



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

Brian Oswaldo Ramos Chavez

Practica UART

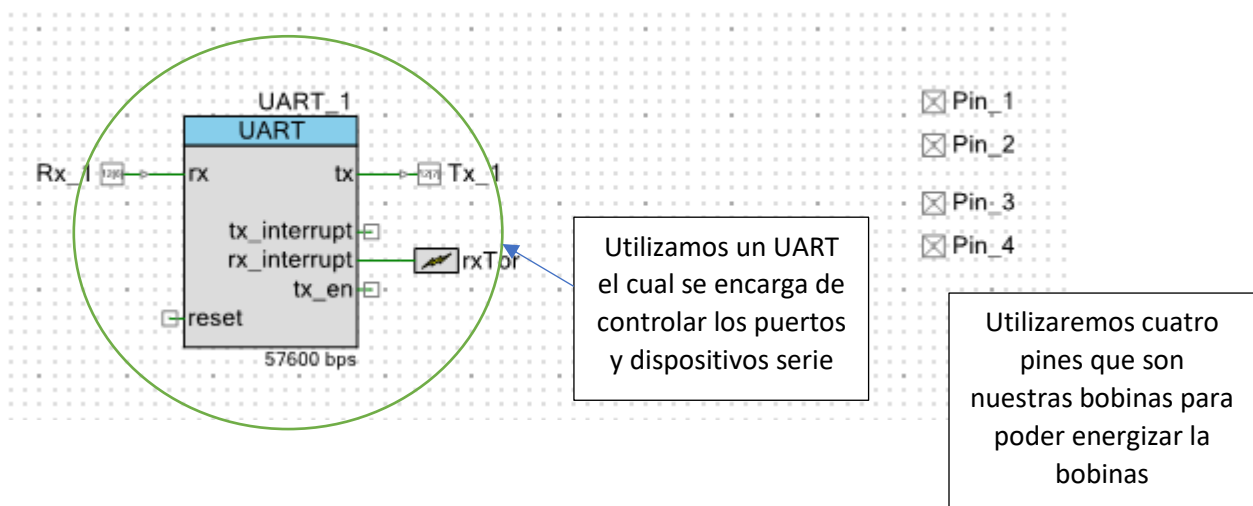
Objetivos: Mover un motor a pasos con ayuda de la PSoC y un controlador de motor y ejecutar cualquier grado que se le ponga en la consola.

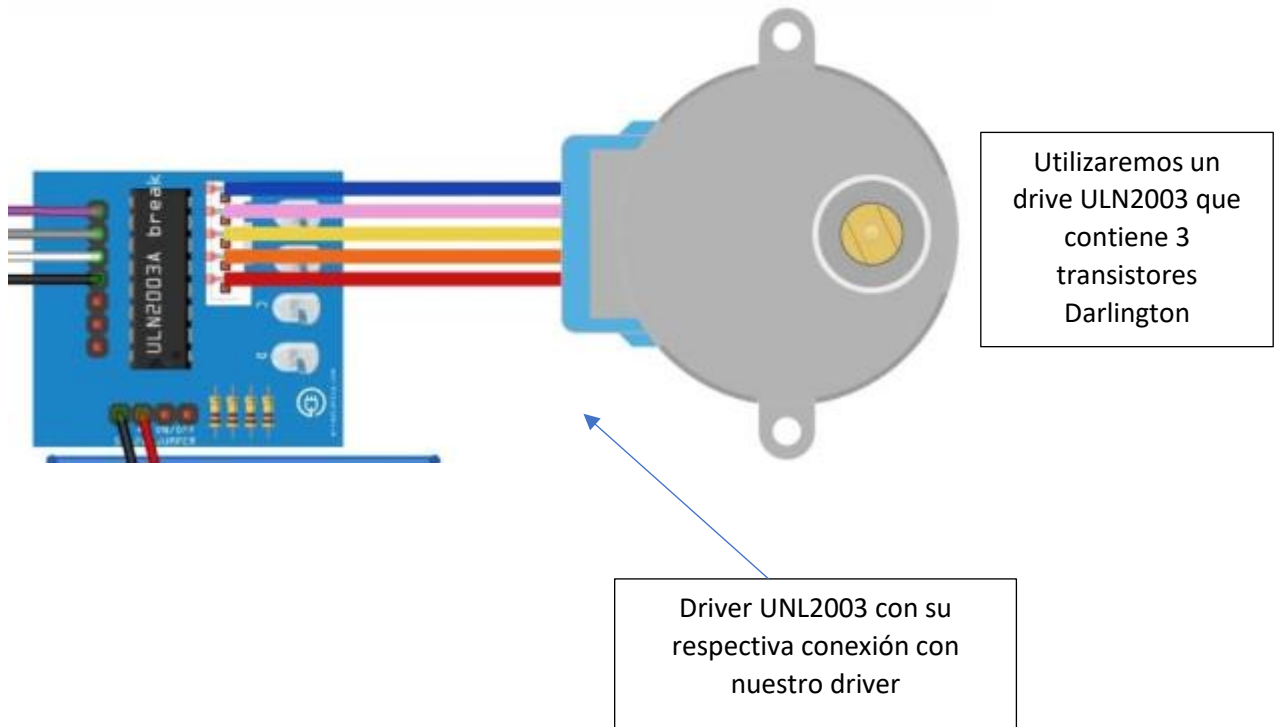
Material:

- **Protoboard**
- **Cables**
- **1 Driver UNL2003**
- **1 Motor paso a paso 28BYJ-48**

Desarrollo:

Crearemos nuestro diagrama en PSoC creator:





El controlador del UART es el componente clave del subsistema de comunicaciones series de una computadora. El UART toma bytes de datos y transmite los bits individuales de forma secuencial. En el destino, un segundo UART reensambla los bits en bytes completos. La transmisión serie de la información digital (bits) a través de un cable único u otros medios es mucho más efectiva en cuanto a costo que la transmisión en paralelo a través de múltiples cables. Se utiliza un UART para convertir la información transmitida entre su forma secuencial y paralela en cada terminal de enlace.

Ahora comenzaremos con la programación de nuestro circuito:

```

void horario(int grados) {
    for (int x = 0; x<=grados * AGrados; x++){
        Pin_1_Write(0);
        Pin_2_Write(0);
        Pin_3_Write(0);
        Pin_4_Write(1);
        CyDelay(vel);

        Pin_1_Write(0);
        Pin_2_Write(0);
        Pin_3_Write(1);
        Pin_4_Write(0);
        CyDelay(vel);

        Pin_1_Write(0);
        Pin_2_Write(1);
        Pin_3_Write(0);
        Pin_4_Write(0);
        CyDelay(vel);

        Pin_1_Write(1);
        Pin_2_Write(0);
        Pin_3_Write(0);
        Pin_4_Write(0);
    }
}

```

Utilizamos un ciclo for para poner en marcha tanto en sentido horario o antihorario depende la elección del usuario, se puede visualizar que utilizamos una secuencia de una fase es decir encendemos únicamente una bobina por vez.

```

] void antihorario(int grados){
]   for (int x = grados * AGrados; x>=0; x--){
      Pin_1_Write(1);
      Pin_2_Write(0);
      Pin_3_Write(0);
      Pin_4_Write(0);
      CyDelay(vel);

      Pin_1_Write(0);
      Pin_2_Write(1);
      Pin_3_Write(0);
      Pin_4_Write(0);
      CyDelay(vel);

      Pin_1_Write(0);
      Pin_2_Write(0);
      Pin_3_Write(1);
      Pin_4_Write(0);
      CyDelay(vel);

      Pin_1_Write(0);
      Pin_2_Write(0);
      Pin_3_Write(0);

```

Sentido anti horario, como se puede observar existe una variable llamada vel esta almacenará el valor de la velocidad con el que el motor hará el cambio de pasos de las bobinas, además de que tenemos una variable que almacena el valor de los grados a girar.

```

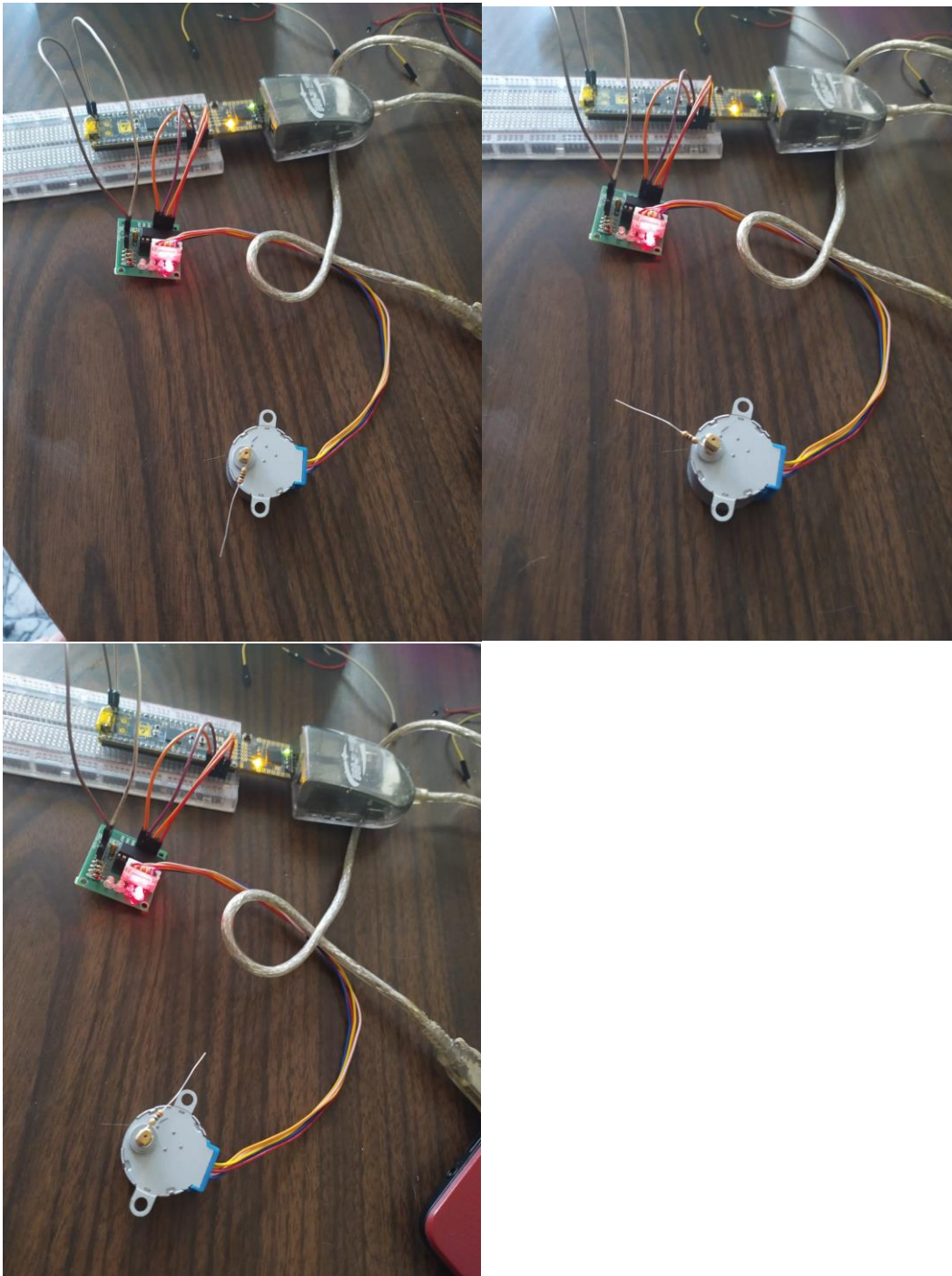
for(;;){
    while (!check_str()) {
        cls();
        print("\n\rGrados a girar: ");
        print(str);
        CyDelay(50);
    }
    if(!isNumber(str)){
        print("\n\rIngrese numero");
    } else {
        sscanf(str, "%d", &grados);
        if(!(grados<=720) || !(grados>=-720)){
            print("\n\rFuera de rango");
        } else {
            if(grados <= 0){
                antihorario(-grados);
                grados = 0;
            } else {
                horario (grados);
            }
        }
    }
}

```

Aquí empieza nuestro código raíz donde ingresara el numero de grados a girar, después entrara en un comparador donde si rebasa 720 o es menor a 720 nuestro código arrojará un enunciando que hay un fuera de rango al usuario, después entrara en otro comparador donde si grados es menor a 0, entrara en el sentido antihorario, si no será al revés.

Para la conexión, utilizamos PUTTY para enlazar nuestra terminal con el programa, utilizando el puerto USB que estaba en uso.

Circuito Armado y funcionando:



Conclusión

La UART es un dispositivo programable en el que pueden establecerse las condiciones que se utilizarán para la transmisión (velocidad, paridad, longitud y bits de parada). En los

primeros PCs, eran circuitos integrados 8250 de National Semiconductor (un chip de 40 patillas DIP -Dual-In-Line-), pero se han ido utilizando otros a lo largo de la evolución del PC.