

“MANEJO DE COMUNICACIÓN SERIAL UART y SPI EN ARDUINO”

Informe N°8

Laboratorio de Sistemas Embebidos

Melanny Dávila

Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
melanny.davila@epn.edu.ec

2nd Jonathan Álvarez

Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
jonathan.alvarez@epn.edu.ec

Abstract—En el siguiente documento, se presenta el fundamento teórico de los protocolos de comunicación UART y SPI. Presentando así sus principales diferencias con el fin de poner analizar que protocolo es conveniente utilizar en diferentes aplicaciones.

Index Terms—Arduino, UART, SPI, asincrónico.

I. INTRODUCCIÓN

La comunicación serial entre dos dispositivos se realiza a través del intercambio de una secuencia de bits, donde se transmite bit a bit, uno por vez, donde, aunque es lenta la comunicación, tiene la ventaja de poder ser transmitida a mayores distancias y utilizar menos líneas de comunicación.

La comunicación serial entre dos dispositivos únicamente utiliza 3 líneas las cuales son:

- Línea de recepción de datos (RX)
- Línea de transmisión de datos (TX)
- Línea común (GND)

Donde GND, RX y TX del microcontrolador Arduino son fácilmente identificables en la placa. El TX y RX de Arduino son los dos pines que emplea el dispositivo para realizar la comunicación por medio del protocolo serial. Los datos, por lo tanto son transmitidos en la línea o pin TX y son recibidos por la línea o pin RX [1].

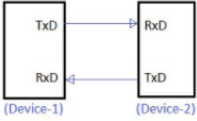
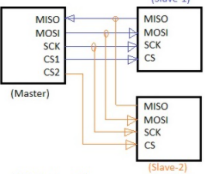
II. OBJETIVOS

- Relacionar al estudiante con el uso y manejo de comunicación serial y SPI de la placa de desarrollo Arduino Uno.
- Identificar diferencias entre comunicación serial UART y SPI en Arduino [2].

III. CUESTIONARIO

A. Establecer un Tabla comparativa entre los dos protocolos de comunicación UART y SPI.

TABLA I
PRINCIPALES DIFERENCIAS ENTRE LOS PROTOCOLOS UART Y SPI

Parámetro	UART	SPI
Acronimo	Universal Asynchronous Receiver/Transmitter	Serial Peripheral Interface
Diagrama de Interfaz		
Designación de pines	TxD: Transmisor RxD: Receptor	SCLK: reloj serial MOSI: entrada esclava, salida maestra MISO: entrada maestra, salida esclava SS: selección de esclavo
Velocidad de datos	La velocidad de datos entre dos dispositivos debe tener el mismo valor.	La velocidad máxima admitida está entre 230kbps y 460kbps
Alcance mínimo	Hasta 50 pies [3]	Más de 50 pies
Tipo de comunicación	Asincrónica	Sincrónica
Número de maestros	No aplica	Uno [4]
Reloj	No hace uso de ninguna señal de reloj común. Cada dispositivo usa una señal de reloj independiente.	Existe una señal de reloj común entre el maestro y el esclavo.
Protocolo	Hace uso de un bit de inicio y uno de parada para 8 bits de datos.	Cada fabricante tiene sus propias especificaciones acerca de los protocolos de comunicación con los periféricos.
Direccionamiento por software	Dado que es una conexión uno a uno entre dos dispositivos, no es necesario el uso de direccionamiento.	Las líneas de selección de esclavos se utilizan para direccionar cualquier esclavo conectado a el maestro.
Ventajas	*Comunicación simple. *Muy popular. *Casi todos los dispositivos presentan soporte UART con conector de 9 pines, conocido como interfaz RS232 [3].	*Simple y no requiere gran procesamiento. *Comunicación full duplex. *Alcance de mayor velocidad de datos. *Uso de menor energía en comparación con I2C.
Desventajas	*Adecuado únicamente para comunicación entre solo dos dispositivos. *Si la velocidad de transmisión no se fija previamente entre los dispositivos existirá distorsión.	*A mayor número de esclavos, aumenta la complejidad de hardware, mayor número de pines usados [4]. *Al agregar nuevos dispositivos, se debe agregar líneas CS lo que genera cambios en software *No existe control de flujo.

B. Escribir ventajas y desventajas del uso de los protocolos de comunicación UART y SPI.

Algunas diferencias a parte de las mencionadas en la tabla 1, son las siguientes:

- **UART**

- Ventajas:
 - * Facilidad en la búsqueda de documentación.
 - * Uso de tres pines (Rx, Tx y GND) para realizar su proceso de comunicación.
 - * Implementación de detección de errores en la información enviada [3].
- Desventajas:
 - * Su alcance es bajo.
 - * Tamaño de trama limitado [3].
 - * No se puede usar más de un maestro y un esclavo.
 - * Uso de bits para sincronización, por lo que es menos eficiente que SPI [4].

- **SPI**

- Ventajas:
 - * Alcance de altas velocidades de transmisión.
 - * Mayor tasa de transmisión de datos, comunicación full duplex [4].
 - * No hace uso de bits para sincronización por lo que es más eficiente en comparación a UART.
- Desventajas:
 - * Uso de al menos cuatro pines.
 - * Congestión del canal dado que no existe control de flujo.
 - * Protocolo no confiable debido a que no realiza detección de errores [3].

C. Consultar y describir detalladamente el uso de tres aplicaciones que permitan la interconexión de dispositivos por medio de los protocolos UART y SPI.

- **Sensor de presión barométrica:** El sensor de presión barométrica BMP180 está diseñado para leer la presión atmosférica y de esta forma estimar indirectamente la Altura sobre el nivel del mar. La presión atmosférica también varía con el clima, principalmente con la temperatura, pues esta hace cambiar la densidad del aire, que se ve reflejado en un cambio en el peso y por consiguiente en un cambio de presión. Lo que mide el sensor SCP1000 es la presión absoluta (Barométrica) y la temperatura, al censar la temperatura podemos compensar su influencia en la presión y así determinar con mayor exactitud la altitud. El sensor SCP1000 lee los dos bytes de la temperatura. Una vez que tiene la temperatura, lee la presión en dos partes. Primero lee los tres bits más altos, luego los 16 bits más bajos. Combina estos dos en un solo entero largo desplazando los bits altos y luego usando un OR bit a bit para combinarlos con los 16 bits inferiores.

Elementos:

- Placa Arduino UNO

- Sensor SCP1000
- Protoboard
- Cables para protoboard

- **Enfriador de alimentos con movimiento:** Se montará un módulo Bluetooth HC-05 en la parte delantera de la plataforma para un mejor alcance. El resto de los componentes, incluido un controlador de motor L298N, GPS PAM-7Q y brújula HMC6883L, se montaron en el interior y se conectaron al Arduino a través del protoboard. El módulo bluetooth se conectará por medio del protocolo UART a la placa Arduino. Los motores se alimentaron perforando orificios a través de la plataforma y uniéndolos la placa del controlador del motor L298N. El LiPo 3s se conectó como la fuente de alimentación principal y a tierra junto con la fuente de alimentación Arduino para la lógica de control. Elementos:

- Arduino UNO
- Integrado L298N
- Brújula HMC6883L
- Enfriador
- Protoboard
- 2 Motores DC
- LiPo 3s
- Módulo Bluetooth HC-05
- Madera
- Ruedas

- **Control de un potenciómetro:** AD5206 es un potenciómetro digital de 6 canales. Esto significa que tiene seis resistencias variables (potenciómetros) incorporadas para control electrónico individual. Existen tres pines en el chip para cada una de las seis resistencias variables internas, y se pueden conectar de la misma forma que usaría un potenciómetro mecánico.

Elementos:

- Arduino UNO
- Potenciómetro digital AD5206
- 6 LEDs
- 6 resistencias de 220 Ohmios
- Protoboard

Los potenciómetros digitales son útiles cuando necesita variar la resistencia en un circuito electrónicamente en lugar de hacerlo manualmente. Las aplicaciones de ejemplo incluyen atenuación de LED, acondicionamiento de señales de audio y generación de tonos.

D. Realizar un programa en Tinkercad para control inteligente de las luces de una casa. Se utilizará la comunicación SPI para conectar 2 placas de desarrollo Arduino. La primera placa se instalará en un lugar visible de la casa y funcionará como estación de control por lo cual se conectará un sensor infrarrojo, el cual mediante un control remoto permitirá realizar diferentes acciones en el sistema. La segunda placa Arduino estará incrustada en la pared y tendrá conectado las luces de cocina, comedor, sala y estudio. El sistema tendrá los siguientes modos de funcionamiento.

TABLA II
PARÁMETROS DE DISEÑO

Modo	Botón presionado	Luces encendidas
Fiesta	1	Cocina, Comedor, Sala
Cena	2	Cocina, Comedor
Trabajo	3	Estudio
Entretenimiento	4	Sala

```

1 //Circuito utilizando sensor IR y control remoto
2 #include <SPI.h> //Libreria SPI
3 #include <IRremote.h> //Libreria sensor IR
4 //Pin de salida al rele
5 int Pin_Rx = 9; //Definimos el pin de recepcion
6 IRrecv irrecv(Pin_Rx); //Ingresamos el pin para la
   recepcion
7 decode_results results;
8
9 void setup()
10 {
11   Serial.begin(9600);
12   //pinMode(releSala, OUTPUT);
13   //pinMode(releComedor, OUTPUT);
14   //pinMode(releCuarto, OUTPUT);
15   //pinMode(releCocina, OUTPUT);
16   SPI.begin(); //Iniciamos la conexion SPI
17   SPI.beginTransaction(SPI_Settings(14000000,MSBFIRST,
   SPI_MODE0));
18   digitalWrite(SS, HIGH); //Ponemos como defecto al
   esclavo
19   Serial.println("Habilitando IRin");
20   irrecv.enableIRIn(); //Activa la recepcion IR
21   Serial.println("Disponible IRin");
22 }
23 void loop()
24 {
25   byte Mastersend, Mastereceive;
26   if (irrecv.decode(&results)) { //El valor de la tecla
   presionada se lamacena para ser procesada
27   unsigned int value = results.value; //Variable sin
   signo para testear
28   Serial.println(value);
29   switch(value){
30     //Dependiendo del boton enviara el modo al
   arduino esclavo
31     case 2295:
32       Serial.println("Envia");
33       digitalWrite(SS, LOW);
34       SPI.transfer('1');
35       break;
36     case 34935:
37       digitalWrite(SS, LOW);
38       SPI.transfer('2');
39       break;
40     case 18615:
41       digitalWrite(SS, LOW);
42       SPI.transfer('3');
43       break;
44     case 10455:
45       digitalWrite(SS, LOW);
46       SPI.transfer('4');
47       break;
48   }
49   digitalWrite(SS, HIGH);
50   irrecv.resume(); // Recive el siguiente dato del
   control remoto
51 }
52 delay(250);
53 }

```

Script 1. Código fuente para la implementación SPI de un sistema con control remoto para el Maestro

```

1 //Placa esclava
2 //Controla el encendido de los modos solicitados
3 #include<SPI.h>
4 volatile bool received;
5 volatile char Slaverceived;
6 char dato = '0';
7 int Cocina = 4, Comedor = 5, Sala = 6, Estudio = 7;
8 void setup()
9 {
10   Serial.begin(9600);
11   pinMode(MISO, OUTPUT);
12   SPCR |= bit(SPE);
13   received = false;
14   SPI.attachInterrupt();
15 }
16 ISR (SPI_STC_vect){
17   byte Slaverceived = SPDR;
18   received = true;
19   Serial.println("Interrupcion");
20 }
21 void encenderModo(char caso){
22   switch(caso){
23     //Se escoge los modos predeterminados con el
   caracter recibido
24     case '1':
25       //Modo 1
26       digitalWrite(Cocina, HIGH);
27       digitalWrite(Comedor, HIGH);
28       digitalWrite(Sala, HIGH);
29       digitalWrite(Estudio, LOW);
30       break;
31     //Modo 2
32     case '2':
33       digitalWrite(Cocina, HIGH);
34       digitalWrite(Comedor, HIGH);
35       digitalWrite(Sala, LOW);
36       digitalWrite(Estudio, LOW);
37       break;
38     //Modo 3
39     case '3':
40       digitalWrite(Cocina, LOW);
41       digitalWrite(Comedor, LOW);
42       digitalWrite(Sala, LOW);
43       digitalWrite(Estudio, HIGH);
44       break;
45     //Modo 4
46     case '4':
47       digitalWrite(Cocina, LOW);
48       digitalWrite(Comedor, LOW);
49       digitalWrite(Sala, HIGH);
50       digitalWrite(Estudio, HIGH);
51       break;
52   }
53 }
54 void loop(){
55   if(received){
56     received = false;
57     Serial.println("Detecta");
58     Serial.println(Slaverceived);
59     encenderModo(Slaverceived);
60   }
61   delay(200);
62 }
63

```

Script 2. Código fuente para la implementación SPI de un sistema con control remoto para el Esclavo

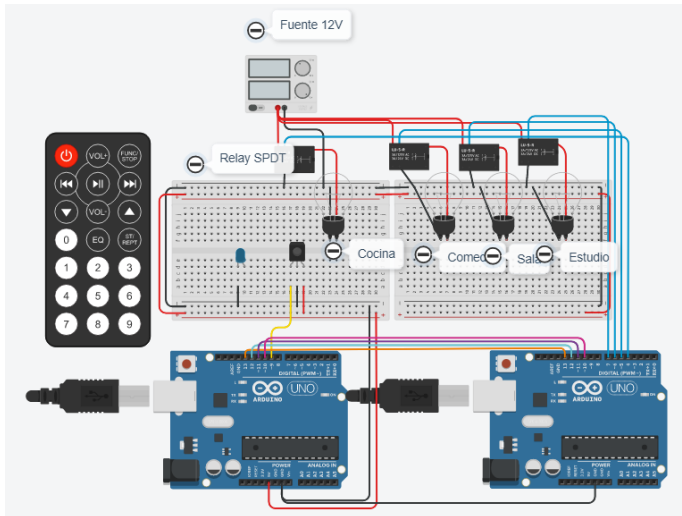


Fig. 1. Circuito de control de modos de iluminación con SPI en arduino.

E. Conclusiones:

Jonathan Álvarez

- Los protocolos de comunicación son de gran importancia ya que nos permiten comunicarnos entre dispositivos que tengan los protocolos y poder intercambiar información o agregar mayor cantidad de funcionalidad.
- Es posible controlar usando un dispositivo teclados y pantallas con otros los sensores y establecer comunicaciones entre ellos superando la limitación de pines que pueden tener estos aparatos.
- La elección del protocolo adecuado dependerá de la aplicación que estemos utilizando, ya que si deseamos controlar varios dispositivos los protocolos Maestro Esclavo serán los apropiados.

Melanny Dávila

- La implementación del protocolo de comunicación UART permite trabajar de una manera adecuada ciertas aplicaciones dado que el número de pines utilizados es bastante bajo, permitiendo así utilizar la diferencia de pines de la placa Arduino Uno el control de otros dispositivos como sensores, bobinas, bocinas, entre otros.
- El uso del protocolo de comunicación SPI limita la cantidad de pines disponibles, sin embargo, dado que se utilizan pines para sincronismo, el envío de información es controlado y de esta forma se evita saturar al receptor.
- Dado que UART utiliza bits de paridad para el envío de información, se garantiza que se pueda realizar corrección de errores en las tramas de datos en recepción; permitiendo así trabajar con un protocolo confiable el cual brinda seguridad en la integridad de los datos del usuario.

F. Recomendaciones:

Jonathan Álvarez

- Revisar la adecuada conexión entre las placas arduino para los protocolos teniendo en cuenta que la tierra se debe conectar entre los Arduinos

- Tener tiempos de retraso para que las placas logren comunicarse y no enviar al mismo tiempo por el canal, o para que la otra placa procese la información.

Melanny Dávila

- Considerar las características de cada uno de los protocolos previo a su implementación ya que de esta manera se podrá decidir cual de ellos se ajusta mejor a la aplicación y sobretodo considerando sus limitaciones.
- Es importante verificar que exista conexión entre los dos microcontroladores para ello se recomienda el uso del monitor serial ya que en él se puede imprimir los datos transmitidos y recibidos.
- El uso de un display LCD permite la realización de aplicaciones mejor diseñadas y mediante el uso de su librería se puede imprimir caracteres de una manera más sencilla en comparación al barrido de displays. Así mismo, se puede imprimir más caracteres en comparación a los 16 únicos caracteres que permite mostrar un display de 7 segmentos.

REFERENCES

- [1] "Comunicación Serial con Arduino - [enero, 2021]", Control Automático Educación, jun. 15, 2019. <https://controlautomaticoeducacion.com/arduino/comunicacion-serial-con-arduino/> (accedido feb. 03, 2021).
- [2] E. Espinosa, E. Tatayo, "MANEJO DE COMUNICACIÓN SERIAL UART y SPI EN ARDUINO". C.P. SISTEMAS EMBEBIDOS, Accedido: feb. 03, 2020. [En línea].
- [3] "UART vs SPI vs I2C — Difference between UART,SPI and I2C". <https://www.rfwireless-world.com/Terminology/UART-vs-SPI-vs-I2C.html> (accedido feb. 03, 2021).
- [4] "Understanding the Difference Between UART vs SPI", Total Phase Blog, jun. 29, 2016. <https://www.totalphase.com/blog/2016/06/spi-vs-uart-similarities-differences/> (accedido feb. 03, 2021).