

UNIVERSIDAD DE SONSONATE

Arquitectura de Computadoras

Guía 6 – Servomotor

Instructor: Ricardo González



Facultad de Ingeniería y Ciencias Naturales



Contenido

ServoMotor	3
EJERCICIOS.....	6

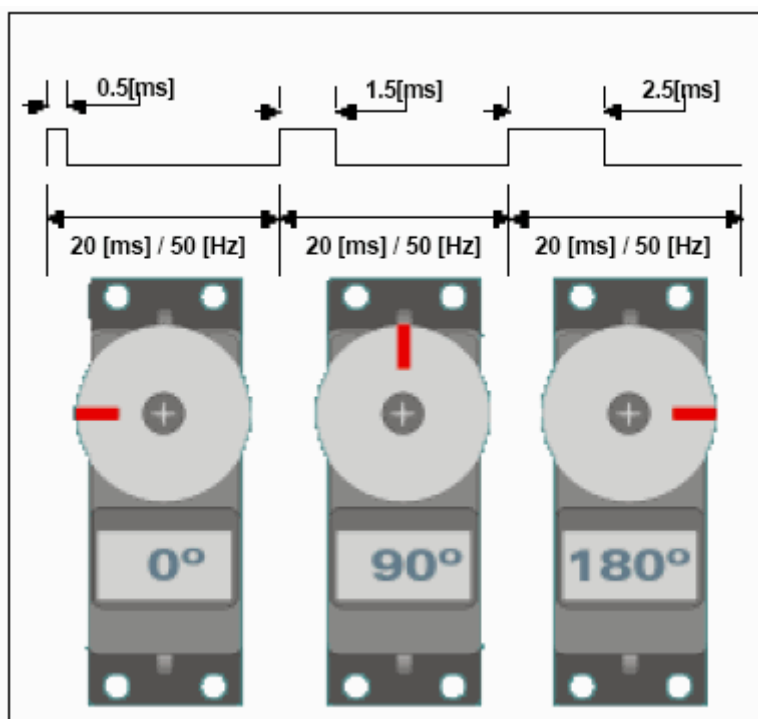


ServoMotor

Un servomotor esta constituido por un motor de corriente continua, unas ruedas dentadas que trabajan como reductoras , lo que le da una potencia considerable, y una pequeña tarjeta de circuito impreso con la electronica necesaria para su control. Existen 2 tipos de servomotor, los cuales son:

- ✓ **ServoMotor de 180°**
- ✓ **ServoMotor de 360°**

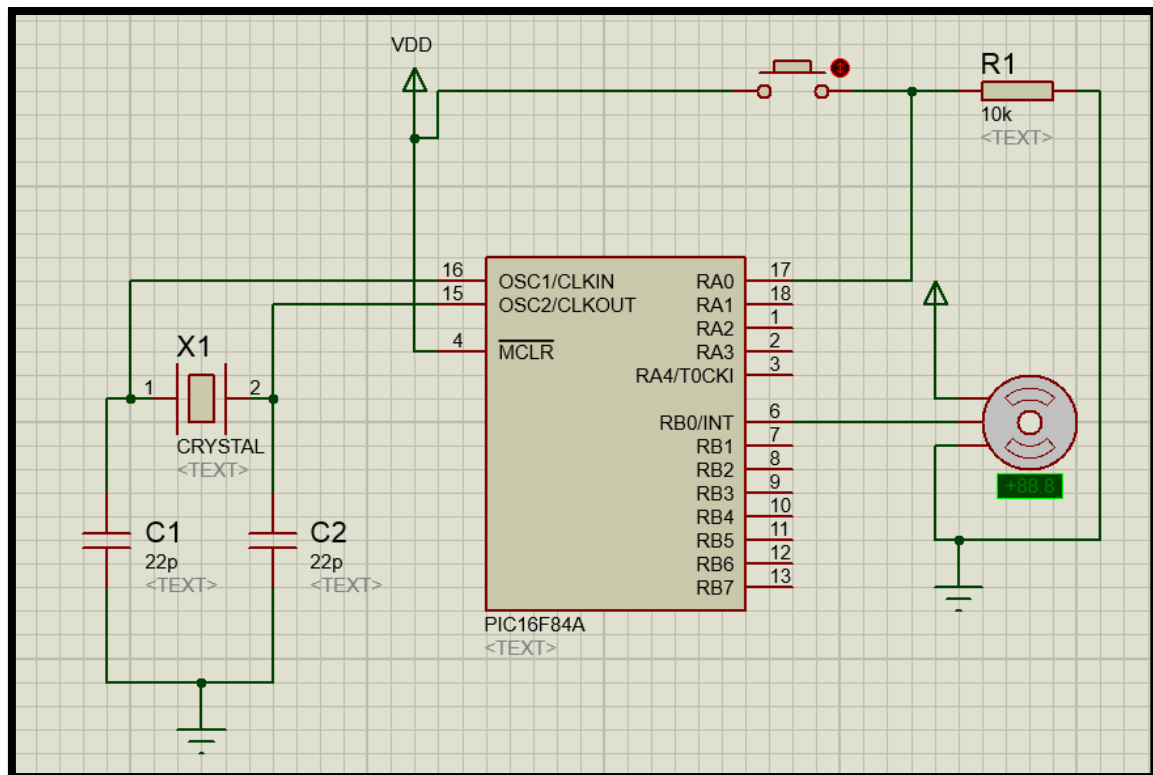
El funcionamiento del servomotor es muy similar en ambos casos ya que la tensión de alimentación de los servos suele estar comprendida entre los 4 y 8 voltios. El control de un servo se limita a indicar en qué posición se debe situar, mediante una señal cuadrada **TTL (Lógica Transistor a Transistor)**, modulada en achura de impulso PWM (*Pulse Width Modulation*). La duración del nivel alto de la señal indica la posición donde queremos poner el eje del motor esto es aplicable a un servomotor de 180°, mientras que con un servomotor de 360°, va permitir controlar el giro y la velocidad del eje del servomotor. El potenciómetro que el servomotor tiene solidariamente al eje del motor indica al circuito electrónico del control interno mediante una retroalimentación, si se está llegando a la posición deseada.



La duración de los pulsos indica el ángulo de giro del motor, cada servomotor tiene su margen de operación, que corresponde al ancho del pulso máximo y mínimo que el servo entiende y que en principio no puede sobrepasar, cabe resaltar que estos modelos varían dependiendo del modelo del servomotor utilizado.



Ejemplo: A continuación se muestra la configuración de un circuito para trabajar con un servomotor utilizando el PIC16F84A, cuando realicemos la configuración del circuito debemos de tomar en cuenta que el **crystal** de cuarzo que utilizamos debe de estar configurado a **1Mhz**, y para no trabajar con valores negativos con el **servomotor**, damos doble click en el motor y modificamos los valores que aparecen **-90, +90** y colocamos los valores de **0 y 180**. Para trabajar con ángulos positivos. La configuración del circuito es la siguiente:



Hay que recordar que para cada servo de determinado fabricante se utiliza determinado tren de pulsos, a continuación se muestra una tabla con alguna de las especificaciones respectivas a los trenes de pulsos que se utilizan en algunos motores:

Fabricante	Duración del pulso ms			Hz	Colocación de cables		
	Min.	Central	Max		+ batt	-batt	Pwm
Futaba	0.9	1.5	2.1	50	Rojo	Negro	Blanco
Hitech	0.9	1.5	2.1	50	Rojo	Negro	Amarillo
Graupner/jr	0.8	1.5	2.2	50	Rojo	Marrón	Naranja
Multiplex	1.05	1.6	2.15	40	Rojo	Negro	Amarillo
Robbe	0.65	1.3	1.95	50	Rojo	Negro	Blanco
Simprop	1.2	1.7	2.2	50	Rojo	Azul	Negro

Como observamos en la tabla la variación del tiempo entre los pulso cambia dependiendo del fabricante, esto quiere decir que cada vez que utilizemos un servo de un fabricante distinto con el cual probamos nuestro firmware será necesario volver a ajustar el tren de pulsaciones.



Continuando con el ejemplo la manipulación de nuestro estará a cargo del siguiente firmware:

```
1  #include <xc.h>
2  #define _XTAL_FREQ 1000000
3  //Declaracion de variables
4  int proceso = 0;
5  void girarmotor(int pulsos);
6
7  void main(void)
8  {
9      //Configuramos los puertos
10     int pulsos = 0;
11     TRISA = 1;          //Configuramos el puerto A como entrada
12     TRISB = 0;          //Configuramos el puerto B como salida
13     PORTA = 0;          //Inicializamos los pines del PUERTO A
14     PORTB = 0;          //Inicializamos los pines del PUERTO B
15
16     while (1){
17         //Validamos si se presiono el boton
18         if(PORTAbits.RA0){
19             proceso = 1;
20             if(pulsos == 3) {
21                 pulsos = 1;
22             }else {
23                 pulsos += 1;
24             }
25             __delay_ms(500);
26         }
27         //Realizamos el tren de pulsos de corriente
28         girarmotor(pulsos);
29     };
30
31
32 void girarmotor(int pulsos){
33     if(proceso == 1){
34         switch(pulsos){
35             case 1: //POSICIONAMOS EL MOTOR SERVO A 90°
36                 //Realizamos pulso de 2ms
37                 for(int i= 0; i < 16; i++ ){
38                     PORTBbits.RB0 = 1;
39                     __delay_ms(1.5);
40                     PORTBbits.RB0 = 0;
41                     __delay_ms(20);
42                 }
43                 proceso = 0;
44                 break;
45             case 2: //POSICIONAMOS EL MOTOR SERVO A 180°
46                 //Realizamos 16 pulso de 4ms
47                 for(int i= 0; i < 16; i++ ){
48                     PORTBbits.RB0 = 1;
49                     __delay_ms(2);
50                     PORTBbits.RB0 = 0;
51                     __delay_ms(20);
52                 }
53                 proceso = 0;
54                 break;
55             case 3: //POSICIONAMOS EL MOTOR SERVO A 180°
56                 //Realizamos 16 pulso de 4ms
57                 for(int i= 0; i < 16; i++ ){
58                     PORTBbits.RB0 = 1;
59                     __delay_ms(0.6);
60                     PORTBbits.RB0 = 0;
61                     __delay_ms(20);
62                 }
63                 proceso = 0;
64                 break;
65             }
66         }
67     }
```



EJERCICIOS.

Para cada uno de los ítems se tiene que realizar en distintos proyectos utilizando el PIC16F84A.

- 1) En este ejercicio manipularemos 5 motores servo, la idea de utilizar esta cantidad de motores, es que simularemos la manipulación de una mano robótica, esto quiere decir que cada uno de los motores servo representara un dedo de la mano robótica, y para manipular cada uno de los motores agregaremos 5 botones que controlaran cada uno de los motores. A continuación se muestra una imagen de lo que se pretende simular:

