CAPÍTULO VIII

Desarrollo Electrónico para la Locomoción del Robot

8. DESARROLLO ELECTRÓNICO PARA LA LOCOMOCIÓN DEL ROBOT.

En este capítulo se explica la implementación del sistema electrónico necesario para hacer posible el funcionamiento del robot de acuerdo a los objetivos propuestos.

8.1 Arquitectura del Hardware

En la actualidad la elaboración de robots se ha convertido en una tarea accesible debido a la aparición de los microcontroladores. Un microcontrolador, como ya se describió con anterioridad, es un circuito integrado o chip que integra en las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un circuito integrado. Como ya vimos, una rama familia de este tipo de dispositivos son los PIC que conforman una parte denominada microcontroladores tipo RISC fabricados por Microchip Technology Inc.

Para la selección del microprocesador a utilizar es necesario tomar en cuenta cuales y cuántos son los elementos necesarios para conformar al robot, dichas necesidades son:

- Una entrada a convertidor A/D
- Una salida digital para led emisor
- 18 salidas digitales para control de servomotores por medio de PWM
- Facilidad de programación
- Facilidad de adquisición

Existen muchos microcontroladores en el mercado que cumplen con dichas características, sin embargo debido a nuestra experiencia, la facilidad de adquisición, la sencillez de manejo y sobre todo las características especiales con las que cuenta el PIC 18F4550 (cuyas características han sido mencionadas con anterioridad), se decidió utilizarlo como circuito de control.

Para lograr un diseño electrónico adecuado se desarrollaron y tomaron en consideración principalmente 2 diagramas uno externo general o de bloques y uno eléctrico de la composición de todos los elementos del circuito. En el diagrama de bloques externo se representan los diferentes elementos físicos que forman parte del control del robot caminante, esto se observa en la figura 8.1

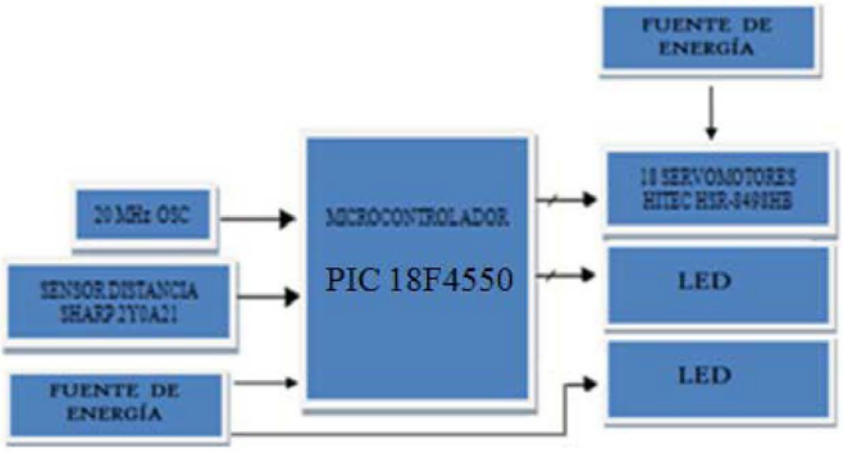

Figura 8.1 Diagrama de Bloques Externo

El diagrama eléctrico de la representación a bloques anterior se muestra a continuación en la figura 8.2:

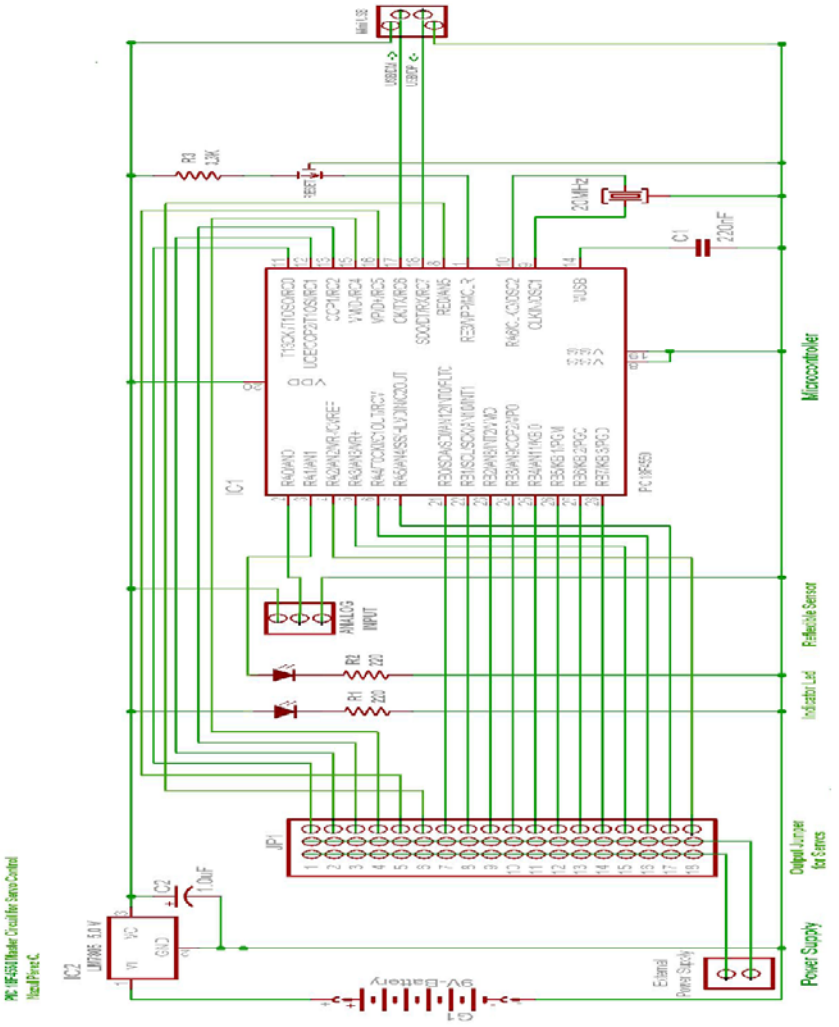

Figura 8.2 Diagrama eléctrico

8.2 Fuentes de Energía

El robot caminante consta de 2 paquetes de baterías de niquel-cadmio recargables: un paquete para energizar el microprocesador y la electrónica que posee una capacidad de 6 volts a 1000 mAh y el segundo paquete para energizar los 18 servomotores que está conformado por 2 grupos de baterías de de 3.7 volts a 1300mAh en serie proporcionando 7.4 volts a 1300mAh.

Cuenta a su vez con 3 interruptores: 2 microswitch deslizables y un micropush. El primer microswitch se utiliza para energizar los motores, el segundo para energizar la tarjeta de control y con el micro push se genera un pulso de reset para el circuito de control. La figura 8.4 muestra las pilas utilizadas en el robot caminador.

Figura 8.4 Pilas utilizadas en el robot

8.3 Entradas Analógicas

En el diagrama podemos ver que la única entrada analógica es el sensor de proximidad cuyo objetivo es la detección de un objeto posicionado al frente del robot, para esto el microcontrolador tiene por entrada en el modulo de conversión analógico-digital la señal de un sensor infrarrojo de presencia con alcance de 4 a 80 cm.

El Sharp GP2Y0A21 es un sensor de infrarrojos que proporciona una lectura continua de la distancia medida como una tensión analógica dentro de un rango de 4" a 32". La tensión de alimentación es de 5V y la tensión de salida varia unos 2 voltios de diferencia entre el margen mínimo y el máximo de la distancia medida. El encapsulado es similar a otros sensores Sharp, pero presenta una mayor distancia entre la lente y el sensor con el fin de aumentar el rango de trabajo. La conexión se realiza mediante un conector JST de 3 vías, 2 de esas vías se utilizan para la alimentación y la otra para la salida de datos. La salida está disponible de forma continua y su valor es actualizado cada 39 ms. La salida se conecta a la entrada del convertidor analógico digital del microprocesador el cual convierte la distancia en un valor. El valor proporcionado a una distancia de 26 cm entre el sensor y el objeto es utilizado como punto crítico de decisión en la realización de posibles respuestas del robot, en nuestro caso es el momento de giro del robot. La figura 8.5 muestra la fotografía de este sensor.

Figura 8.5. Sensor utilizado en el robot caminante

8.4 Entradas Digitales

Como se puede ver en el diagrama de bloques general, la única entrada de datos digitales es la que se requiere para la programación propia del microprocesador. Para dicha comunicación aprovechamos la ventaja de este modelo de microcontrolador el cual posee un modulo integrado para la comunicación por medio del protocolo USB El microcontrolador PIC 18F4550 contiene una interfaz serie compatible con el SIE

(Serial Interface Engine, máquina con comunicación serie) USB(Universal Serial Bus), ya sean de "Full Speed" (2.0) y "Low Speed" (1.1) que permite la comunicación rápida entre cualquier dispositivo USB y el microcontrolador PIC. El SIE se puede interconectar directamente al USB, utilizando el Transmisor / Receptor interno, o puede conectarse a través de un Transmisor / Receptor externo. El PIC tiene un regulador interno de 3,3 Volt para accionar el Transmisor / Receptor interno en aplicaciones de 5 Volt (ver figura 8.6)

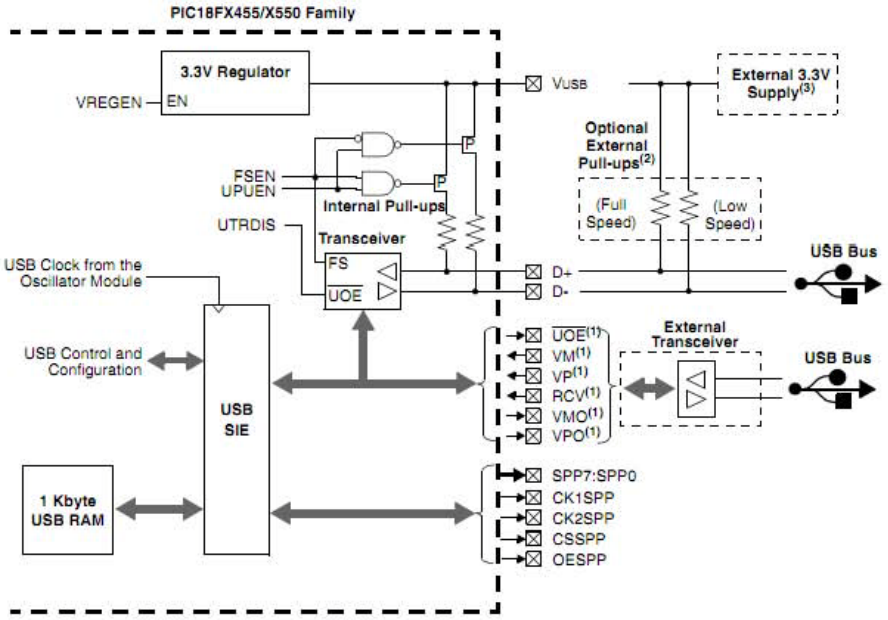

Figura 8.6 Estructura del PIC

Las operaciones del modulo USB se configuran y controlan mediante registros. En total hay 22 registros para manejar las transacciones del USB. Los principales registros son:

- Registro de control del USB (UCON): contiene los bits necesarios para dirigir el comportamiento del modulo durante la transferencia. El registro contiene los bits que gobiernan los permisos del periférico principal del USB, y desactivación de la transferencia de paquetes.

- Registro de configuración del USB (UCFG): contiene la mayor parte de los bits que dirigen el comportamiento del modulo USB como la velocidad del bus, permiso de las resistencias Pull-up del PIC, permiso del transmisor del PIC y el uso del Buffer Ping-pong.

- Registro de estado de la transferencia del USB (USTAT): cuando el SIE publica una interrupción de transferencia completa por el USB, hay que leer USTAT para determinar el estado de la transferencia.

- Registro de dirección de dispositivo USB (UADDR): contiene una única dirección del USB que el periférico descifra cuando esta activo. Se pone a "0" cuando recibe un Reset del USB, la dirección del USB la tiene que escribir el microcontrolador durante la parte de Setup como parte del Firmware de ayuda del USB.

Es importante mencionar que para el manejo del protocolo de comunicaciones USB y la administración de los datos enviados y transmitidos se utiliza una herramienta muy socorrida denominada BootLoader.
Un BootLoader es un conjunto de instrucciones que forman un programa que se mantiene fijo y sin modificación en la memoria de un microprocesador, en este caso un microcontrolador, para permitir un manejo y/o actualización de los programas internos (*firmware*) sin necesidad de utilizar programadores (*hardware*) específicos.

Normalmente para la programación este tipo de circuitos es necesario el uso de programadores o quemadores de microcontroladores cada vez que se requiere modificar algún programa. Con el uso de un BootLoader solo es necesaria una única vez para cargar el mencionado BootLoader al microcontrolador y luego basta con una sencilla aplicación en el ordenador para cambiar el funcionamiento de los programas desarrollados, utilizando para esto la conexión al puerto USB (figura 8.7)
Una manera sencilla de explicar el uso de los BootLoaders es comparándolos como el sistema operativo de una computadora, su función es la administración de tareas y programas que ejecuta el sistema.

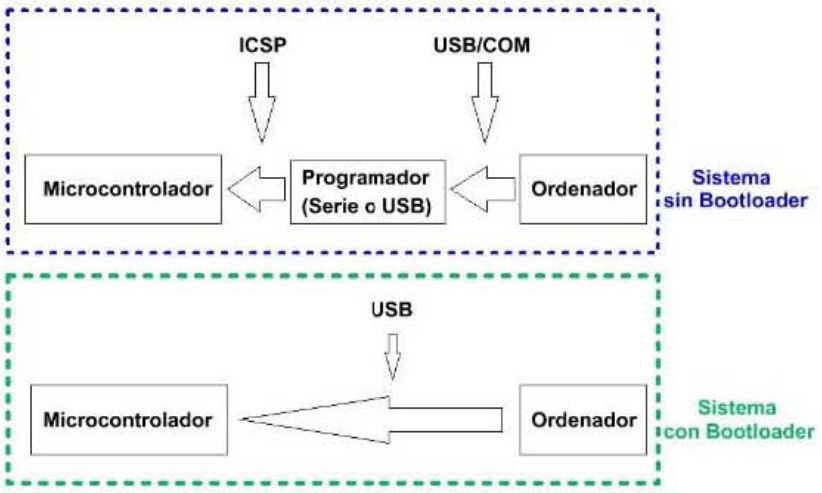

Figura 8.7 Diferencias Bootloader

Con esta poderosa herramienta tanto la comunicación como la modificación del programa general de locomoción resulta muy simple limitándonos el trabajo al perfeccionamiento de las rutinas del robot.

8.5 Salidas Digitales

Como podemos observar en el diagrama interno 8.3, en el robot se tienen 3 salidas digitales:

- La primera para un led emisor , cuya función es indicar el energizado del circuito,
- La segunda para el segundo led emisor cuya función es la representación visual al momento de ser detectado un objeto por el sensor de proximidad,
- El tercer tipo de salida digital es la que proporciona las señales PWM a cada servomotor.

8.6 Envío de Señales PWM

Para el envío de las señales, 18 de los puertos entrada salida se configuran como salidas digitales y por medio de ellos se envían las 18 señales en PWM, una señal para cada servomotor.

Es importante mencionar que la "condición ideal" consiste en que el circuito mantenga un envió constante de las 18 señales de PWM a cada servomotor con el fin de mantener la información de la posición en la que se encuentra cada uno, desafortunadamente esto no es posible ya que nuestro circuito trabaja de manera secuencial y basa su funcionamiento en la ejecución continua de instrucciones.

Sin embargo cada PWM es totalmente independiente esto quiere decir que el control de cada motor en cada pata es individual por lo que es posible entonces repetir el envío de la señal de control PWM de cada motor en un tiempo relativamente pequeño para tratar de simular la "condición ideal".

En la figura 8.8 se puede ver gráficamente lo antes mencionado:

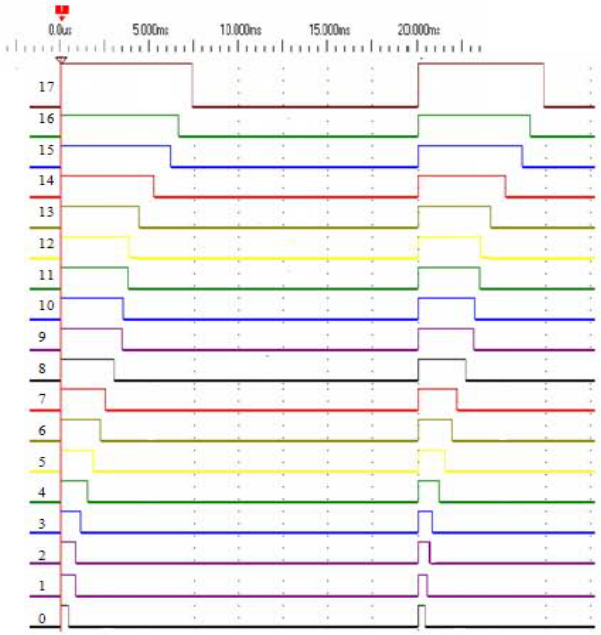

Figura 8.8 PWM para cada motor

Para generar la señal de PWM en cada puerto existen muchas opciones entre ellas destacan: Salidas en alto y bajo, Ciclos de Trabajo e Interrupciones.

En nuestro caso la opción en la que mejor resultado obtuvimos fue la conmutación controlada de salidas en alto y bajo. La anchura del pulso viene definida por el retardo creado por software y que mantendrá a nivel lógico 1 la salida durante la duración de este, al finalizar dicho retardo la salida pasa a nivel lógico 0 y entrará en funcionamiento otra rutina de retardo con el tiempo necesario para completar los 20ms y obtener la frecuencia de 50Hz necesaria. Este proceso se repetirá indefinidamente por cada una de las 18 líneas de salida. Este set de instrucciones es

vuelto a enviar con una frecuencia de 10kHz para lograr semejar la "condición ideal" y mantener al robot ya sea de pie o en movimiento según sea el caso.

8.7 Rutina general

El control de los movimientos que se tendrá sobre el robot caminante se lleva a cabo mediante subrutinas: Posicionamiento Inicial (posición 0), Avance Frontal y Rotación 180°.

En la figura 8.9 se muestra el trayecto y movimientos del robot caminante; se inicia desde una posición inicial (inicio/meta) y conforme el sensor indique que no se tiene un obstáculo el robot caminará hasta encontrar uno. En ese momento seguirá una secuencia de movimientos que le permita dar un giro de 180° posteriormente regresara a la rutina de avance hacia la meta de regreso.

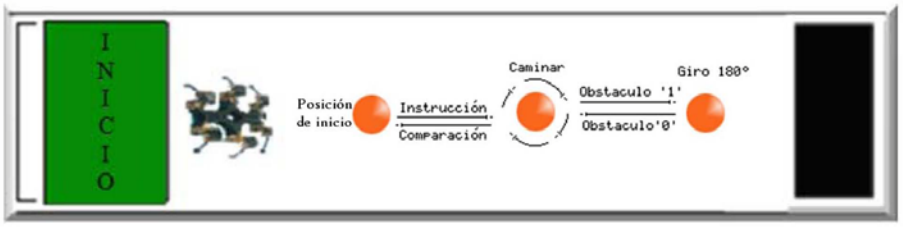

Figura 8.9 Trayectoria a seguir para el robot caminante

De acuerdo a las reglas de la competencia solo se lleva cabo una vez el trayecto a la pared por lo que no es un ciclo y por lo tanto el robot no debe adquirir la posición inicial de salida (viendo hacia la pared), esto implicaría otro giro de 180° sino que únicamente bastará con cruzar la meta.

A continuación se ejemplifica el diagrama de flujo de la figura 8.10 del programa utilizado para la competencia:

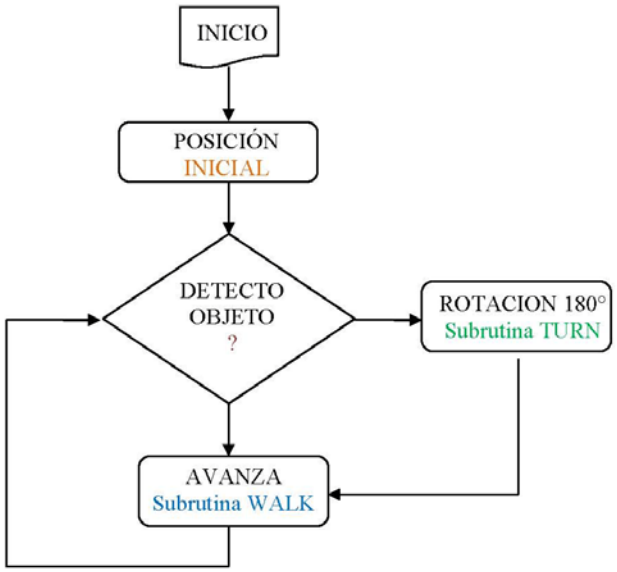

Figura 8.10 Diagrama de flujo

A continuación se describen las subrutinas del programa utilizado para la locomoción del robot:

///CÓDIGO DE DESPLAZAMIENTO, BUSQUEDA Y GIRO 180° ///

// Inicialización y Configuración del Microprocesador

```
#include <18F4550.h>
#fuses
HS,NOWDT,NOPROTECT,NOUSBDIV,CPU
DIV1,VREGEN
#use delay(clock=20000000)
////////////bootloader ////////////
#build(reset=0x800)
#build(interrupt=0x808)
#org 0x0000, 0x07ff void bootloader()
//////////////////////////////////
void main()
{int i;
int j;
output_low  (PIN_B0);
output_low  (PIN_B1);
output_low  (PIN_B2);
output_low  (PIN_B3);
```

// Subrutina de Posición Inicial

```
for (i=1;i<=25;++i)
  {
   output_high(PIN_A0);
   output_high(PIN_A1);
   output_high(PIN_B4);
   output_high(PIN_B5);
   output_high(PIN_B6);
   output_high(PIN_B7);
   delay_us(1300);
   output_low(PIN_A0);
   output_low(PIN_A1);
   output_low(PIN_B4);
   output_low(PIN_B5);
   output_low(PIN_B6);
   output_low(PIN_B7);
   delay_us(18700);
  }
delay_ms (2000);
goto Inicio;
```

// Punto de desición (Lectura de Dato)

```
Inicio:
{
 if( input(PIN_A2) )
   {
   output_high(PIN_A3);
   delay_ms(10);
   goto Turn;
   }
else
  {
   output_low (PIN_A3);
   delay_ms(10);
   goto walk;
   }
```

```
//Subrutina de Avance

WALK:

{
output_high(PIN_B0);
delay_ms(250);

 for (i=1;i<=15;++i)

  {
   output_high (PIN_A1);
   output_high (PIN_B6);
   output_high (PIN_B5);
   delay_us(1700);
   output_low   (PIN_A1);
   output_low   (PIN_B6);
   output_low   (PIN_B5);
   delay_us(18300) }
  delay_ms (250);
  output_low    (PIN_B0);
  delay_ms (100);
  output_high   (PIN_B1);
  delay_ms(280);
  }

for (i=1;i<=15;++i)
  {
   output_high(PIN_A1);
   output_high(PIN_B6);
   output_high(PIN_B5);
   delay_us(1100);
   output_low(PIN_A1);
   output_low(PIN_B6);
   output_low(PIN_B5);
   delay_us(18900);
   }


for (i=1;i<=15;++i)

  {
   output_high(PIN_A0);
   output_high(PIN_B4);
   output_high(PIN_B7);
   delay_us(1700);
   output_low(PIN_A0);
   output_low(PIN_B4);
   output_low(PIN_B7);
   delay_us(18300);
   }
   delay_ms(250);
   output_low  (PIN_B1);
   delay_ms (100);
   output_high (PIN_B0);
    delay_ms(250);

for (i=1;i<=15;++i)
  {
   output_high(PIN_A0);
   output_high(PIN_B4);
   output_high(PIN_B7);
   delay_us(1100);
   output_low(PIN_A0);
   output_low(PIN_B4);
   output_low(PIN_B7);
   delay_us(18900);
   }
   goto Inicio;
   }
```

```
//Subrutina de Giro

TURN:
{  For (j=1; j<=8;++j)
{   output_high(PIN_B0);
  delay_ms(250);
  for (i=1;i<=20;++i)
  {output_high(PIN_A1);
   output_high(PIN_B5);
   delay_us(1800);
   output_low(PIN_A1);
   output_low(PIN_B5);
   delay_us(18200);
   }
   for (i=1;i<=20;++i)
  {output_high(PIN_B6);
   delay_us(1000);
   output_low(PIN_B6);
   delay_us(19000);}
  delay_ms (250);
  output_low  (PIN_B0);
  delay_ms (100);
  output_high (PIN_B1);
  delay_ms(280);
  for (i=1;i<=20;++i)
  {output_high(PIN_A1);
   output_high(PIN_B5);
   delay_us(1000);
   output_low(PIN_A1);
   output_low(PIN_B5);
   delay_us(19000);}
   for (i=1;i<=20;++i)
  {output_high(PIN_B6);
   delay_us(1800);
   output_low(PIN_B6);
   delay_us(18200);}
   for (i=1;i<=20;++i)
  {output_high(PIN_A0);
   output_high(PIN_B4);
   delay_us(1000);
   output_low(PIN_A0);
   output_low(PIN_B4);
   delay_us(19000);}
   for (i=1;i<=20;++i)
  {output_high(PIN_B7);
   delay_us(1800);
   output_low(PIN_B7);
}
   delay_us(18200);}
   delay_ms(250);
   output_low  (PIN_B1);
   delay_ms (100);
   output_high (PIN_B0);
   delay_ms(250);
   for (i=1;i<=20;++i)
  {output_high(PIN_A0);
   output_high(PIN_B4);
   delay_us(1800);
   output_low(PIN_A0);
   output_low(PIN_B4);
   delay_us(18200);
   }
   delay_ms (10);
   for (i=1;i<=20;++i)
  {output_high(PIN_B7);
   delay_us(1000);
   output_low(PIN_B7);
   delay_us(19000);
   }
}
   for (i=1;i<=50;++i)
  {output_high(PIN_A0);
   output_high(PIN_A1);
   output_high(PIN_B4);
   output_high(PIN_B5);
   output_high(PIN_B6);
   output_high(PIN_B7);
   delay_us(1300);
   output_low(PIN_A0);
   output_low(PIN_A1);
   output_low(PIN_B4);
   output_low(PIN_B5);
   output_low(PIN_B6);
   output_low(PIN_B7);
   delay_us(18700);
   }
   delay_ms(100);
   output_low  (PIN_B0);
   output_low  (PIN_B1);
   output_low  (PIN_B2);
   output_low  (PIN_B3);
   goto Inicio;
   }
  }
```