

Microcontroladores

Semana 11

Por Kalun José Lau Gan

1

Preguntas previas

- ¿Veremos manipulación de servos?
 - Si, hoy atenderemos ese tema
- ¿Cuáles son los temas pendientes en el curso?
 - Comunicación serial I2C
 - Comunicación serial UART
 - Protocolos e interfaces propietarias (sensores DHT11, 1-wire)
 - Otras interfaces y protocolos (RS485, CAN)
 - PCB
 - Modelado 3D y manufactura aditiva
 - Micros de 32bits (ESP32, Rpi-Pico, STM32, PIC32)

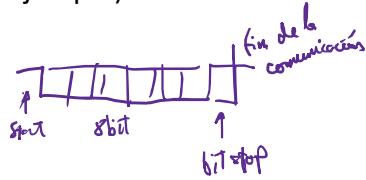
2

Preguntas previas

- ¿Cómo se transmiten los datos en el UART?
 - Usando el registro UxTXB (U1TXB por ejemplo)

```

U1TXB = 'H';
while (U1ERRRbits.TXMTIF == 0);
U1TXB = 'O';
while (U1ERRRbits.TXMTIF == 0);
U1TXB = 'L';
while (U1ERRRbits.TXMTIF == 0);
U1TXB = 'A';
while (U1ERRRbits.TXMTIF == 0);
  
```



```

unsigned int bateria = 16399;
// determina: primero H y luego L
U1TXB = (bateria >> 8) & 0X00FF;
while (U1ERRRbits.TXMTIF == 0);
U1TXB = bateria & 0X00FF;
while (U1ERRRbits.TXMTIF == 0);
  
```

3

El servo

- Elemento electromecánico realimentado (posee un lazo de control cerrado)
- Empleado comúnmente en hobby para radiocontrol de vehículos terrestres, aéreos y acuáticos, para el control de posición (ángulo)
 - Acelerador, freno, dirección vehicular, alerones (flaps), timón, robótica.
- Son clasificados por: tamaño, torque, velocidad, precisión, tamaño



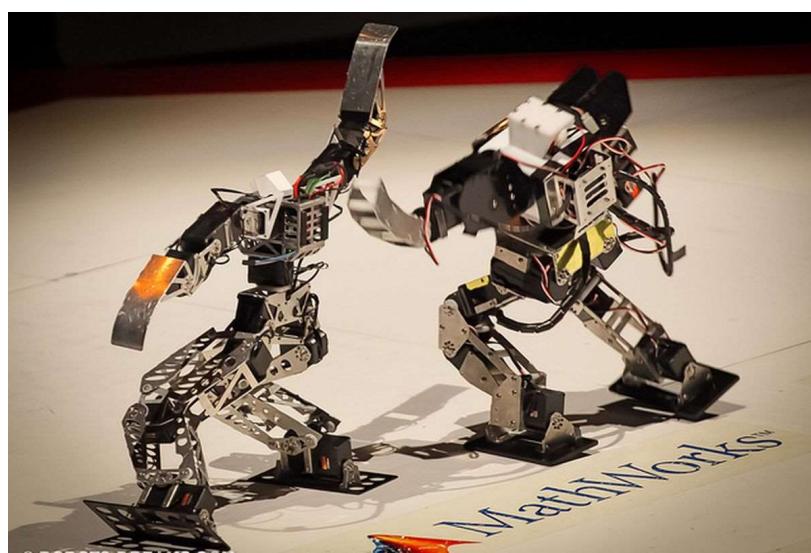
4

Servos en el hobby profesional



5

Servos en el hobby profesional



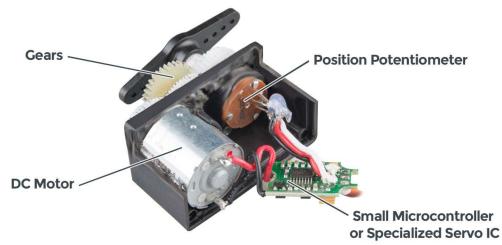
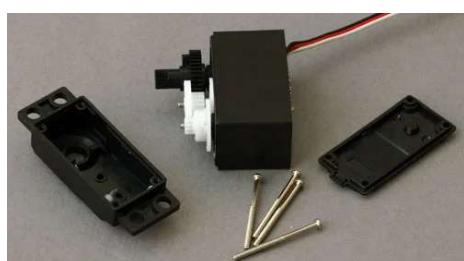
6



7

El servo

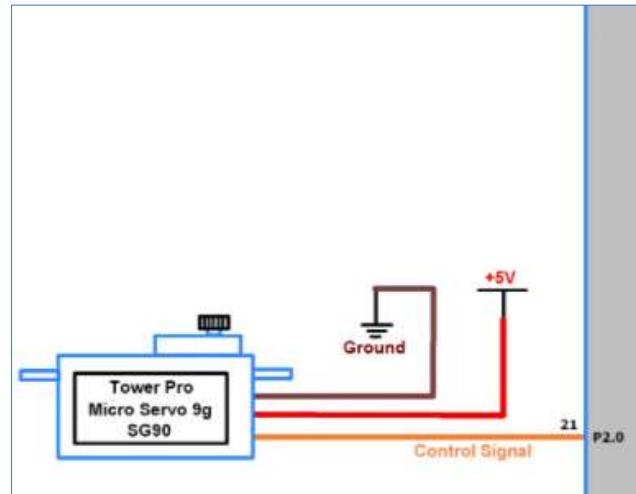
- Internamente posee un motor realimentado con un potenciómetro y mecanismo de reducción.



8

El servo

- Conexión con el microcontrolador:



9

Specs del miniservo sg90:

- De preferencia usar fuente externa para los servomecanismos

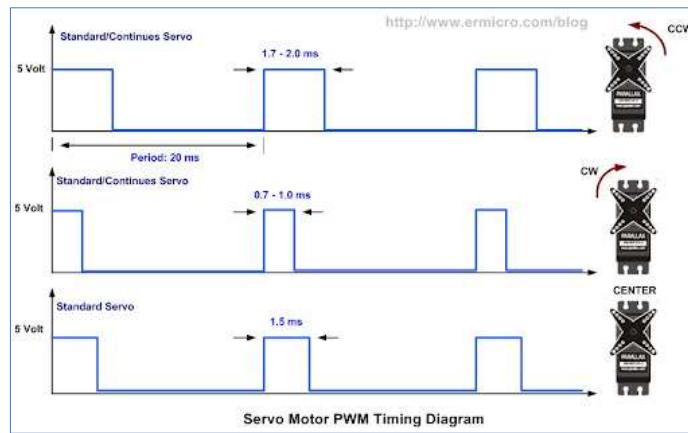
SPECIFICATIONS:

| | Operating Voltage | |
|-------------------------------------|----------------------------|----------------------------|
| | 4.8V | 6.0V |
| Stall Torque | 1.3 kg/cm (18.09 oz/in) | 1.5 kg/cm (20.86 oz/in) |
| Max Speed | 0.12sec/60° | 0.12sec/60° |
| Idle Current | 5 mA | 6 mA |
| No Load Running Current | 100 mA | 120 mA |
| Stall Current | 700 mA | 800 mA |
| Operating Voltage Range | 4.8V to 6V | |
| Pulse Width Range | 900 to 2100 µs | |
| Limit Angle | 120° (900 to 2100 µs) | |
| Dead Band Width | 10 µs | |
| Stop position | 1500 (± 5) µs | |
| CW Rotation Signal Range | 900 to 1500 µs | |
| CCW Rotation Signal Range | 1500 to 2100 µs | |
| Internal Gears | Plastic | |
| Cable Length | 20 cm (7.9") | |
| Dimensions (detailed dimensions) | | |
| Length | 32.6 mm (1.28") | |
| Width | 12.5 mm (0.49") | |
| Height | 27.3 mm (1.08") | |
| Weight (of servo itself) | 9 g (0.32 oz) | |
| Weight (including horns and screws) | 15.1 g (0.53 oz) | |

10

El servo

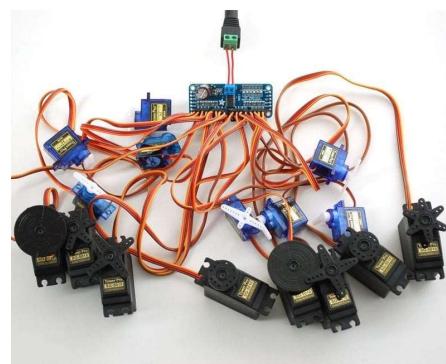
- Modo de funcionamiento:
 - Tren de pulsos de periodo 20ms ($f=50\text{Hz}$)
 - El ancho del pulso (1.0ms – 2.0ms) positivo determinará la posición del eje del servo
 - Al quitarle el tren de pulsos el servo se inactivará



11

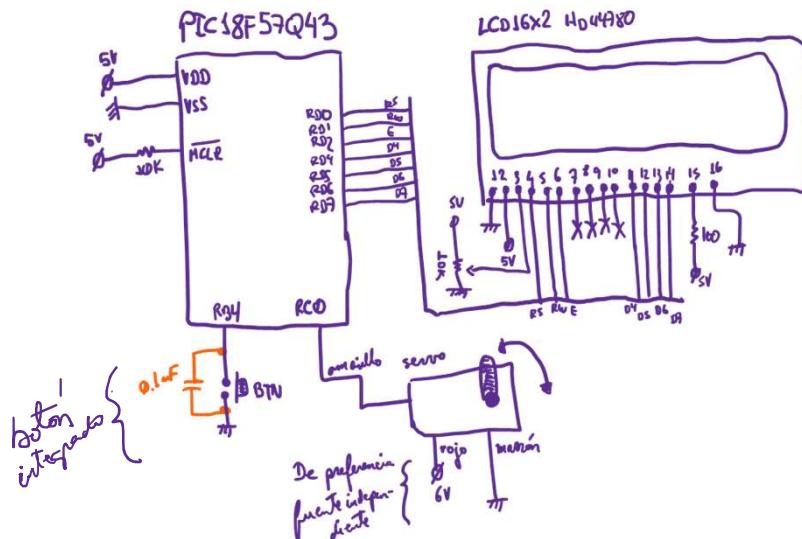
Opciones para obtener la señal requerida para manipular el servo

- Utilizando retardos ✓
- PWM a través del módulo CCP ✗ ✓
- Uso de temporizadores (Timer0)
- PCA9685A (controlador I2C para sistemas con LEDs)



12

Circuito de prueba para manipular un servo con el Curiosity Nano PIC18F57Q43



13

¿Cómo generar la onda cuadrada de 50Hz con el módulo CCP del PIC18F57Q43?

- El CCP en modo PWM nos permitiría obtener dicha señal cuadrada
 - Tenemos que validar si el CCP-PWM permite generar una señal a 50Hz

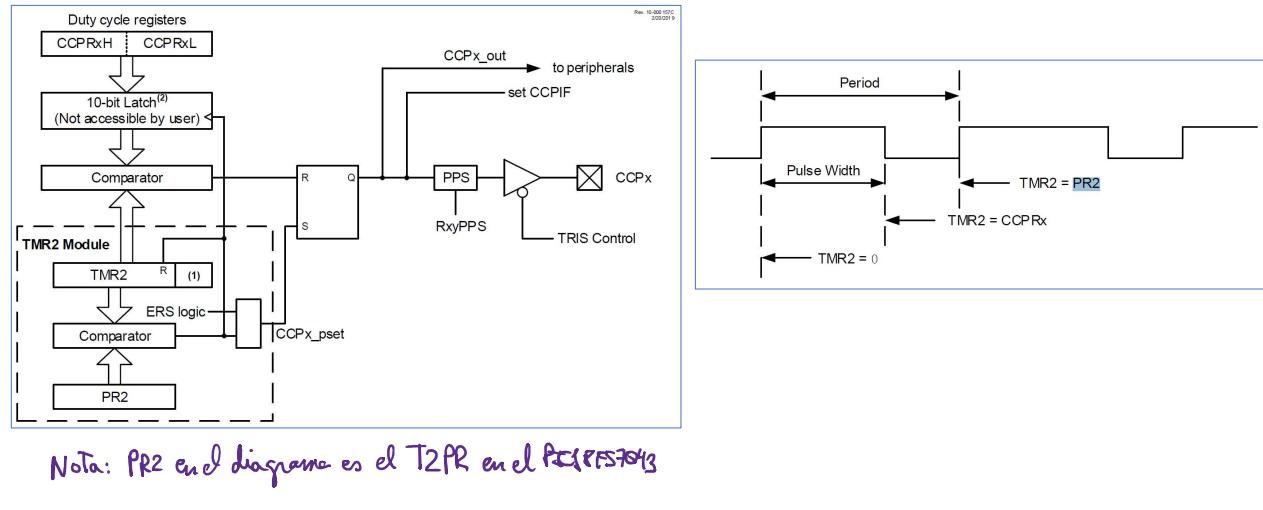
Table 28-1. CCP Mode - Timer Resources

| CCP Mode | Timer Resource |
|----------|--------------------------|
| Capture | Timer1, Timer3 or Timer5 |
| Compare | |
| PWM | Timer2, Timer4 or Timer6 |

14

¿Cómo generar la onda cuadrada de 50Hz con el módulo CCP del PIC18F57Q43?

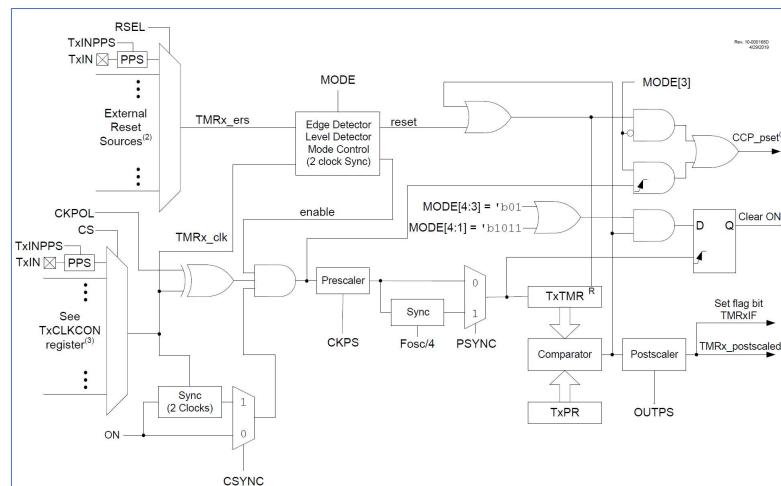
- Diagrama de bloques del modo PWM



15

¿Cómo generar la onda cuadrada de 50Hz con el módulo CCP del PIC18F57Q43?

- En el modo PWM del CCP, el periodo lo determina Timer2:



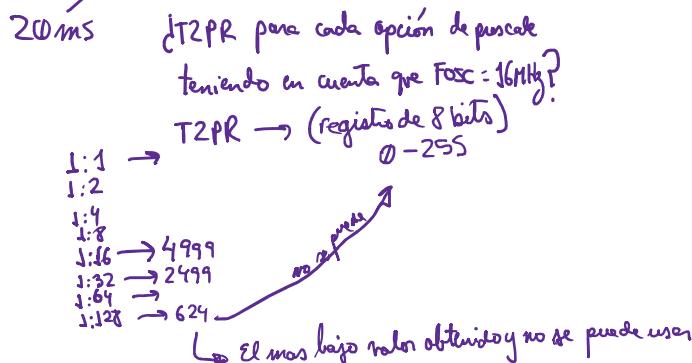
16

¿Cómo generar la onda cuadrada de 50Hz con el módulo CCP del PIC18F57Q43?

- En el modo PWM del CCP, el periodo lo determina Timer2:

Equation 28-1. PWM Period

$$\text{PWM Period} = [(T2PR + 1)] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$



| CKPS[2:0] Timer Clock Prescale Select | |
|---------------------------------------|-----------------|
| Value | Description |
| 111 | 1:128 Prescaler |
| 110 | 1:64 Prescaler |
| 101 | 1:32 Prescaler |
| 100 | 1:16 Prescaler |
| 011 | 1:8 Prescaler |
| 010 | 1:4 Prescaler |
| 001 | 1:2 Prescaler |
| 000 | 1:1 Prescaler |

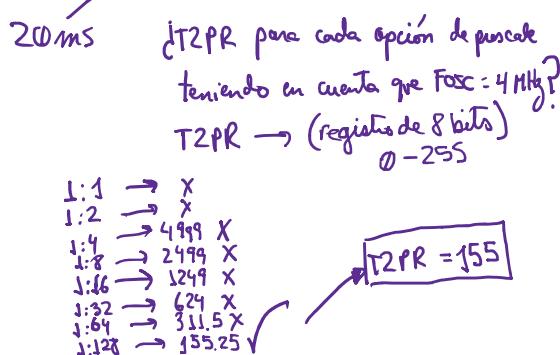
17

¿Cómo generar la onda cuadrada de 50Hz con el módulo CCP del PIC18F57Q43?

- En el modo PWM del CCP, el periodo lo determina Timer2:

Equation 28-1. PWM Period

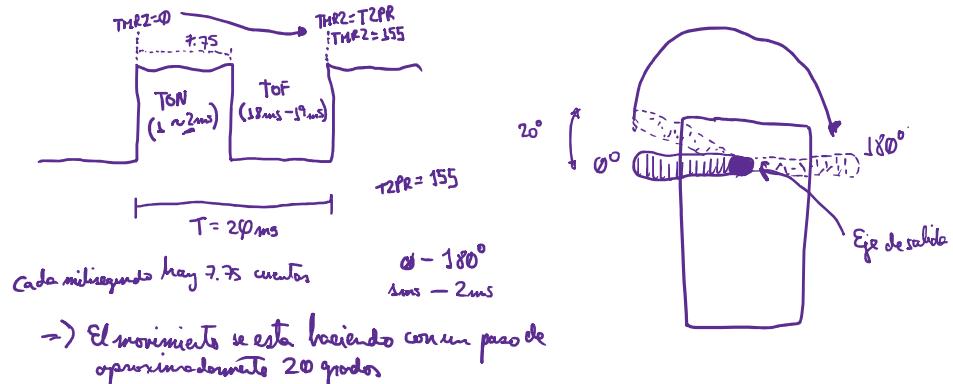
$$\text{PWM Period} = [(T2PR + 1)] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$



| CKPS[2:0] Timer Clock Prescale Select | |
|---------------------------------------|-----------------|
| Value | Description |
| 111 | 1:128 Prescaler |
| 110 | 1:64 Prescaler |
| 101 | 1:32 Prescaler |
| 100 | 1:16 Prescaler |
| 011 | 1:8 Prescaler |
| 010 | 1:4 Prescaler |
| 001 | 1:2 Prescaler |
| 000 | 1:1 Prescaler |

18

Resolución del duty cycle de tu PWM

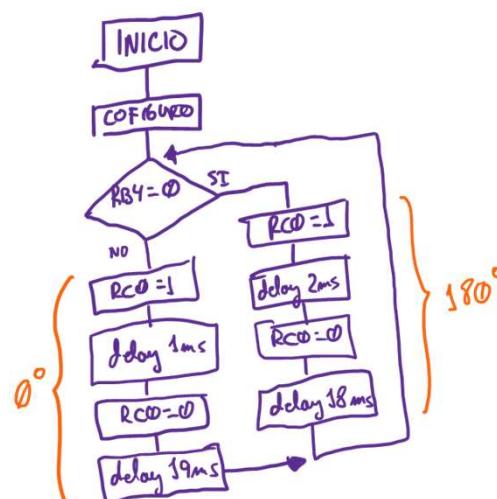
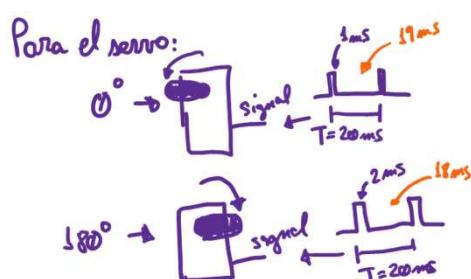


- En consecuencia: No se puede usar el CCP en modo PWM para controlar un servo porque se va a mover escalonadamente. Cada cambio de valor en el duty cycle determinado por CCPRx lo va a realizar en 20 grados angulares del eje del servomecanismo.

19

Estrategia de usar retardos - `__delay_us();`

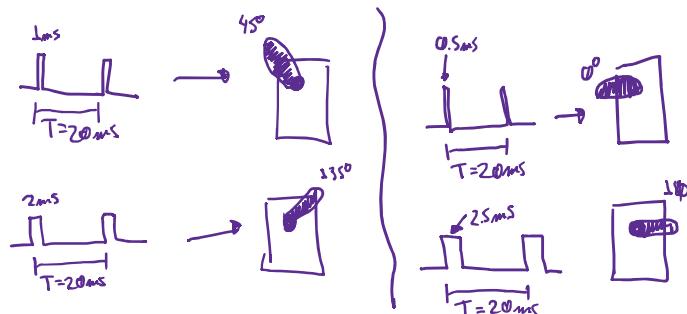
- Usando el botón, intercambiar entre 0° y 180° el servomecanismo



20

El servo no se mueve los 180°!

- Los servos sg90 requieren de un rango mas amplio del ancho del pulso positivo para poder movilizarse mas grados



21

Código ejemplo empleando retardos

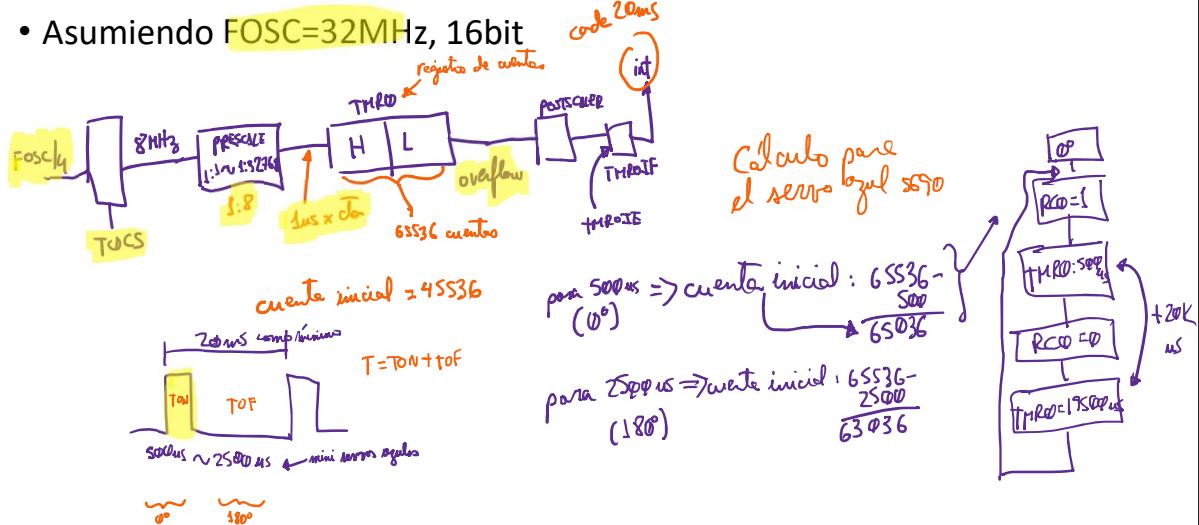
```

1 #include <xc.h>
2 #include "cabecera.h"
3 #include "LCD.h"
4 #define _XTAL_FREQ 32000000UL
5
6 void configuro(void){
7     //configuracion del oscilador
8     OSCCON1 = 0x60;
9     OSCFRQ = 0x06;           //HFINTOSC a 32MHz
10    OSCEN = 0x40;
11    //configuracion de las E/S
12    TRISBbits.TRISB4 = 1;   //RB4 entrada
13    ANSELBbits.ANSELB4 = 0; //RB4 digital
14    WPUBbits.WPUB4 = 1;    //RB4 con pullup activado
15    TRISCbits.TRISCO = 0;  //RC0 salida
16    ANSELCbits.ANSELCO = 1; //RC0 digital
17 }
18
19 void lcd_init(void){
20     TRISD = 0x00;
21     ANSELD = 0x00;
22     LCD_CONFIG();
23     __delay_ms(22);
24     BORRAR_LCD();
25     CURSOR_HOME();
26     CURSOR_ONOFF(OFF);
27 }
28
29 void main(void) {
30     configuro();
31     lcd_init();
32     POS_CURSOR(1,0);
33     ESCRIBE_MENSAJE("Servo UPCino", 12);
34     while(1) {
35         if(PORTBbits.RB4 == 0){
36             LATCbits.LATCO = 1;
37             __delay_us(500);
38             LATCbits.LATCO = 0;
39             __delay_us(19500);
40         }
41         else{
42             LATCbits.LATCO = 1;
43             __delay_us(2500);
44             LATCbits.LATCO = 0;
45             __delay_us(17500);
46         }
47     }
48 }
```

22

Empleando el Timer0 para temporizar 20ms del periodo necesario para el servo

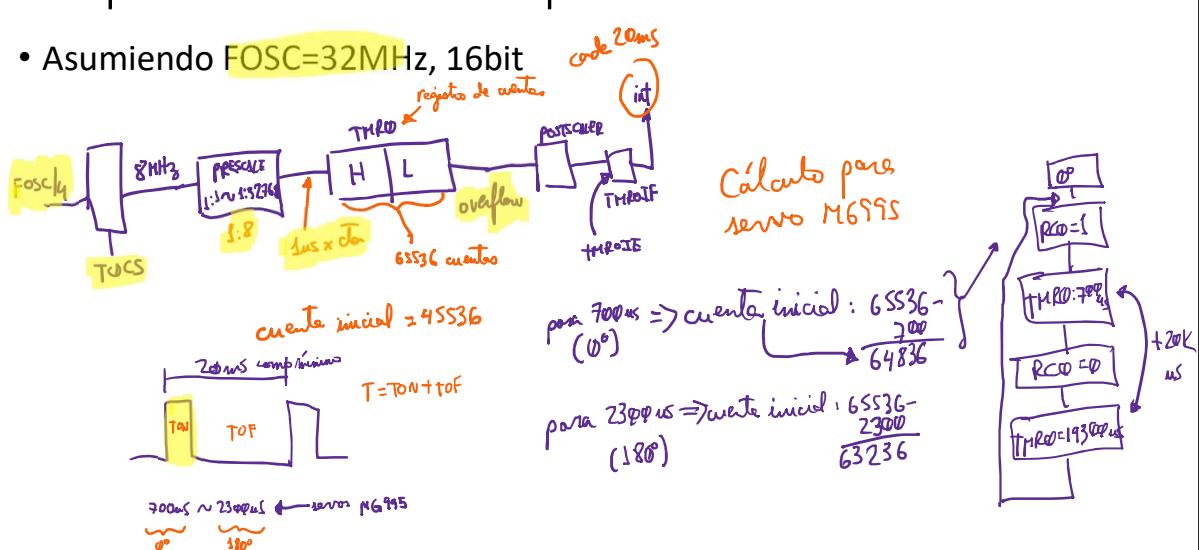
- Asumiendo $\text{FOSC}=32\text{MHz}$, 16bit



23

Empleando el Timer0 para temporizar 20ms del periodo necesario para el servo

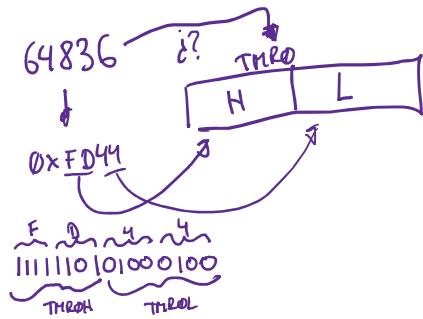
- Asumiendo $\text{FOSC}=32\text{MHz}$, 16bit



24

¿Cómo ingreso un número de cuenta inicial que esta en formato decimal a los registros de cuenta del Timer0?

Se tiene un número 64836 que deseamos colocar como cuenta inicial



$$THROH = (64836 \gg 8) \& 0x00FF$$

\Downarrow
00000000|11111101
00000000|000000111111
—————
00000000|11111101

$$THROL = 64836 \& 0x00FF$$

\Downarrow
1111110101000100
0000000011111111
—————
00000000010000100

25

La relación T, TON, TOF con cuentas iniciales y posición de servo.

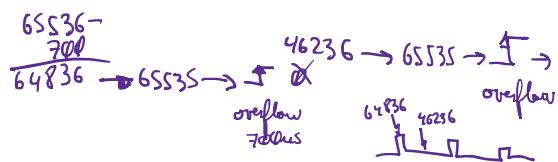
$$T = TON + TOF$$

$$T = 200000\mu s$$

$1\mu s \times 200$

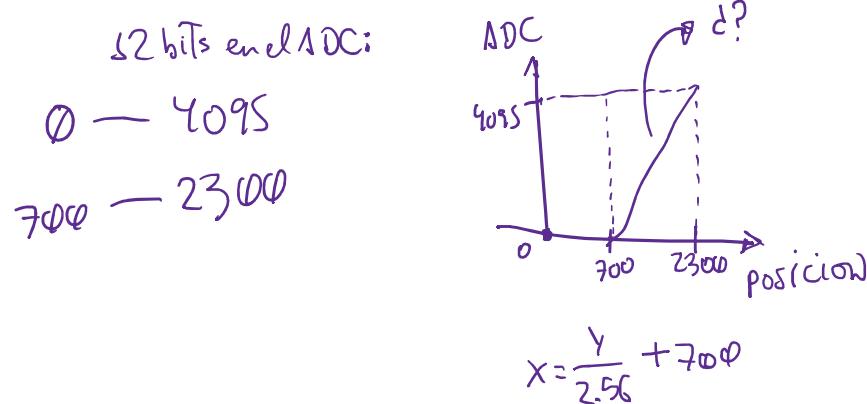
| 0° TON | $7000\mu s$ | X | TOF | $19300\mu s$ | $200000 - X$ |
|------------------|-------------|---|-----|--------------|--------------|
|------------------|-------------|---|-----|--------------|--------------|

| 180° TON | 23000 | X | TOF | 17700 | $200000 - X$ |
|--------------------|---------|---|-----|---------|--------------|
|--------------------|---------|---|-----|---------|--------------|



26

Relación ADC con temporizado de servo



27

Cuestionario:

- Ya que se ha analizado el Timer0 y el Timer3 como fuente de tiempo para obtener los períodos de un servo. Es posible manipular dos servos, uno con el Timer0 y otro con el Timer3 junto con el manejo adecuado de las interrupciones.

28

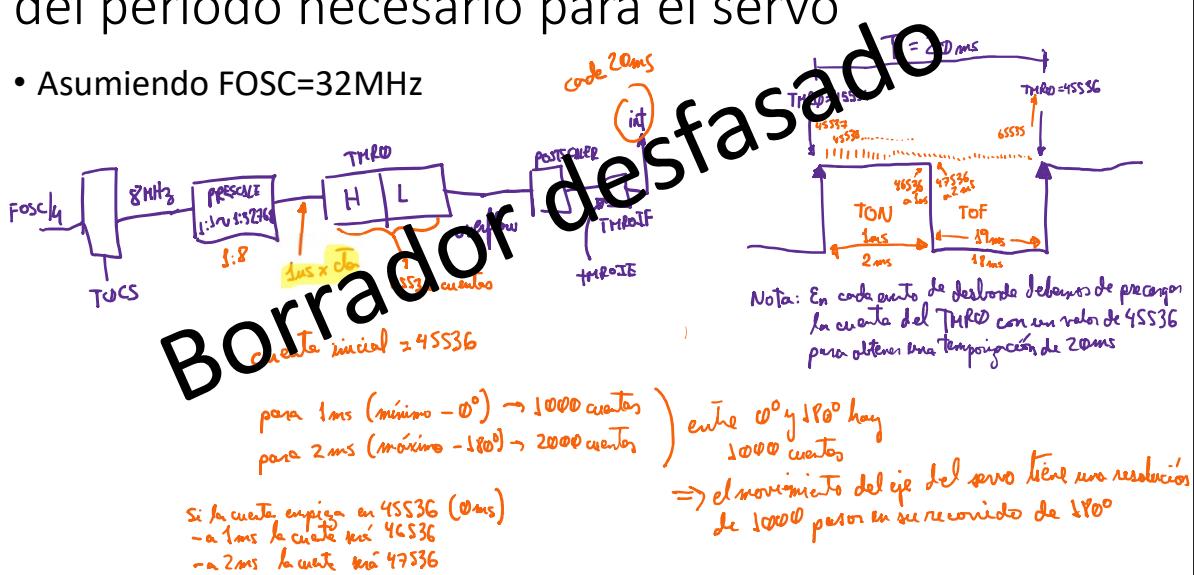
Fin de la sesión

- Destinar una hora el sábado y una hora el domingo para repasar el curso.

29

Empleando el Timer0 para temporizar 20ms del periodo necesario para el servo

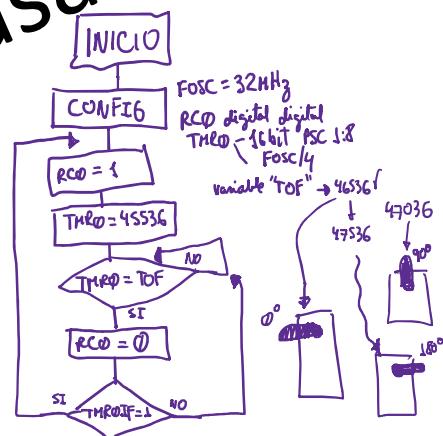
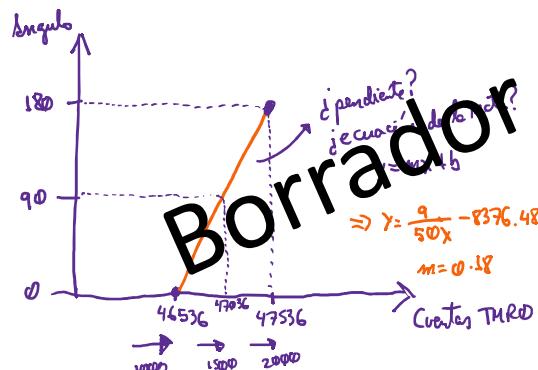
- Asumiendo FOSC=32MHz



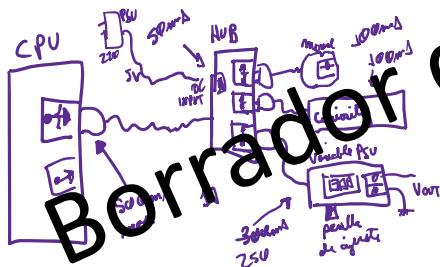
30

Empleando el Timer0 para temporizar 20ms del periodo necesario para el servo

- Escalamiento ángulo de servo vs cuentas del Timer0



31



32