Laboratorio de Diseño de Robots Móviles
Practica No. 4
Operación de un robot móvil usando el microcontrolador PIC16F877

Objetivo: Controlar la operación del robot móvil construido en las practicas anteriores usando máquinas de estados.
Desarrollo: Para cada uno de los siguientes apartados, realizar los diseños electrónicos y programas que se piden.
Duración: Tres semanas.

1. Configurando el puerto A del PIC como entrada, conecte dos sensores de contacto a dos pines de esté. Muestre el valor leído de los sensores en una hyperterminal.

2. Escriba una función en C que controle los movimientos de un robot móvil, en está se indicará el ángulo, en radianes, que el robot primero girará y la distancia que después avanzara. La función deberá tener el siguiente formato:

```
move_robot(float distancia, float angulo)
{

}
```

3. Escriba una función en C que regrese el valor del sensor indicado. La función deberá tener el siguiente formato:

```
float show_sensor(char *sensor, int num_sensor)
{

 float x;

 return(x);
}
```

3. Cargue el programa en C que se encuentra en el apendice, que ejecuta el algoritmo de un robot móvil que evade obstáculos, como el que se muestra en la figura 1, en su robot. En la figura 2 se muestra el algoritmo de un robot móvil que evade obstáculos, cuando los sensores de tacto Si y Sd sensan un obstáculo sus valores son igual a cero, en caso contrario es uno.

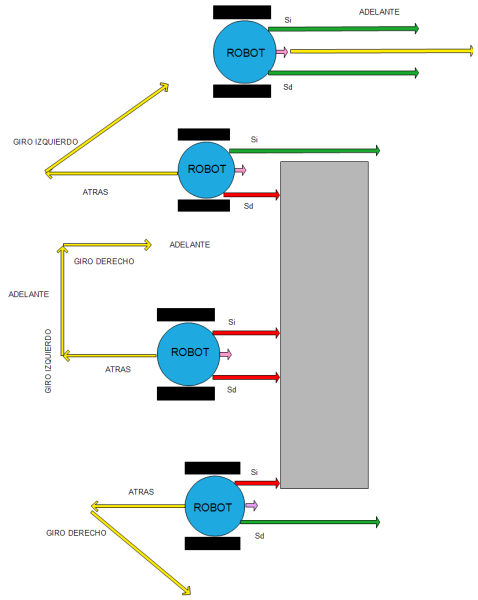

Figura 1. Robot Móvil que evade obstáculos

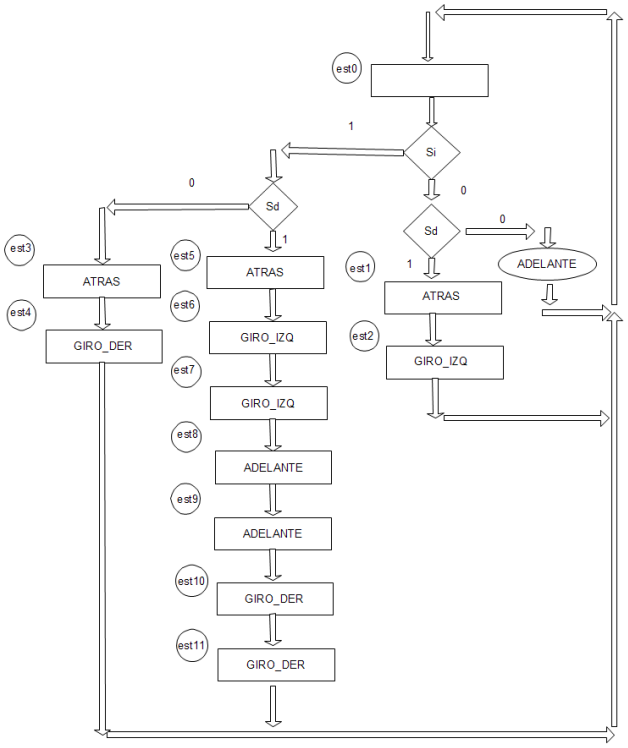

Figura 3. Algoritmo de un robot móvil que evade obstáculos

Apéndice

Programa en C para un robot móvil que evade obstáculos.

```
#include <16f877.h>
#device ADC=8
#include <stdlib.h>
#fuses HS,NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7)
#org 0x1FFF, 0x1FFF void loader16F877(void){}

// Definicion de constantes
#define ADELANTE move_robot(AVANCE, 0.0f);
#define ATRAS move_robot(-AVANCE, 0.0f)
#define GIRO_IZQ move_robot(0.0f, 0.7854f)
#define GIRO_DER move_robot(0.0f, -0.7854f)
#define SENSOR_IZQ 1
#define SENSOR_DER 2
#define SECURE 5.0f

move_robot(float distancia, float angulo)
{
}

float show_sensor(char *sensor, int num_sensor)
{
 float x;
```

```
 return(x);
}

// Esta funcion lee el sensor indicado
int lee_sensor(int num_sensor){
 float sensor;
 int valor_sensor;
 char buffer[20];
 char nombre_sensor[20];

 strcpy(nombre_sensor,"contacto");
 sensor = show_sensor(nombre_sensor, num_sensor);

 if (sensor > SECURE)valor_sensor=0;
 else valor_sensor=1;

 return(valor_sensor);
}

// Funcion que evade obstaculos
void evade()
{
	float AVANCE=10.;
	int estado;
	int Si, Sd;
	char buffer[20];

	// Estado inicial
	estado = 0;

	// Loop infinito
	while(1)
	{
		printf("Estado presente: %d\n\r",estado);

		// Acciones
		switch ( estado )
		{
			case 0:	// est0
				// Lee sensores
				Sd = lee_sensor(SENSOR_DER);
				Si = lee_sensor(SENSOR_IZQ);

				printf("\n\rLectura sensores Si %d Sd %d\n\r",Si,Sd);
				if (Si == 0)
					if (Sd == 0) ADELANTE;
					break;
			case 1:	// est1
				ATRAS;
				break;
			case 2: // est2
				GIRO_IZQ;
				break;
			case 3:	// est3
				ATRAS;
				break;
			case 4: // est4
				GIRO_DER;
				break;
			case 5: // est5
				ATRAS;
```

```
                        break;
                case 6: // est6
                        GIRO_IZQ;
                        break;
                case 7: // est7
                        GIRO_IZQ;
                        break;
                case 8:  // est8
                        ADELANTE;
                        break;
                case 9: // est9
                        ADELANTE;
                        break;
                case 10: // est10
                        GIRO_DER;
                        break;
                case 11: // est11
                        GIRO_DER;
                        break;
        }

        // Transiciones
        switch ( estado )
        {
                case 0:  if (Si == 0)
                                if (Sd == 0) estado = 0;
                                else estado = 1;
                        else
                                if (Sd == 0) estado = 3;
                                else estado = 5;
                        break;
                case 1:  estado= 2;
                        break;
                case 2:  estado = 0;
                        break;
                case 3:  estado = 4;
                        break;
                case 4:  estado = 0;
                        break;
                case 5:  estado = 6;
                        break;
                case 6:  estado = 7;
                        break;
                case 7:  estado = 8;
                        break;
                case 8:  estado = 9;
                        break;
                case 9:  estado = 10;
                        break;
                case 10:estado = 11;
                        break;
                case 11:estado = 0;
                        break;
        }
    }
}

void main(){
    // Ejecuta el algoritmo de evasion de obstaculos
    evade();
}
```