diferencia de que la variación de la intensidad de la tira led, dependerá del valor percibido, que corresponde al sensor de distancia.



Nota: Si bien en este proyecto la comunicación se produce en un único sentido, es posible establecer una comunicación entre cualquiera de estos dispositivos, haciendo una conexión circular, donde la salida del primer pico se conecta a la entrada del siguiente y así sucesivamente hasta llegar a la salida del último pico que se conecta a la entrada del primer pico. Todo depende del propósito del proyecto.

Diagramas De Flujo:



Circuito Electrónico

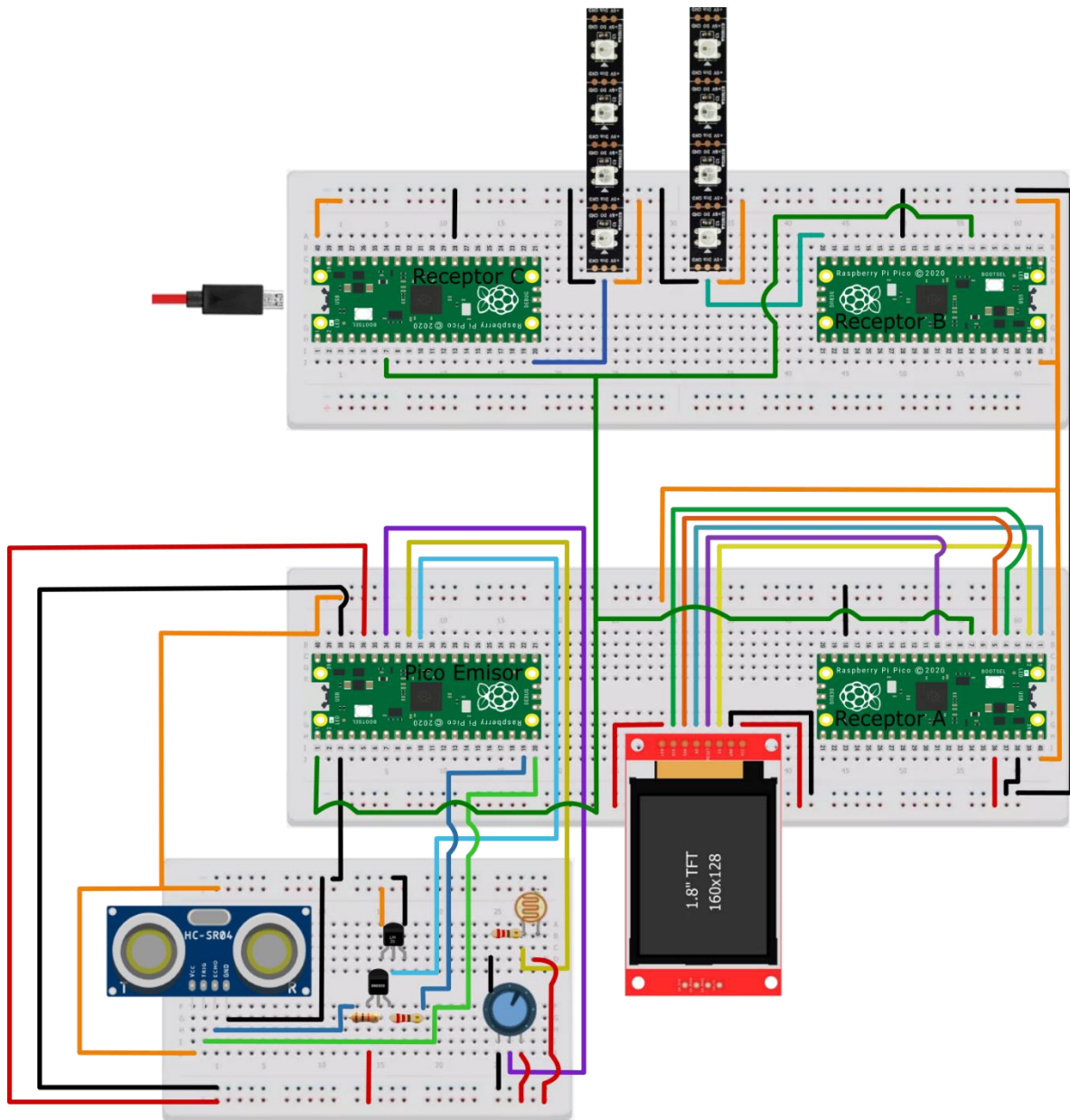
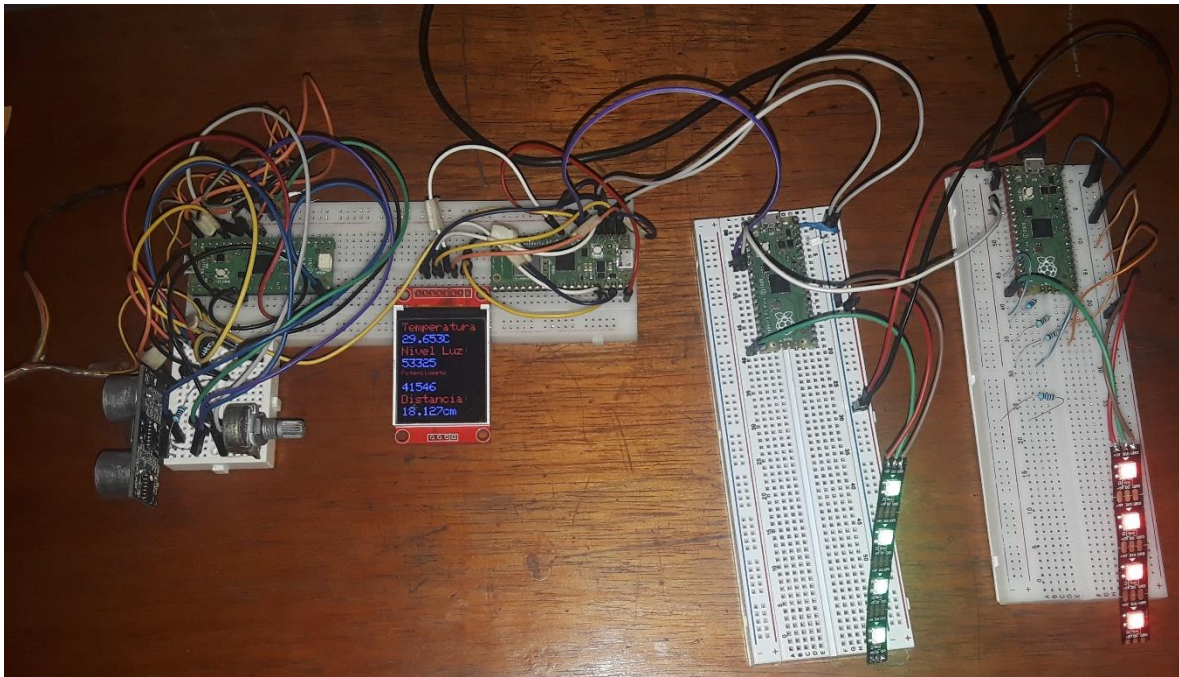


Imagen del circuito



Modificaciones al código de la practica anterior:

Se establecen direcciones diferentes

```
11 slaveA = b'0xA'  
12 slaveB = b'0xB'  
13 slaveC = b'0xC'
```

Se agregaron envíos adicionales de información

```
66 def sendDates():  
67     #loading values  
68     temperature=readTemperature()  
69     light = readLight()  
70     potenciometer = readPotenciometer()  
71     distance = readDistance()  
72  
73     #sending values  
74     uart.write(slaveA)  
75     sendValueFloat(temperature)  
76     sendValueInt(light)  
77     sendValueInt(potenciometer)  
78     sendValueFloat(distance)  
79  
80     uart.write(slaveB)  
81     sendValueInt(potenciometer)  
82  
83     uart.write(slaveC)  
84     sendValueInt(distance)
```

Código alojado en:

https://github.com/EulisesBrazon/sensors_and_display_UART/tree/four_raspberry

Referencia utilizadas:

Python Documentation. “*class UART – duplex serial communication bus*”.

Recuperado de: <https://docs.micropython.org/en/latest/library/machine.UART.html>