

**UNIVERSIDAD POLITÉCNICA**  
**DE LA ZONA METROPOLITANA DE GUADALAJARA**

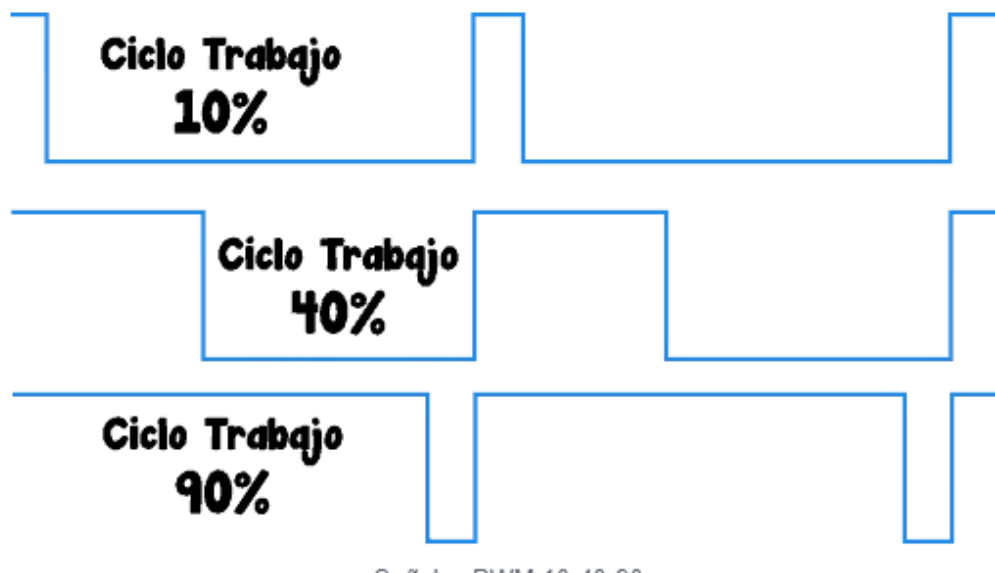
# 3\_2\_PWM

MOISES EMANUEL MARTINEZ NOYOLA

PWM son siglas en inglés que significan *Pulse Width Modulation* y que lo podemos traducir a español como Modulación de ancho de pulso.

La modulación de ancho de pulso está formada por una señal de onda cuadrada que no siempre tiene la misma relación entre el tiempo que esta en alto y el tiempo que está en bajo.

En la siguiente imagen vemos una señal que varía entre 5 voltios y 0 voltios. A lo largo del tiempo la señal varía entre dos valores de tensión. Durante un tiempo determinado la señal se encuentra en el nivel alto ( en este caso 5v ) y durante otro periodo de tiempo se encuentra en el segundo valor de tensión (en este caso 0v).



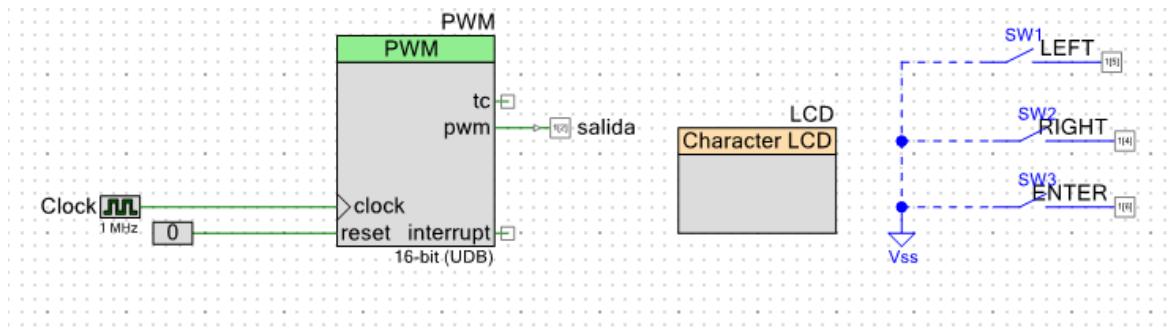
El tiempo que la señal se encuentra en el nivel alto lo denominamos como tiempo on (  $T_{on}$  ) mientras que el tiempo que está en nivel bajo lo denominamos tiempo off (  $T_{off}$  ). La suma del tiempo on y el tiempo off es el periodo de la señal (  $T$  ).

Y como en toda señal periódica, el inverso de del periodo (  $1 / T$  ) es la frecuencia de la señal.

## Código en utilizado en c

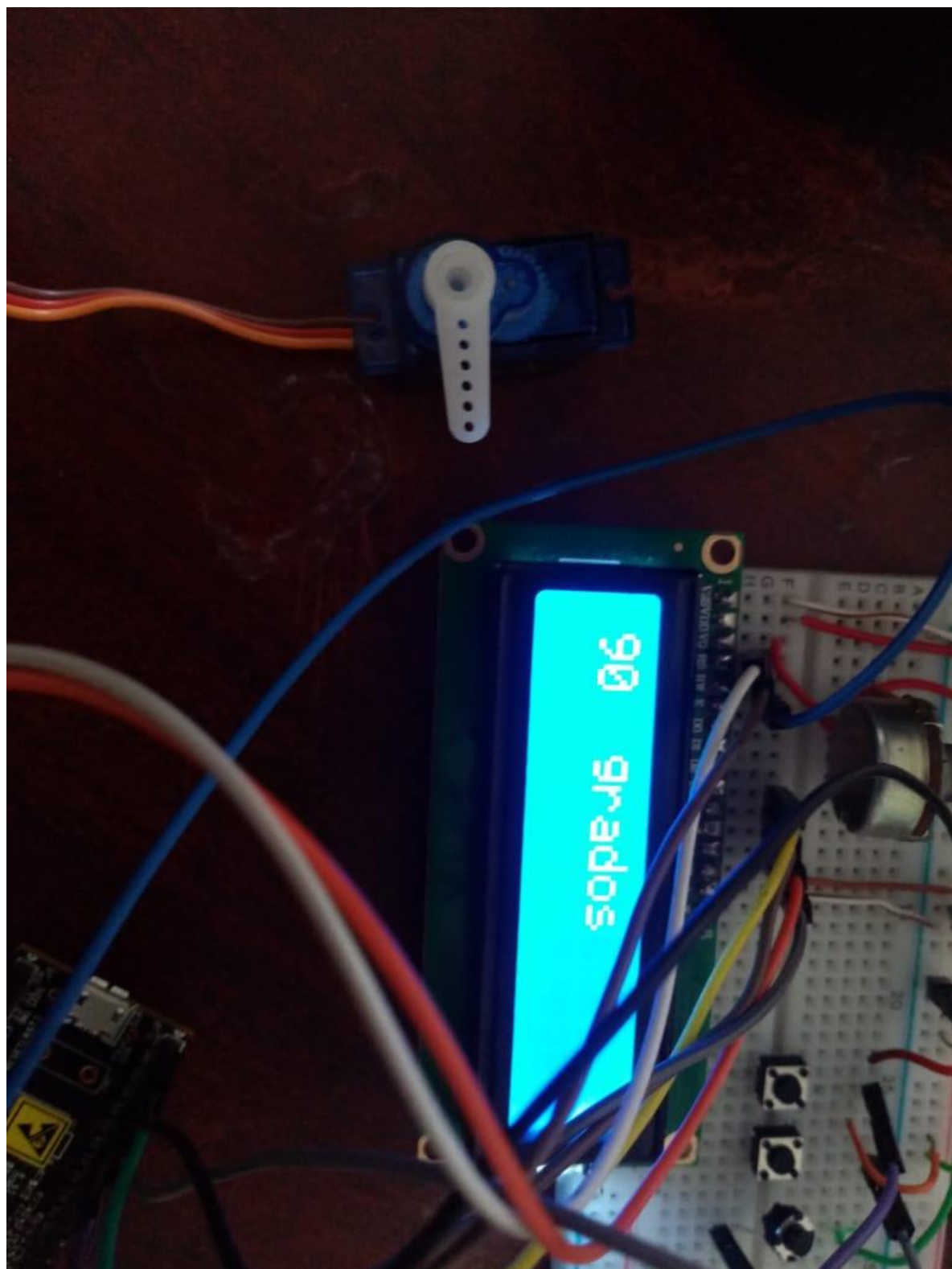
```
1
2 #include "project.h"
3
4 int main(void)
5 {
6     PWM_Start();
7     LCD_Start();
8     LCD_ClearDisplay();
9     CyGlobalIntEnable;
10
11     int x= 1250;           //90°
12     int y= 83;            //incrementos de 10°
13     int z= ((x-500)*180)/1494; //Operación para convertir a grados
14
15
16     for(;;)
17     {
18         LCD_ClearDisplay();
19         LCD_Position(0,5);
20         LCD_PrintString("Grados");
21         LCD_Position(1,8);
22         LCD_PrintNumber(z);
23         CyDelay(600);
24
25         if (RIGHT_Read()){
26             x= x+y;
27             z= ((x-500)*180)/1494;
28             CyDelay(600);
29         }
30
31         //-----
32         LCD_ClearDisplay();
33         LCD_Position(0,5);
34         LCD_PrintString("Grados");
35         LCD_Position(1,8);
36         LCD_PrintNumber(z);
37         CyDelay(600);
38
39         if (LEFT_Read()){
40             x= x-y;
41             z= ((x-500)*180)/1494;
42             CyDelay(600);
43         }
44
45         if (ENTER_Read()){
46             PWM_WriteCompare(x);
47             LCD_ClearDisplay();
48             LCD_Position(0,5);
49             LCD_PrintString("Grados");
50             LCD_Position(1,8);
51             LCD_PrintNumber(z);
52             CyDelay(600);
53         }
54     }
55 }
56
```

## Topdesing



## Funcionamiento en físico







## Conclusión.

A la hora de hacer funcionar nuestro robot, todo lo realizado en esta práctica nos podría ser muy útil para posicionar nuestro robot, pero en esta ocasión realizamos la practica con un servo motor y los motores que utilizaremos son a pasos, por lo que, lo aprendido de este trabajo no lo aplicaremos a nuestro proyecto por el momento.

La podríamos utilizar para accionar alguna herramienta en nuestro robot.

## Referencias

Enrique Gomez. (2017). PWM. 7/03/290, de Digital Sitio web:  
<https://www.rinconingenieril.es/que-es-pwm-y-para-que-sirve/>