



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

Fabian Canales Ochoa

8-A

Ing. Mecatrónica

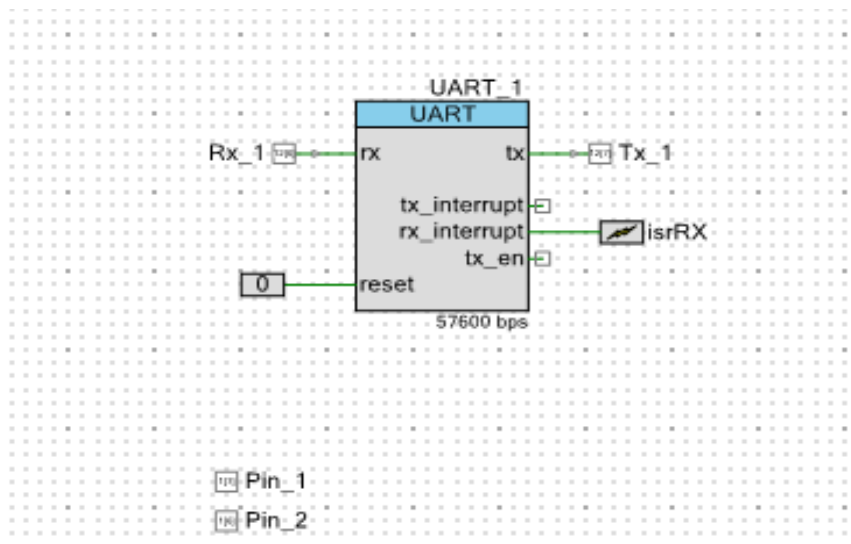
Sistemas Embebidos

Practica UART

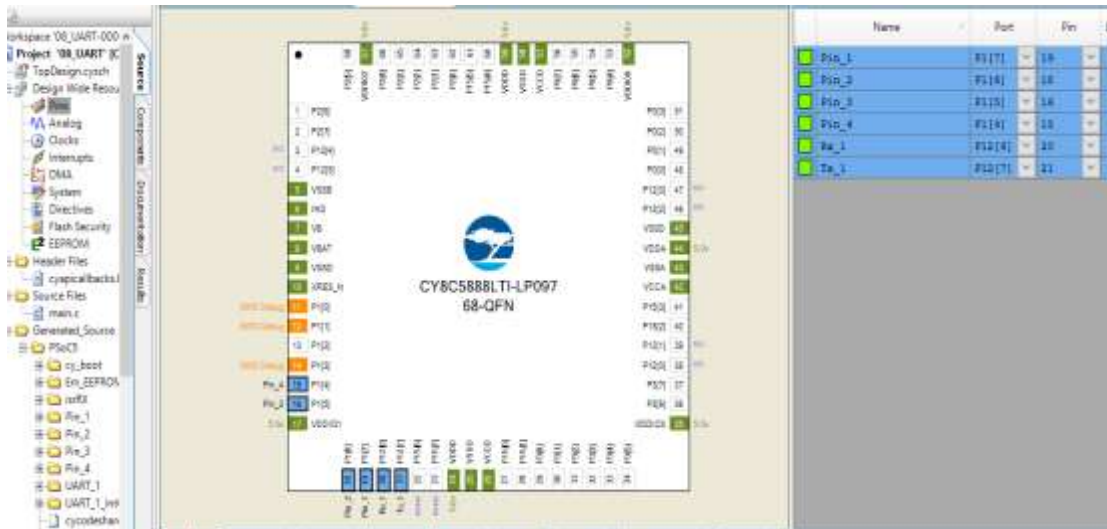
Introducción

El objetivo de la práctica es hacer una comunicación entre la comunicación de la Psco un motor a pasos y la terminal PUTTY para lograr hacer que el motor girara 360 y -360 también que indique el proceso del motor.

- 1- La primera parte de la practica fue realizar el diagrama para el puerto UART:



En este caso solo se necesitaron dos pines de salida ya que el motor va de la mano con un driver que controla el motor así que



solo se necesita un pulso y dirección por eso los dos pines. Esta es la dirección de los pines hacia mi psoc:

2- Después compilamos, ya que todo esté bien pasamos a la programación.

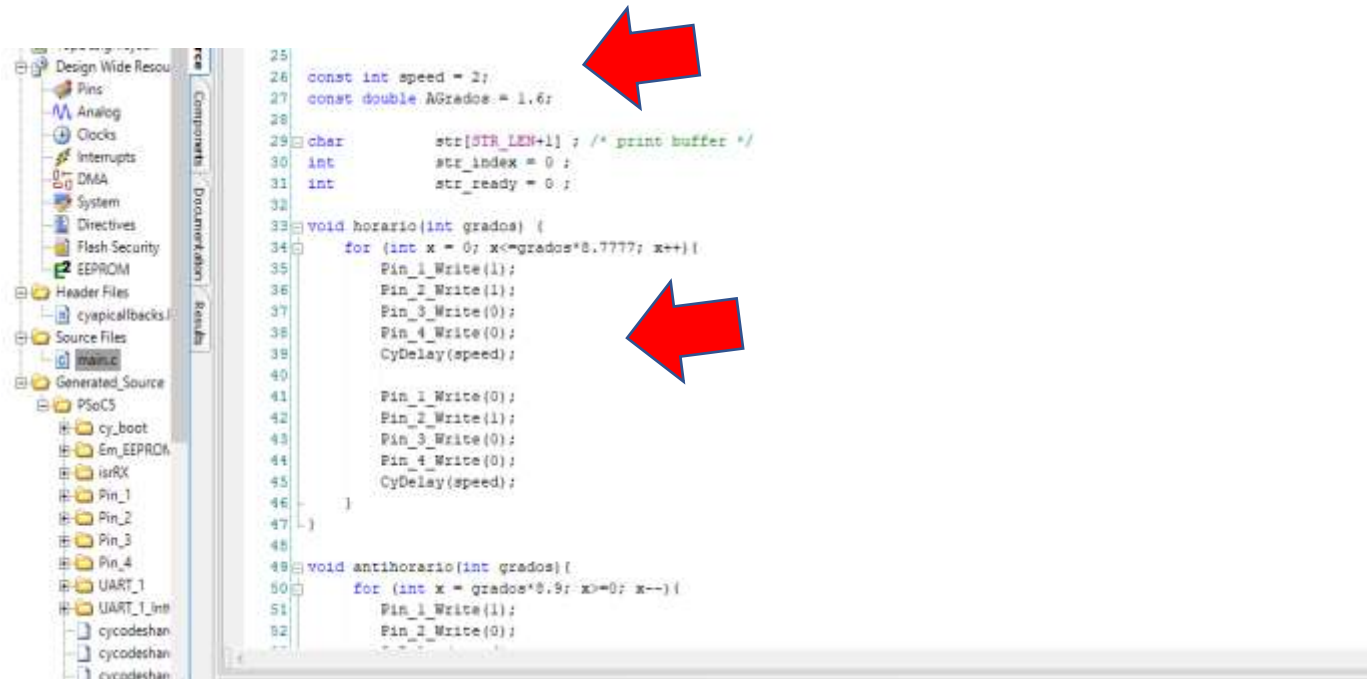
```

1  //
2  //
3  // Copyright YOUR COMPANY, THE YEAR
4  // All Rights Reserved
5  // UNPUBLISHED, LICENSED SOFTWARE.
6  //
7  // CONFIDENTIAL AND PROPRIETARY INFORMATION
8  // WHICH IS THE PROPERTY OF your company.
9  //
10 //
11 //
12 #include "project.h"
13 #include <string.h>
14 #include <stdlib.h>
15 #include <stdio.h>
16 #include <ctype.h>
17
18 #define RX_LEN 64
19 #define RX_BUF_LEN 128
20
21 #define SPACE ' '
22 #define TAB '\t'
23 #define CR '\r'
24 #define LF '\n'
25
26 const int speed = 2;
27 const double AGrados = 1.6;
28
29 char str[RX_LEN+1]; /* print buffer */
30 int str_index = 0;
31 int str_ready = 0;
32

```


Esas son las librerías necesarias para poder hacer la comunicación de la psoc y putty.

Esta parte del programa es para controlar la velocidad y precisión del motor y darles control a los pines de salida:

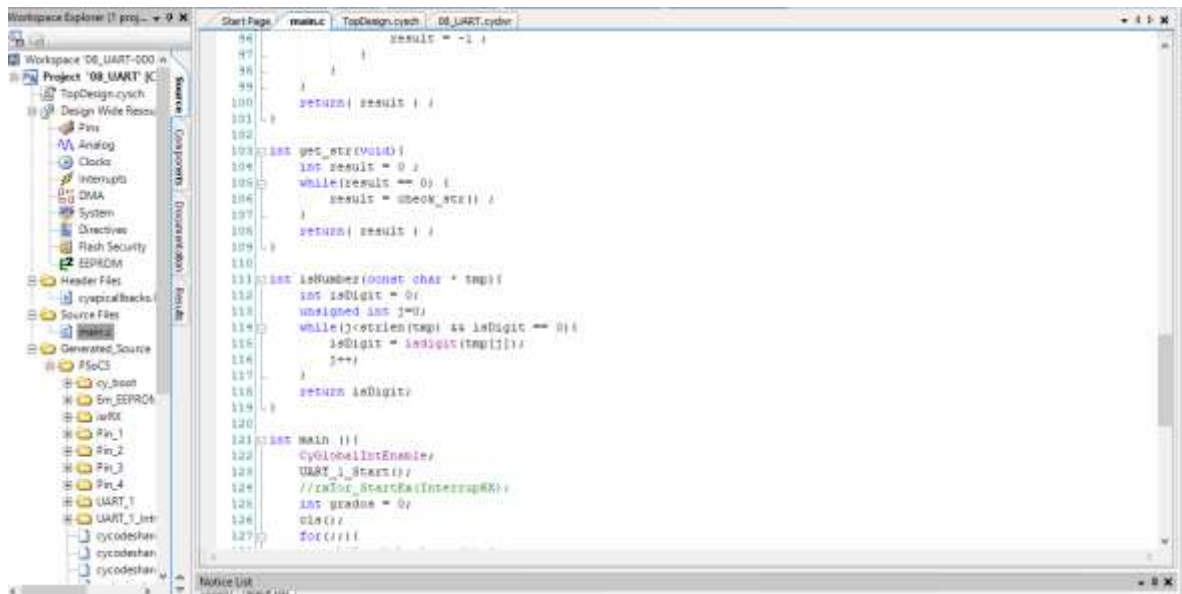


```
25
26 const int speed = 2;
27 const double AGrados = 1.6;
28
29 char    str[STR_LEN+1] ; /* print buffer */
30 int     str_index = 0 ;
31 int     str_ready = 0 ;
32
33 void horario(int grados) {
34     for (int x = 0; x<=grados*8.7777; x++){
35         Pin_1_Write(1);
36         Pin_2_Write(1);
37         Pin_3_Write(0);
38         Pin_4_Write(0);
39         CyDelay(speed);
40
41         Pin_1_Write(0);
42         Pin_2_Write(1);
43         Pin_3_Write(0);
44         Pin_4_Write(0);
45         CyDelay(speed);
46     }
47 }
48
49 void antihorario(int grados){
50     for (int x = grados*8.9; x>0; x--){
51         Pin_1_Write(1);
52         Pin_2_Write(0);
53     }
54 }
```

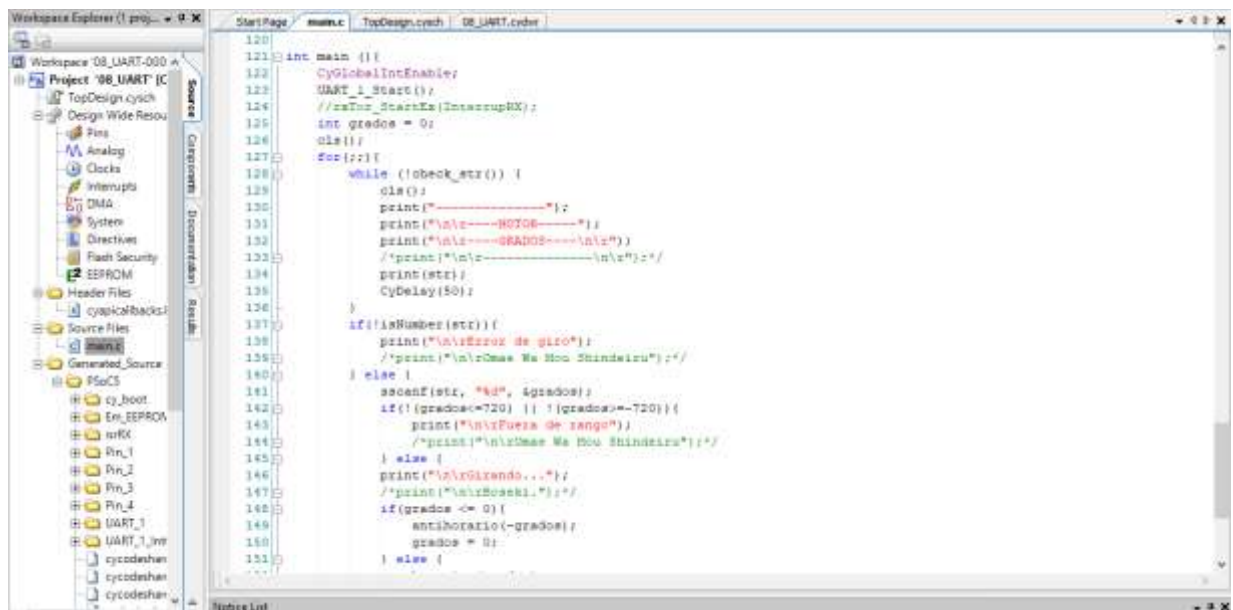
Las siguientes imágenes son lo que resta del código:



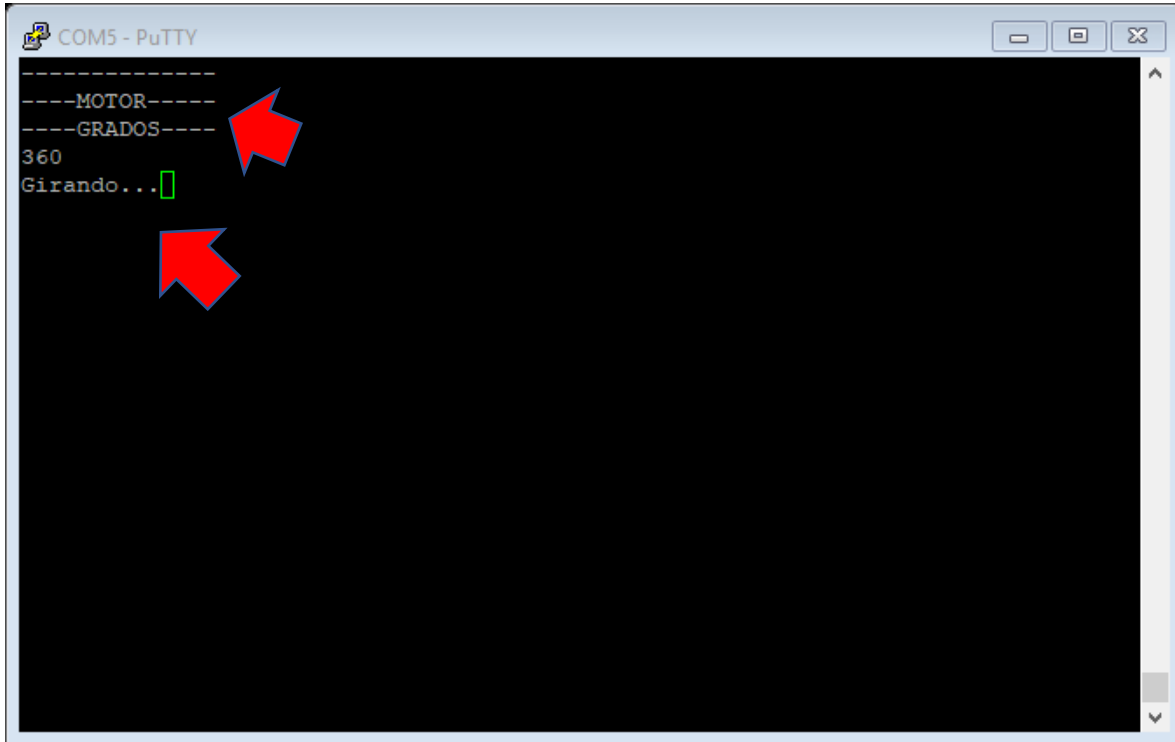
```
40
41 inline int is_delimiter(char c){
42     int result = 0 ;
43     switch(c) {
44         case CR:
45         case LF:
46         case TAB:
47         case SPACE:
48             result = 1 ;
49             break ;
50     }
51     return result ;
52 }
53
54 void cile(){
55     UART_1_PutString("%03d\r\n",0);
56 }
57
58 void print(char *str){
59     UART_1_PutString(str) ;
60 }
61
62 int check_str(void){
63     int result = 0 ;
64     while((result = UART_1_GetByte()) < 0) {
65         result = is_delimiter(c) ;
66         if (result) { /* a string delimiter was detected */
67             str_index = 0 ;
68             str_ready = 0 ;
69         } else { /* still in the middle of a string */
70             str_index++ ;
71             str_ready = 1 ;
72         }
73     }
74 }
```



Y ya por último ponemos la siguiente programación para que se vea en la terminal de putty os datos que se desea:



Como se muestra en la siguiente imagen, esta es la terminal putty y nos indica que es un motor grados y en este caso le pusimos 360 grados y nos indica que está realizando el proceso (Girando)



```
COM5 - PuTTY
-----
---MOTOR---
---GRADOS---
360
Girando...█
```

Y estas son fotos del circuito ya armado;



Conclusión.

Fue difícil mas que nada en la programación y que son cosas avanzadas en el caso de programar tuve que requerir de ayuda para realizar mi programación y el armado del motor, pero los resultados fueron muy buenos ya que se podrían hasta utilizar para el control de los motores de nuestro robot.