



# **Universidad Nacional Autónoma de México**

Facultad de Ingeniería  
División de Ingeniería Eléctrica  
Ingeniería en Computación



## **Proyecto 4 “Puerto serie inalámbrico”**

### **Integrantes:**

- Castelan Ramos Carlos
- Castillo Montes Pamela
- Hernández Jaimes Rogelio Yael

**Materia:** Microcomputadoras

**Grupo:** 01

**Semestre:** 2023-2

**Fecha de entrega:** 29 de mayo 2023



## Reporte Proyecto 4 *"Puerto Serie Inalámbrico"*

### **Circuito Alambrado**

El sistema mínimo está formado por:

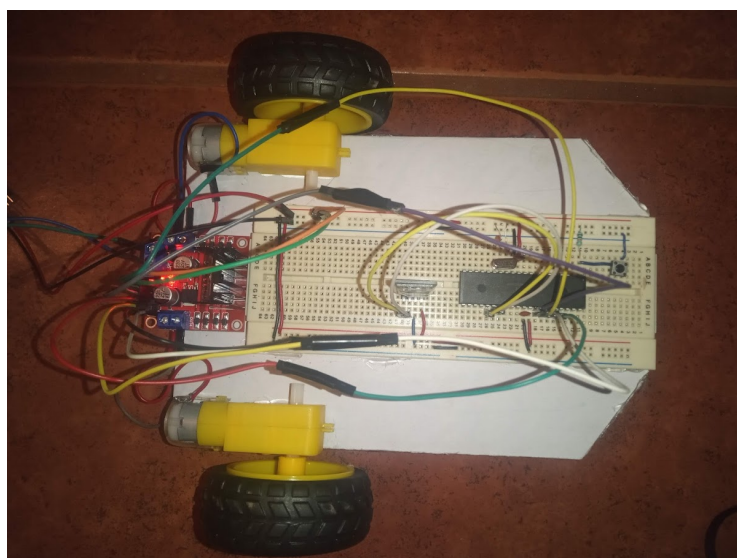
1. Microcontrolador PIC16F877A
2. Cristal de cuarzo 20 MHZ
3. 2 capacitores cerámicos de 22pF
4. 1 push button de 2 pines
5. 1 resistencia 10 K $\Omega$
6. 2 capacitores cerámicos 0.1  $\mu$ F
7. Alambre/Jumpers
8. Protoboard
9. Fuente de poder 5 Volts
10. Transmisor Serial UART/TTL con salida USB

El sistema funcional del proyecto está dado por:

1. 2 Rotomotores de corriente directa.
2. Driver controlador de motores L298N.
3. Módulo Bluetooth HC06
4. Aplicación Móvil de control

### **Alambrado físico**

El alambrado físico del proyecto contempló el uso de dos fuentes de alimentación, para el circuito mínimo fue utilizada la fuente de 5 volts que usualmente usamos, así también se ocupó una fuente para alimentar a los dos rotomotores administrados directamente por el driver controlador de motores L298N, esta fuente de corriente directa permite voltajes de 5 a 12 volts, en nuestro caso se ocupó una fuente alámbrica de 5 volts.



*Imagen 2 Alambrado físico*



Por otra parte el sistema es soportado por una base de papel ilustración reforzada, en donde a sus lados los rotomotores tienen colocadas llantas de plástico con goma, mientras que para el eje superior se colocó una llanta loca de metal.

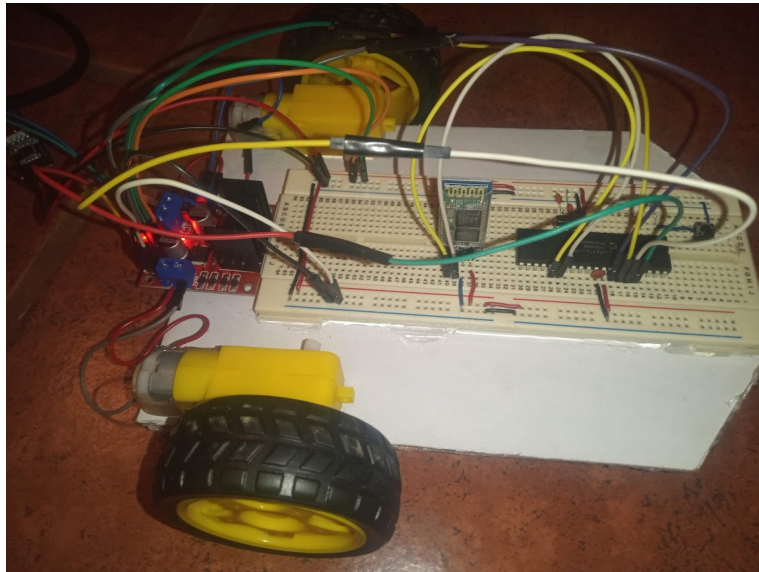


Imagen 2 Alambrado físico

Finalmente la conexión se realizó mediante un módulo bluetooth HC06 con características únicamente de recepción, el cual se vinculó a un teléfono celular mediante el uso de una contraseña genérica "1234", este fue conectado a los pines de transmisión y recepción del PIC, además fue administrado con 5 volts con la misma fuente del PIC.

### Programa ASM

```
1      PROCESSOR 16F877
2      INCLUDE <P16F877.INC>
3      valor1 equ h'21'
4      valor2 equ h'22'
5      valor3 equ h'23'
6      cte1A1 equ .1
7      cte2A1 equ .13
8      cte3A1 equ .255
9
10     ORG 0 ;Vector de reset
11     GOTO INICIO
12     ORG 5
13 INICIO: BSF STATUS,RP0      ;ACCEDEMOS BANCO 1
14         BCF STATUS,RP1
15         BSF TXSTA,BRGH      ;TASA DE TRANSMISION ALTA
16         MOVLM D'129'        ;TASA DE TRANSMISION DE 9600 Bauds
17         MOVWF SPBRG         ;SPBRG<- D'129'
18         BCF TXSTA,SYNCR     ;Habilitamos modo asincrono
19         BSF TXSTA,TXEN      ;Habilitamos
20         CLRF TRISD          ;Configuramos PORTD como salida
21         CLRF TRISB          ;Configuramos PORTB como salida
22         BCF STATUS,RP0      ;Regresamos al banco 0
23         CLRF PORTD          ;Limpiamos el PORTD
24         CLRF PORTB          ;Limpiamos el PORTB
25         BSF RCSTA,SPEN      ;Habilitamos el puerto serie
26         BSF RCSTA,CREN      ;Habilitamos la recepción continua en la comunicacion asincrona
```

Imagen 3 Configuración inicial de programa

El programa inicia declarando registros auxiliares para operar junto con el vector *reset* y la dirección de inicio del programa. Los registros auxiliares declarados tienen el siguiente uso:

- *valor*, *valor1* y *valor2*: Los cuales son registros para operar en la subrutina de retardo y están relacionados a la conversión a segundos.

Prosiguiendo con la estructura del programa, encontramos la configuración de la PIC. Se configura el puerto serie con una tasa de transmisión alta a 9600 Bauds. Así mismo, habilitamos el modo asíncrono y la transmisión. De igual forma, se realiza la



configuración de los registros TRISB y TRISD en 0 para posteriormente configurar PORTB y PORTD como salidas. De regreso en el banco 1, se realiza una limpieza de los puertos paralelos con el objetivo de evitar errores de salida. Así mismo, en el banco 0, habilitamos el puerto serie y la recepción continua asíncrona.

```

RECIBE:    BTFS PIR1,RCIF      ;¿PIR1.RC1F==1? ESPERAMOS A QUE SE RECIBA UN DATO
           GOTO RECIBE        ;NO, SIGUE A LA ESPERA DE LA RECEPCION
           MOVF RCREG,W        ;SI, OBTEN EL DATO LEÍDO Y ALMACENALO EN W
           ;MOVWF TXREG        ;MOVEMOS EL DATO LEÍDO AL REGISTRO DE TRANSMISION
           MOVLW "A" ;CÓDIGO ASCII DEL A
           XORWF RCREG,W
           BTFS STATUS,Z      ;¿El valor leído es igual que W?
           GOTO ADELANTE      ;Si, es igual
           ;NO
           MOVLW "D" ;CÓDIGO ASCII DEL D
           XORWF RCREG,W
           BTFS STATUS,Z      ;¿El valor leído es igual que W?
           GOTO DERECHA       ;SI

           MOVLW "S" ;CÓDIGO ASCII DEL S
           XORWF RCREG,W
           BTFS STATUS,Z      ;¿El valor leído es igual que W?
           GOTO PARAR         ;SI

           MOVLW "I" ;CÓDIGO ASCII DEL I
           XORWF RCREG,W
           BTFS STATUS,Z      ;¿El valor leído es igual que W?
           GOTO IZQUIERDA     ;SI

           MOVLW "T" ;CÓDIGO ASCII DEL T
           XORWF RCREG,W
           BTFS STATUS,Z      ;¿El valor leído es igual que W?
           GOTO ATRAS         ;SI

```

Imagen 4: Proceso de recepción de datos y comparación

Esta parte del código evalúa el valor recibido por el puerto serie. El valor que enviemos por el módulo bluetooth ingresa directamente en el puerto serie del controlador, donde se evalúa respecto a los valores de movimiento posibles. Comparamos su ASCII pues este es el valor que recibe. Los cinco casos planteados corresponden a la asignación:

Avanza	A
Retrocede	T
Izquierda	I
Derecha	D
Para	S

Estos valores son los que se envían según la opción pulsada en la aplicación. El programa evalúa qué valor se recibió con la operación XOR. Si la bandera Z se activa indica que los elementos fueron iguales y no se salta la instrucción GOTO al caso correspondiente.

### **Caso 1: Adelante**

Para este caso simplemente hace que ambos motores giren a la derecha. Esta configuración es '10' en cada motor. Dado a que estos se asignan de forma contigua se establecen de esta forma. Tras asignar el valor al puerto B generamos un pequeño retardo para poder observar el movimiento de los motores antes de hacer un cambio de dirección.

```

56      ADELANTE ;HACIA ADELANTE
57      MOVLW B'1010'; M1=DERECHA M2=DERECHA
58      MOVWF PORTB
59      CALL RETARDO
60      GOTO RECIBE

```

Imagen 5: Caso de avance



### Caso 2: Derecha

```
61 DERECHA ;HACIA LA DERECHA
62      MOVLW B'1001' ;M1=DERECHA M2=IZQUIERDA
63      MOVWF PORTB
64      CALL RETARDO
65      GOTO RECIBE
```

### Caso 3: Parar

En este caso los motores no requieren una dirección, simplemente basta con limpiar el puerto B.

```
66 PARAR ;PARA
67      CLRF PORTB
68      GOTO RECIBE
```

### Caso 4: Izquierda

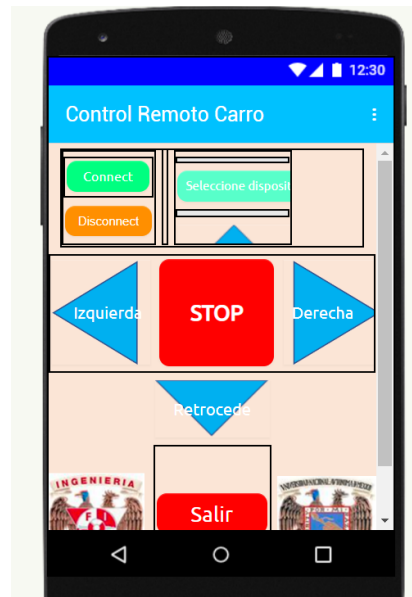
```
69 IZQUIERDA ;HACIA LA IZQUIERDA
70      MOVLW B'0110' ;M1=IZQUIERDA M2=DERCHA
71      MOVWF PORTB
72      CALL RETARDO
73      GOTO RECIBE
```

### Caso 5: Atrás

```
74 ATRAS:      MOVLW B'0101' ;M1=IZQUIERDA M2=IZQUIERDA
75      MOVWF PORTB
76      CALL RETARDO
77      GOTO RECIBE
```

## Desarrollo de aplicación

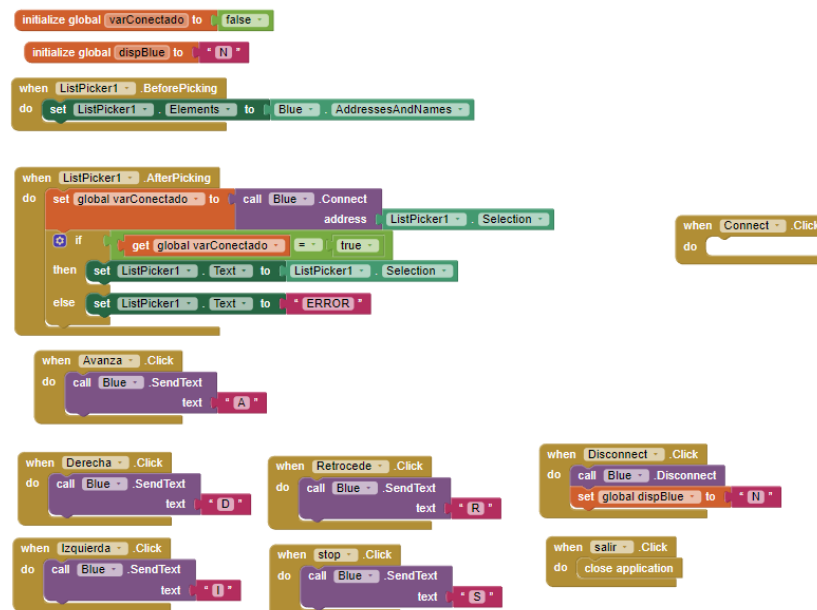
El mapeo de elementos en nuestra aplicación fue desarrollado de forma intuitiva, por lo que los elementos se asignaron como diseño propio. En la plataforma dicha asignación se observa de la siguiente manera:



Esta aplicación contiene un botón para cada caso de los descritos con anterioridad. Así mismo, se tiene una cliente bluetooth para poder conectarse con el módulo de Bluetooth del controlador. Para este se definen tres botones especiales. Uno permitirá listar los dispositivos Bluetooth disponibles (y sincronizados), disconnect permite concluir la conexión con los dispositivos. El connect permitiría conectar el dispositivo seleccionado, sin embargo,



esta conexión ya se hace desde la selección de dispositivo. La funcionalidad de los botones quedan ocultos para el usuario, los cuales son:



## Conclusiones:

### **Castelan Ramos Carlos**

Para el desarrollo de este proyecto logramos implementar el uso de señales inalámbricas haciendo uso de un módulo bluetooth haciendo uso de los puertos serie con los pines de transmisión y recepción del PIC logramos interpretar las instrucciones de nuestra aplicación móvil para hacer mover nuestro pequeño carro. Así también adquirimos conocimientos para el armado de diferentes dispositivos físicos como los rotomotores junto a su driver para el entendimiento de señales físicas locales implementamos diferentes métodos de prueba para localizar errores en armado y así contemplar una mejor solución de funcionamiento físico.

### **Castillo Montes Pamela**

Para este proyecto, se utilizaron las bases sentadas en los anteriores respectó al uso y lectura de puertos para comunicar el sistema mínimo con una gran variedad de dispositivos y periféricos. En específico, se hizo uso de los conocimientos adquiridos dentro de las prácticas 5 a 7 de la misma materia.

Por otro lado, se adquirió el conocimiento necesario para realizar una aplicación móvil funcional haciendo uso de herramientas de alto nivel, las cuales realizan sus funciones mediante bloques, mismo que permitió una fácil y rápida forma de desarrollar esta app.

De igual forma se realizó la integración con un módulo de tipo bluetooth dentro de nuestro sistema para la recepción de datos, concluyendo así satisfactoriamente la creación y desarrollo del proyecto.

### **Hernández Jaimes Rogelio Yael**

En el diseño de sistemas se tiene que los dispositivos a controlar no pueden tener una conexión cableada todo el tiempo. Ya sea por la dificultad de acceso o que su naturaleza



dificultaría esto. De ahí que se empleen conexiones inalámbricas como el Bluetooth. Ahora bien, la información que este provee no es necesariamente bits de control como con los que opera el microcontrolador, sino que contiene mayor información y que procede de equipos diferentes al controlador. La comunicación entre computadora y el controlador no es igual realmente, sin embargo, podemos adecuar la información para que se maneje en el microcontrolador. En el caso del módulo bluetooth seguimos el mismo principio. Verdaderamente el módulo por sí mismo no produce la información, sino que transmite lo que otro dispositivo indicó. El puerto serie del controlador permite comunicar el PIC con dispositivos más complejos como los celulares y computadoras, pudiendo interactuar de forma más gráfica con esta. Ejemplo de esto es la creación de aplicaciones que un usuario puede manejar desde el celular y con el puede controlar el PIC.