

Ingeniería en Electrónica y Telecomunicaciones

Integrantes del equipo:

Silva Palafox Adrián

Trejo De Arcos Felipe Adriel

Velázquez Reyes Jimmy Daniel

Materia: Diseño con Microprocesadores








Grupo: 501

**“Proyecto del Primer Parcial – Robot Seguidor de Línea”**

Maestro: Luis Gerardo Vilchis Soto

15 de septiembre del 2023, León Guanajuato.

## I. Ficha técnica:

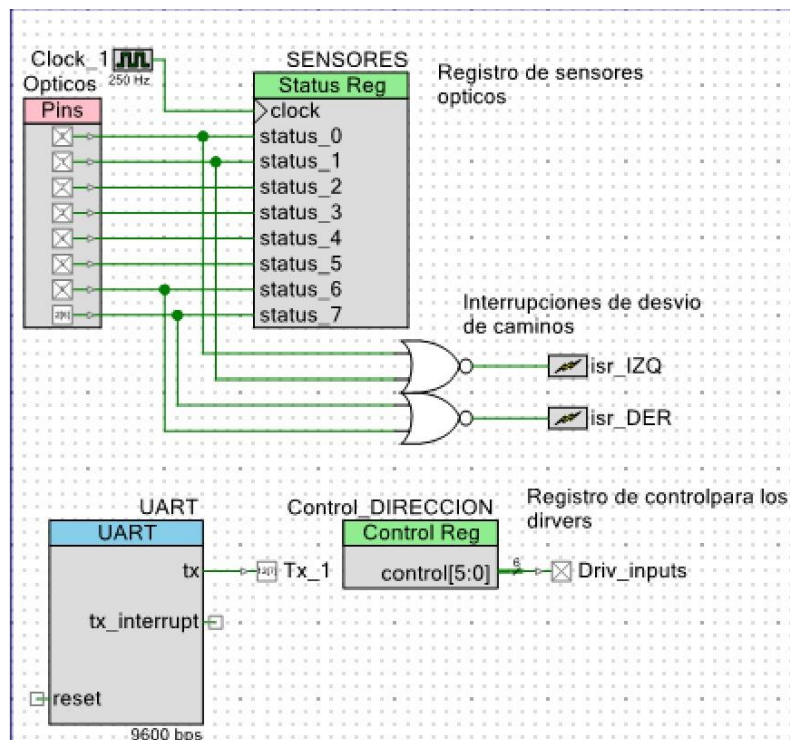
Componentes electrónicos utilizados			
Capacitor electrolítico 1μF a 50V		IC 74HC14	
Capacitor cerámico 104		IC L293B	
Diodo Schottky 1N4148			
Interruptor		Pila 2200 mAh	

## Microcontrolador utilizado
















CY8CKIT-059



## II. Top design:



III. **Pines:** estos fueron los pines utilizados para el funcionamiento.

	Name	Port
	Driv_inputs[0]	P12[3] ▾
	Driv_inputs[1]	P12[4] ▾
	Driv_inputs[2]	P12[5] ▾
	Driv_inputs[3]	P0[1] ▾
	Driv_inputs[4]	P0[2] ▾
	Driv_inputs[5]	P0[3] ▾
	Opticos[0]	P0[4] ▾
	Opticos[1]	P0[5] ▾
	Opticos[2]	P0[6] ▾
	Opticos[3]	P0[7] ▾
	Opticos[4]	P2[3] ▾
	Opticos[5]	P2[4] ▾
	Opticos[6]	P2[5] ▾
	Opticos[7]	P2[6] ▾
	Tx_1	P12[7] ▾

IV. **Código:**

Las siguientes imágenes son el main.c del proyecto.

```

1  #include "string.h"
2  #include "project.h"
3  //ECOGEMOS LAS TABLAS DE VERDAD SEGUN EL DRIVER DE LOS MOTORES
4  // CON MODULO L293N.h o L298.h
5  #include "L293.h"
6
7  //Función para binarizar un numero en un array de caracteres
8  void getBinary(uint8_t n, char *reg);
9
10 //INTERRUPCIONES DE DESVIO
11 uint8_t flag_derecha=0,flag_izquierda=0;
12 CY_ISR(isr_DER){//Los sensores de la derecha están sobre línea
13     isr_DER_ClearPending();
14     flag_derecha = ~flag_derecha;
15 }
16
17 CY_ISR(isr_IZQ){//Los sennsosres de la izquierda están sobre línea
18     isr_IZQ_ClearPending();
19     flag_izquierda = ~flag_izquierda;
20 }
21
22
23 int main(void)
24 {
25     CyGlobalIntEnable;
26     isr_DER_StartEx(isr_IZQ);
27     isr_IZQ_StartEx(isr_DER);
28     isr_IZQ_ClearPending();
29     isr_IZQ_ClearPending();
30
31     //COMENTABLE_ SOLO ES PARA DEBUGUEAR
32     /*
33     UART_Start();
34     UART_ClearTxBuffer();
35     char LECTURA[8];
36     */
37
38     uint8_t sensores;//Para la lectura de los sensores
39
40     Control_DIRECCION_Write(BREAK);
41     CyDelay(3000);

```

```

43     for(;;)
44     {
45         //sensamos los motores
46         sensores = SENSORES_Read();
47
48         //Comentable NO necesario
49         //getBinary(sensores,LECTURA);
50         //UART_PutString(LECTURA);
51         //UART_PutChar('\n');
52
53
54         //Se ejecutan los cambios de dirección cada vez que la interrupción llegue
55         if(flag_derecha == 0xFF){
56             flag_derecha = ~flag_derecha;
57             Control_DIRECCION_Write(LEFT_HACK);
58         }
59         if(flag_izquierda == 0xFF){
60             flag_izquierda = ~flag_izquierda;
61             Control_DIRECCION_Write(RIGHT_HACK);
62
63         }
64
65         //Va pa adelante simepre que se lea en los sensores 0x11100111
66         if(sensores == 0xE7){
67             Control_DIRECCION_Write(GO);
68             CyDelay(100);
69         }
70
71     }
72 }
73
74 void getBinary(uint8_t n, char *reg){
75     int loop;
76     /*
77     3th bit to 0th bit
78     Tremendo hack
79     En ifs 0-->False !=0-->True
80     bit,bit,bit,bti,espacio
81     */
82     for(loop=7; loop>=0; loop--)

```

```

83         {
84             if( (0x01 << loop) & n)//Concuenda en 1&1???
85                 reg[loop] = '1';
86             else
87                 reg[loop] = '0';
88         }
89     }

```

Las siguientes capturas corresponden al L293.h :

```

1  //*****
2  //Control de giro de motores
3  /*
4      EN|1A|2A
5      1 |0 |1      #define Derecha 0x05
6      1 |1 |0      #define Izquierda 0x06
7      1 |0 |0      #define Paro_seco 0x04
8      1 |1 |1      #Paro_SECO2  0x07
9      0 |X |X      #define Paro_Z  0x00
10
11      Para el segndo motor hay que sumar para obtener la máscara correcta
12
13      Registro de control que es de 8bit los ultimos dos más significantes no importan
14      x| |x |bit5 |bit4 |bit3 |bit2 |bit1 |bit0
15      . .  ENb  |B1  |B2  |ENa  |A1  |A2      Estos son los pines de Control del ic L293b
16  */
17
18  //Combinaciones de la tabla de verdad por un MOTOR
19  #define freno 0x04
20  #define atras 0x06
21  #define frente 0x05
22  #define OFF  0x00
23
24  //Combinaciones de la direcciones que puede tomar el seguidor 2 MOTORES

```

```

25      /*
26
27          MOTOR_b      MOTOR_a
28          0000000| | | | | | | | 0000000
29          oo00000|      |0000000
30          oo00000|      |0000000
31                  |      |
32                  |      |
33                  | | | | | | | |
34      En un registro de control
35      8bits --> x | x | ENB | IN_3 | IN_4| ENA | IN_1 | IN_2
36  */
37  #define GO 0x2D //MB freno MA frente
38  #define BACK 0x36 //MB atras MA atras
39  #define BREAK 0x24 //MB paro MA paro
40  #define RIGHT 0x2C //MB frente MA par0
41  #define LEFT 0x25 //MB paro MA frente
42
43  //MÁSCARAS PARA HACKS DE GIRO
44  #define RIGHT_HACK 0X2E //MB frente MA atrás
45  #define LEFT_HACK 0X35 //MB freno MA atrás

```

```

47  //*****
48  //MÁSCARAS PARA UN CONTROL DE PUENTE H CON DOS RELEVADORES
49  /*Registro de control que es de 8bit los ultimos dos más significantes no importan
50
51          180dc
52          |
53          |
54          |
55      NC_rele1 ----- NC_rele2
56
57      COMUN_rel1*-----MOTOR---*COMUN_rele2
58
59      NO_rele1 ----- NO_rele2
60          |
61          |
62          |GND
63
64      Para un unico motor y un unico puente H con relevadores
65      |x |x |x |x |x |bit2 |bit1 |bit0
66      . .      |ENa |A1 |A2      Estos son los pines de Control del ic L293b
67
68  */

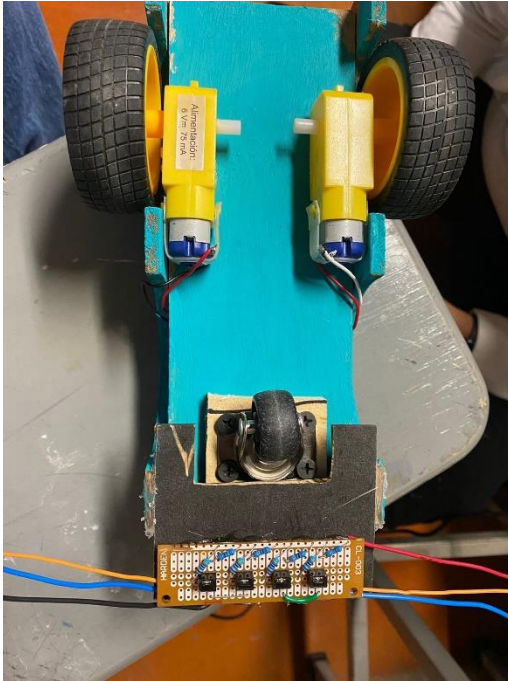
```



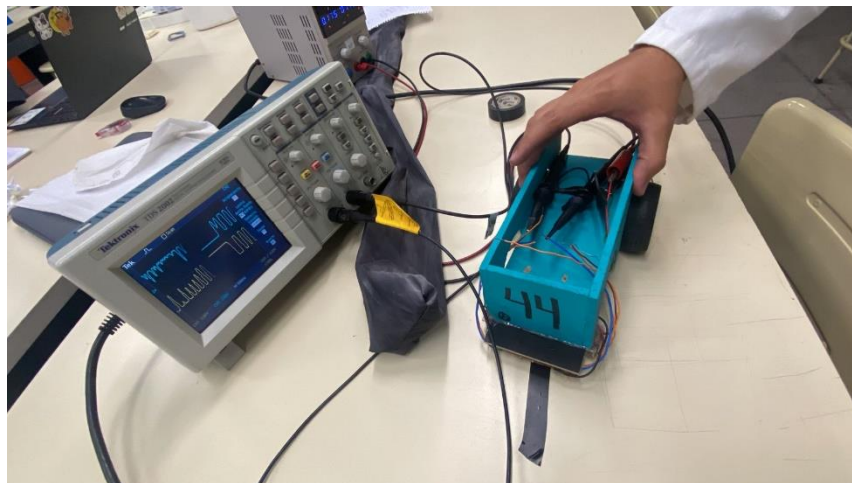
```
69
70     #define GIRO_1 0x02
71     #define GIRO_2 0x01
72     #define PARO    0xFF
73     #define DISABLE (~0x04)
```

## V. Producto final:

Elaboración:



Calibración:



Producto final:

