



**Universidad Nacional Autónoma de
México
Facultad de Ingeniería**



Semestre 2023-2

Laboratorio de Microcomputadoras

**Proyecto 4:
Comunicación Serie Asíncrona.**

Profesor:

M.I. Ruben Anaya García

Alumnos:

Número de cuenta:

Barreiro Valdez Alejandro

317520888

Gil Márquez Arath Emiliano

317083875

Herrera Carrillo Cristhian

317094662

Zepeda Baeza Jessica

317520747

Grupo Teoría: 1

Fecha de realización: 24 de mayo de 2023

Fecha de entrega: 29 de mayo de 2023

Introducción.

La comunicación serie asíncrona es un método de transmisión de datos en el que la información se envía bit a bit, uno después del otro, a través de un canal de comunicación. A diferencia de la comunicación serie síncrona, en la que se utiliza un reloj común para sincronizar la transmisión de datos, en la comunicación serie asíncrona cada bit se envía con su propio temporizador individual.

En una comunicación serie asíncrona, cada trama de datos se compone de bits individuales, que incluyen bits de inicio, datos y bits de parada. El bit de inicio es utilizado para señalar el comienzo de la trama, seguido por los bits de datos que contienen la información que se está transmitiendo. Los bits de parada se utilizan para indicar el final de la trama. Este formato de trama facilita la detección de errores y la sincronización del receptor.

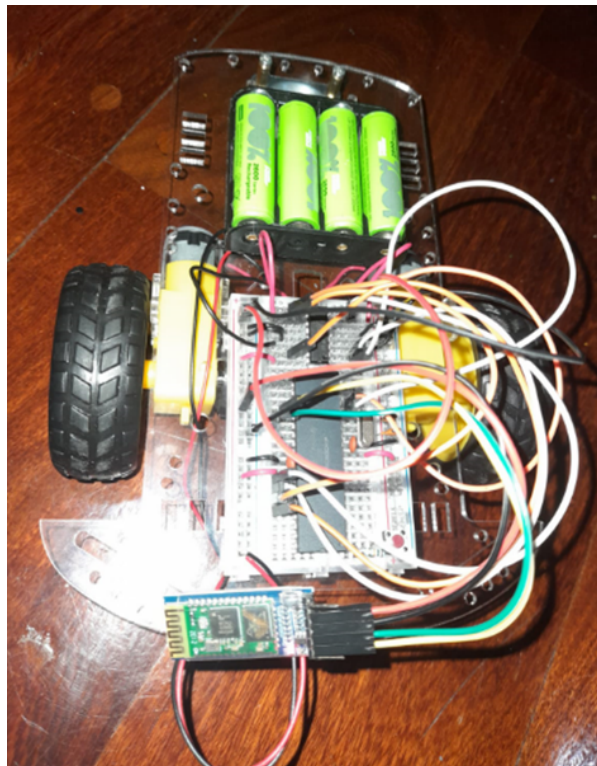
La comunicación serie asíncrona no requiere un reloj común entre el transmisor y el receptor, lo que permite la comunicación entre dispositivos con velocidades de reloj diferentes. Además, es relativamente simple de implementar y se utiliza en una amplia variedad de aplicaciones, desde la transmisión de datos en serie a través de cables o puertos serie hasta la comunicación con periféricos como módems, impresoras, sensores y dispositivos de adquisición de datos.

Sin embargo, también tiene algunas limitaciones. Debido a que no hay una sincronización constante entre el transmisor y el receptor, puede haber errores de sincronización si los dispositivos no están configurados correctamente. Además, la velocidad de transmisión es generalmente más lenta en comparación con otros métodos de comunicación serie, lo que puede ser una limitación en aplicaciones que requieren una alta velocidad de transferencia de datos.

Desarrollo

Para la realización del proyecto 4 se implementó el microcontrolador PIC16F877A en su sistema mínimo ya que será el cerebro del robot y se encargará de procesar las señales y enviar las instrucciones a los motores. Estos motores serán responsables de la tracción del robot, deben ser compatibles con el microcontrolador y capaces de mover el robot con ayuda de unas ruedas adecuadas al tamaño y peso, que serán controladas a lo especificado en el código cargado en

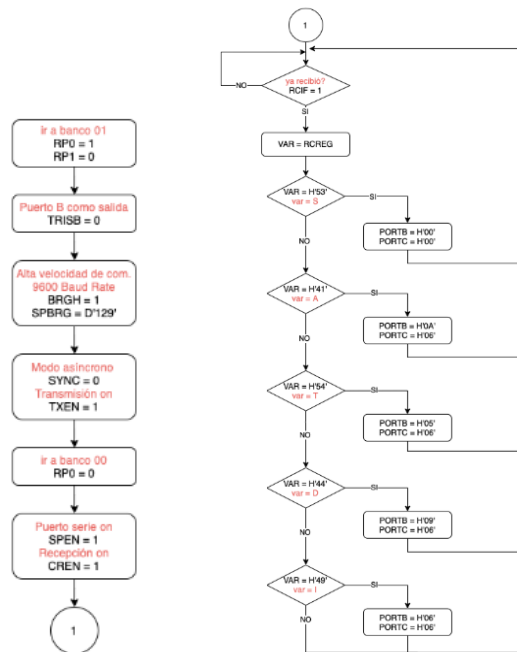
el microcontrolador, donde dependiendo de los comandos enviados debe realizar los movimientos de avanzar hacia adelante, desplazamiento hacia atrás, girar a la derecha, girar a la izquierda y parar. Para la transmisión de señales es requisito utilizar un módulo Bluetooth para permitir la comunicación inalámbrica entre el robot y la aplicación móvil, el cual va colocado a los pines TX (transmisión) y RX (recepción) del microcontrolador, que fue configurado como receptor de los comandos. Es necesario utilizar baterías adecuadas para alimentar el sistema de manera eficiente (microcontrolador, motores y los demás componentes).



Finalmente tenemos un robot móvil controlado mediante una aplicación móvil a través de Bluetooth, del cual podemos enviar comandos desde la aplicación móvil y el robot responderá realizando los movimientos correspondientes.

Algoritmos

Para este proyecto se realizó la siguiente carta ASM:

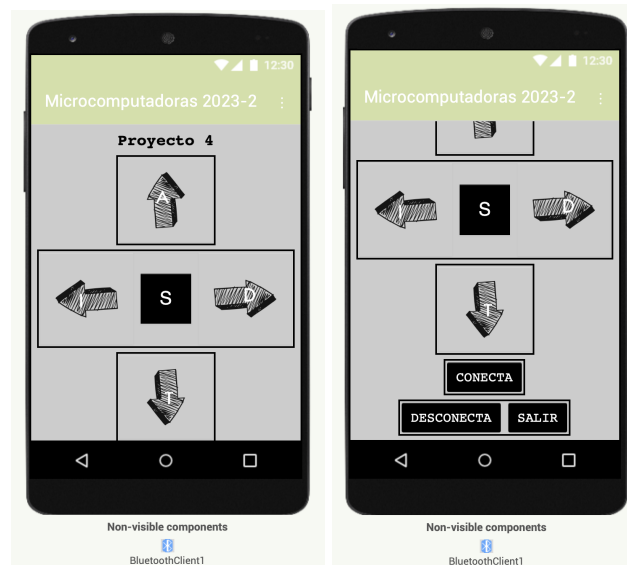


Donde primero se hacen configuraciones iniciales para definir el los puertos B y C como salidas y habilitar la recepción de datos. En el banco 1, se define una tasa de alta velocidad con de 9600 de Baud Rate, se activa el modo asíncrono y se habilita la transmisión. En el banco 0, se habilita el puerto serie y la recepción de datos.

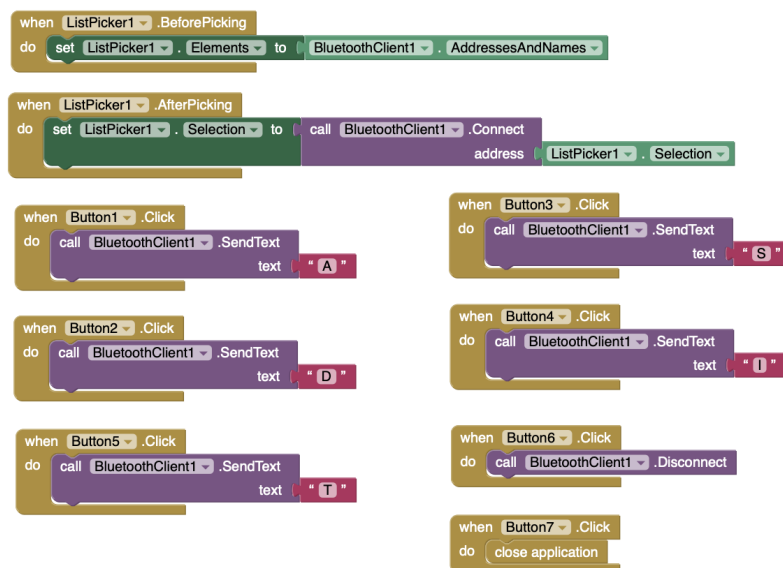
El programa después espera la recepción de un dato, verificando la bandera de recepción completa PCIF. Una vez que la bandera es 1, guarda el dato recibido en una variable VAR. Después se compara el valor de la variable con el ASCII de 'S','A','T','I','D'. En caso de no ser ninguno de estos caracteres, vuelve a esperar a recibir un nuevo dato. En caso de ser alguna de las letras antes mencionadas, se modifica el Puerto B y C para controlar los motores y regresa a recibir un nuevo dato. En caso de que la variable sea 'S', el Puerto B y C toman el valor de 0 para que los motores no se muevan. En caso de que sea 'A', el Puerto B es A y Puerto C es 6 para hacer que los motores vayan a la misma dirección. En caso de que sea 'T', el Puerto B es 5 y Puerto C es 6 para que los motores vayan en la misma dirección pero al revés de 'A'. En caso de que sea 'I', el Puerto B y Puerto C son 6 para hacer que gire a la izquierda con las llantas moviéndose en dirección contraria. Y en caso de que sea 'D', el Puerto B es 9 y el Puerto C es 6 para hacer que gire a la derecha con las llantas moviéndose en dirección contraria pero inverso al movimiento en 'I'.

Debido a que el robot se controló mediante Bluetooth, se desarrolló una aplicación utilizando el entorno de desarrollo MIT App Inventor. La interfaz está compuesta por 5 botones que controlan la dirección en la que se moverá el robot, cada botón con la letra que se manda y en forma de flecha. También se tienen 3 botones más para conectar el

dispositivo Bluetooth, para desconectarlo y para salir de la aplicación. El diseño de la interfaz es el siguiente:



Además se programó la aplicación mediante la unión de bloques de código. En ellos se define como mostrar los dispositivos Bluetooth y cómo enlazar el que se escoja. También se programa qué letra mandar al presionar cierto botón, cómo desconectar el dispositivo Bluetooth y cómo cerrar la aplicación. Los bloques se muestran a continuación.



Programa comentado

```

processor 16f877
include<p16f877.inc>
  
```

```

VAR EQU 0X20 ;Valor de control
  
```

;Valores para el retardo

valor1 equ h'21

valor2 equ h'22

valor3 equ h'23

cte1 equ 20h

cte2 equ 50h

cte3 equ 60h

ORG 0

GOTO inicio

ORG 5

inicio

BSF STATUS,RP0

BCF STATUS,RP1

;Se cambia al banco 01

CLRF TRISB

;Puerto B como salida

CLRF TRISC

;Puerto C como salida

BSF TXSTA,BRGH

;Cambia bandera BRGH

MOVLW D'129'

MOVWF SPBRG

;Baud rate = 9600

BCF TXSTA,SYNC

;Comunicación síncrona

BSF TXSTA,TXEN

;Activa transmisión

BCF STATUS,RP0

;Se cambia al banco 00

BSF RCSTA,SPEN

;Habilita puerto serie

BSF RCSTA,CREN

;Recepción sencilla

RECIBE:

BTFSS PIR1,RCIF

;Bandera de recepción completa

GOTO RECIBE

;Esperar recepción

MOVF RCREG,W

MOVWF VAR

;Control = Recepción

MOVLW 0x53

SUBWF VAR,W

BTFSC STATUS,Z

;Si control = 0x53 "S"

GOTO STOP

;Entonces parar

MOVLW 0x41

SUBWF VAR,W

BTFSC STATUS,Z

;Si control = 0x41 "A"

GOTO ADELANTE

;Entonces avanzar

MOVLW 0x54

SUBWF VAR,W

BTFSC STATUS,Z

;Si control = 0x54 "T"

GOTO TRAS

;Entonces retroceder

MOVLW 0x44

SUBWF VAR,W

BTFSC STATUS,Z

;Si control = 0x44 "D"

GOTO DER

;Mover a la derecha

MOVLW 0x49

SUBWF VAR,W

BTFSC STATUS,Z

;Si control = 0x49 "I"

GOTO IZQ

;Mover a la izquierda

GOTO RECIBE

;Recibir otro dato

STOP

;Parar ambos motores

CLRf PORTC	;C = 0
CLRf PORTB	;B = 0
GOTO RECIBE	;Recibir otro dato
ADELANTE	;Horario ambos motores
MOVLW 0X06	
MOVWF PORTC	;C = 0x06
MOVLW 0X0A	
MOVWF PORTB	;B = 0x0A
GOTO RECIBE	;Recibir otro dato
TRAS	;Antihorario ambos motores
MOVLW 0X06	
MOVWF PORTC	;C = 0x06
MOVLW 0X05	
MOVWF PORTB	;B = 0x05
GOTO RECIBE	;Recibir otro dato
DER	;Dirección a la derecha
MOVLW 0X06	
MOVWF PORTC	;C = 0x06
MOVLW 0X09	
MOVWF PORTB	;C = 0x09
GOTO RECIBE	;Recibir otro dato
IZQ	;Dirección a la izquierda
MOVLW 0X06	
MOVWF PORTC	
MOVWF PORTB	;B = C = 0x06
GOTO RECIBE	;Recibir otro dato
;Subrutina para el retardo	
retardo	
MOVLW cte1	
MOVWF valor1	
tres	
MOVLW cte2	
MOVWF valor2	
dos	
MOVLW cte3	
MOVWF valor3	
uno	
DECFSZ valor3	
GOTO uno	
DECFSZ valor2	
GOTO dos	
DECFSZ valor1	
GOTO tres	
RETURN	
END	

Conclusiones y/o comentarios.

Barreiro Valdez Alejandro:

Aunque la comunicación serie asíncrona tiene algunas limitaciones, como posibles errores de sincronización y velocidades de transmisión más lentas en comparación con otros métodos, sigue siendo una opción popular en muchas aplicaciones. Se utiliza en varios dispositivos y periféricos, desde cables y puertos serie hasta módems, impresoras, sensores y dispositivos de adquisición de datos. En este proyecto la utilizamos para generar una aplicación que controle dos motores de un coche para controlar a distancia sus movimientos. En última instancia, la comunicación serie asíncrona ha demostrado ser una solución confiable y accesible para la transmisión de datos. Se logró alambrar el circuito y utilizar el microcontrolador PIC16F877A, que en conjunto con los módulos utilizados, fue convertido en un programa que le permite al usuario mover un coche por medio de una aplicación Bluetooth. Se utilizaron diversos conceptos vistos en las prácticas del laboratorio como el movimiento de un motor, la transmisión y recepción de datos y las diferentes banderas que se utilizan con estos fines.

Gil Márquez Arath Emiliano:

La comunicación serie asíncrona en un PIC 16F877A es una forma popular y eficiente de transmitir datos secuencialmente entre el microcontrolador y otros dispositivos en este caso nuestro teléfono celular y nuestro robot, donde se pudo apreciar que mediante el uso de una aplicación y por medio de comando de control via bluetooth el robot podía realizar movimientos simples de avanzar hacia adelante, atrás, girar a la derecha y ala izquierda. Esto nos puede mostrar que la comunicación serie asíncrona proporciona una solución simple y flexible para aplicaciones que requieren una comunicación confiable a baja velocidad y distancias cortas. Su implementación requiere configurar correctamente los parámetros de comunicación en el módulo USART del microcontrolador y en el dispositivo receptor.

Herrera Carrillo Cristhian:

La comunicación serie asíncrona es un método de transmisión de datos ampliamente utilizado que ofrece flexibilidad y compatibilidad en una variedad de aplicaciones. Aunque tiene algunas limitaciones, como la posibilidad de errores de sincronización y velocidades de transmisión más lentas en comparación con otros

métodos de comunicación serie, su simplicidad y facilidad de implementación la convierten en una opción popular. La comunicación serie asíncrona permite la transmisión de datos bit a bit, utilizando bits de inicio, datos y bits de parada en cada trama de información. Esto facilita la detección de errores y la sincronización entre el transmisor y el receptor, sin requerir un reloj común.

A pesar de sus limitaciones, la comunicación serie asíncrona sigue siendo una opción confiable y eficiente en muchas situaciones. Los avances tecnológicos continúan mejorando la velocidad y confiabilidad de este método de comunicación, lo que garantiza su relevancia y utilidad en el futuro.

Zepeda Baeza Jessica:

Con este proyecto se logró establecer una comunicación asíncrona utilizando un dispositivo Bluetooth que se conecta a las terminales de Rx y Tx y así controlar el funcionamiento de dos motores. Primero, se observó una nueva forma de utilizar estas terminales ya que anteriormente se utilizaba junto con el puerto serie para cargar el programa. En cuanto al programa, lo más complicado fue establecer las configuraciones que habilitan la recepción de datos ya que lo demás sólo fue hacer comparaciones y de acuerdo a ellas, asignar valores a los puertos. También, una parte importante fue conseguir los motores, drivers y componentes de Bluetooth adecuados, para poder construir todo el robot. Pero conociendo tanto las terminales de los motores y las del microcontrolador fue sencillo hacer las conexiones.