



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Stepper Motor

PRACTICA

Materia:

Diseño de Sistemas Digitales

Grupo:

2CV17

Profesora:

Jiménez Ruíz René Baltazar

Alumno:

Castro Cruces Jorge Eduardo



Boleta:

2015080213

Fecha:

martes, 15 de junio de 2021


INTRODUCCIÓN

**Baltazar Jiménez**
Administrador · 8 h · 


Hola muchachos,
Hoy no habrá clase en línea. Por favor, diseña lo siguiente con un motor a pasos unipolar (investiga).


- Controla el sentido de giro (ya lo has hecho)
- Controla la velocidad, serán dos disponibles, de 5Hz y de 32Hz.

El usuario decide el sentido de giro y la velocidad mediante dos botones.
Que tengas excelente día.

 5

Visto por 37

 Me gusta

 Comentar

DESARROLLO

- El primer paso fue elaborar la tabla de estados de nuestro motor paso a paso:

PASO COMPLETO SIMPLE (WAVE DRIVE)						
PASO	A	B	C	D		
1	1	0	0	0		
2	0	1	0	0		
3	0	0	1	0		
4	0	0	0	1		
PASO	A	B	C	D	IF X == 0 THEN CW	
1	0	1	0	0		
2	0	0	1	0		
3	0	0	0	1		
4	1	0	0	0		
PASO	A	B	C	D	IF X == 1 THEN CCW	
1	0	0	0	1		
2	1	0	0	0		
3	0	1	0	0		
4	0	0	1	0		

- Después, realizamos los cálculos pertinentes para conocer la velocidad a la que va a trabajar nuestro Motor:

Sabemos que la frecuencia es inversa al periodo:

$$f(\text{Hz}) = \frac{1}{t(\text{s})}$$

También, sabemos que:

$$1\text{segundo} = 1000\text{milisegundos}$$

Aplicando la formula anterior, para calcular el periodo que necesitamos de demora entre cada paso del motor:

$$\therefore t(\text{s}) = \frac{1}{f(\text{Hz})}$$

Evaluamos con los datos que el problema nos pide:

$$t(s) = \frac{1}{5\text{Hz}} = 0.2s$$

$$t(s) = \frac{1}{32\text{Hz}} = 0.032s$$

Ya que nosotros aplicamos el método de paso completo simple, significa que vamos al valor anterior lo debemos dividir entre 4, ya que son los pasos que nuestro motor debe realizar para dar 1 vuelta:

$$\frac{t(s)}{4} = 0.05s$$

$$\frac{t(s)}{4} = 0.008s$$

Aplicando una sencilla regla de 3, calculamos el periodo de demora entre cada paso:

$$\frac{1\text{segundo}}{\frac{t(s)}{4}} = \frac{1000\text{ milisegundos}}{X}$$

Evaluamos nuestros valores calculados:

$$\frac{1\text{segundo}}{0.05\text{segundo}} = \frac{1000\text{ milisegundos}}{X}$$

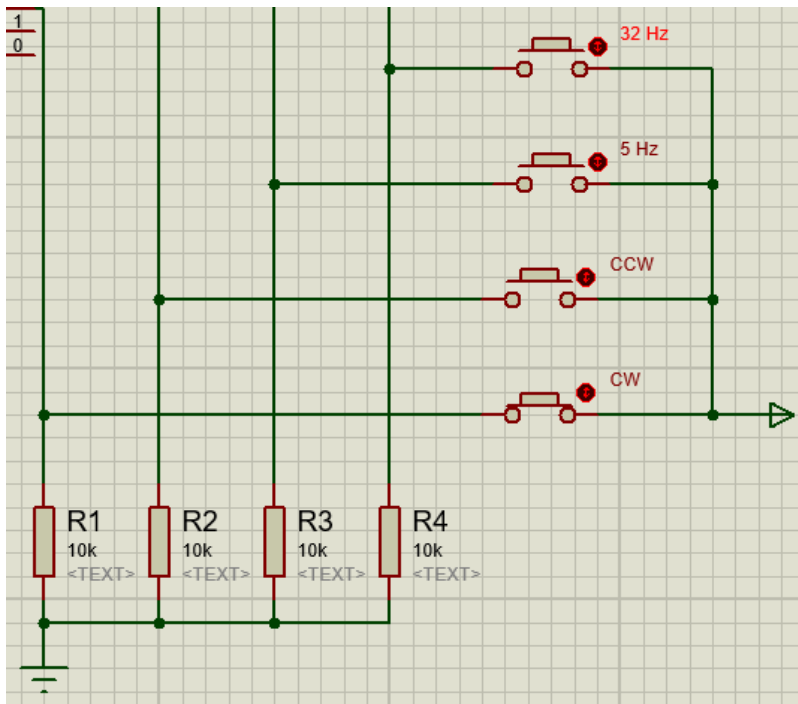
$$\frac{1\text{segundo}}{0.008\text{segundo}} = \frac{1000\text{ milisegundos}}{X}$$

Por último, despejamos a X:

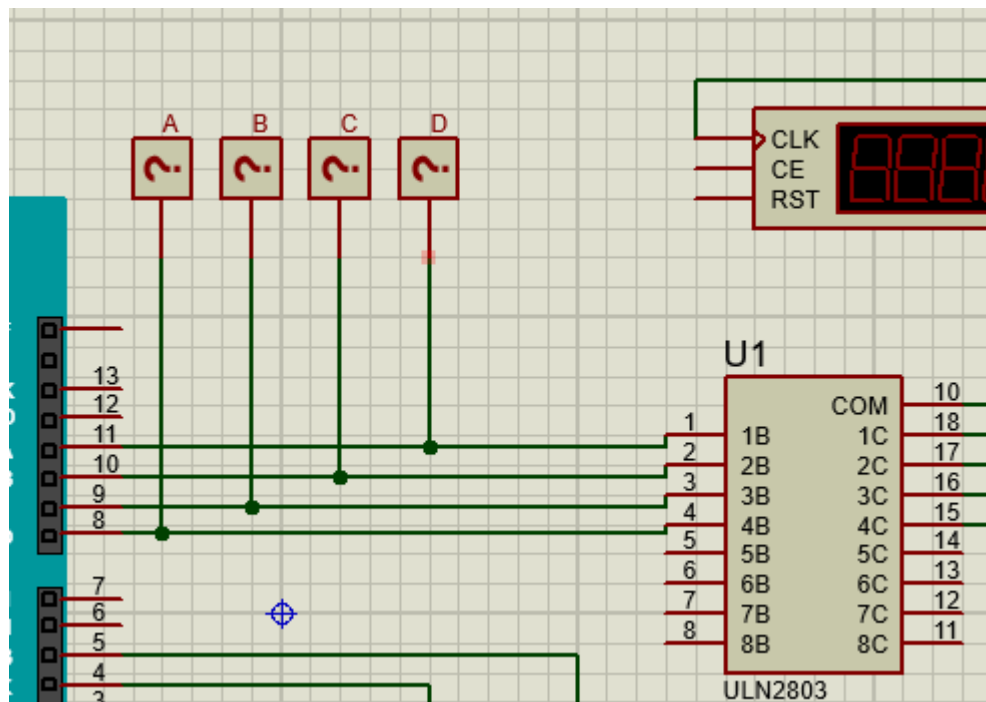
$$\therefore X = \frac{1\text{segundo}}{0.05\text{segundo} * 1000\text{ milisegundos}} = \mathbf{50\text{milisegundos}}$$

$$\therefore X = \frac{1\text{segundo}}{0.008\text{segundo} * 1000\text{ milisegundos}} = \mathbf{8\text{milisegundos}}$$

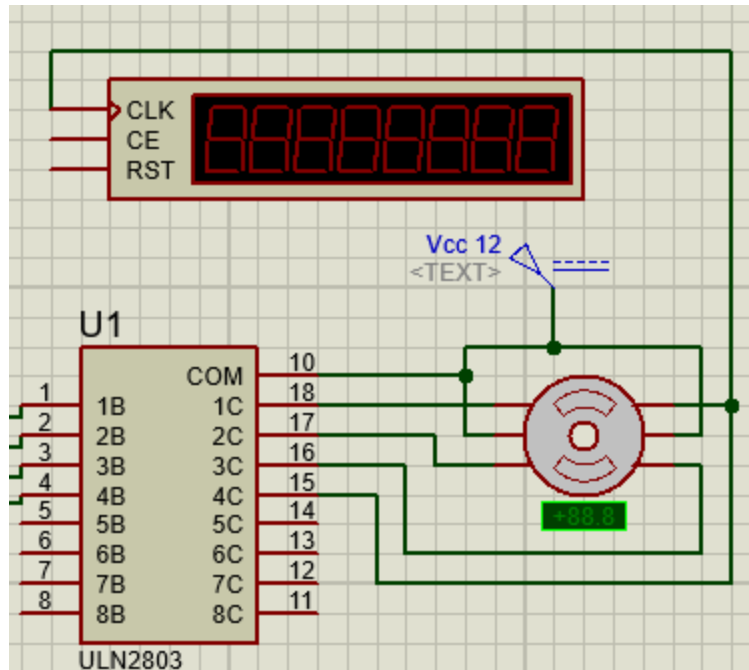
3. Luego, diseñamos el circuito en un simulador (Proteus):
 - a. Conexión de nuestros botones de entrada (Sentido de giro) y conexión de nuestros botones de entrada (Velocidad)



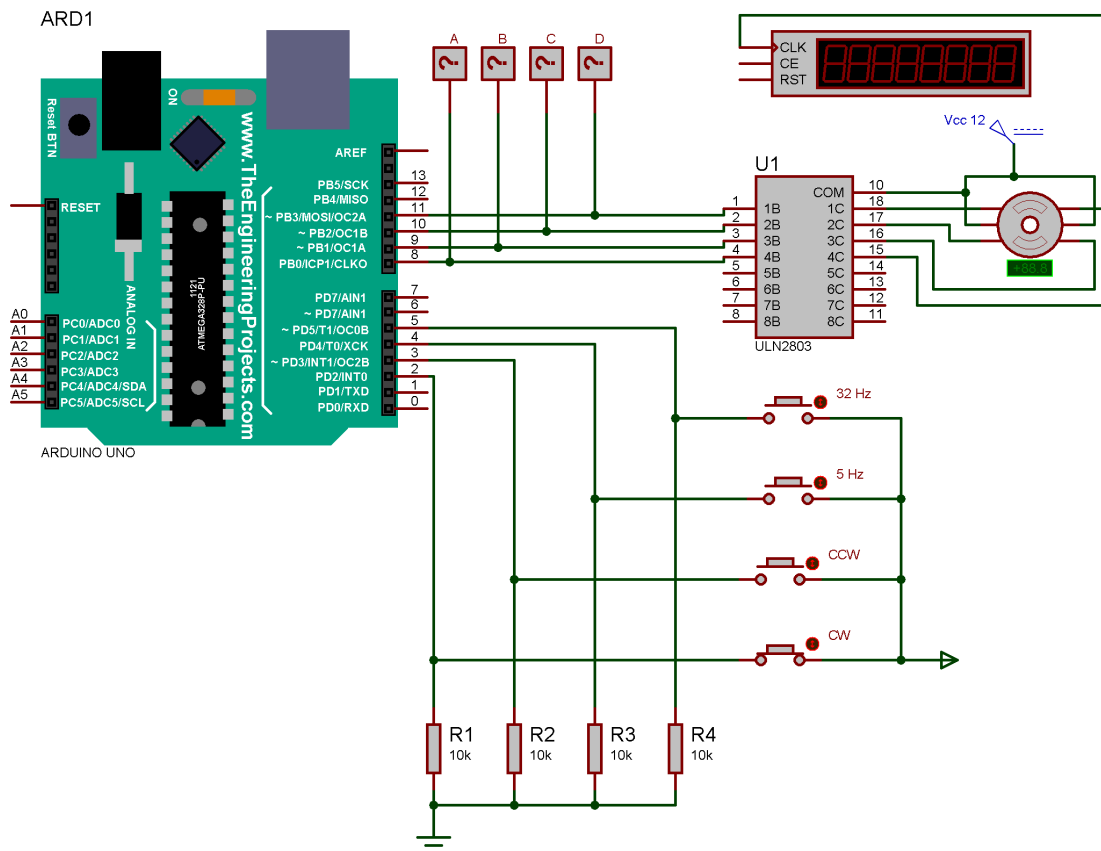
- b. Conexión de nuestro controlador del Motor (ULN2803)



c. Conexión de nuestro motor a pasos unipolar



d. Conexión de nuestra tarjeta de adquisición (ARDUINO UNO)

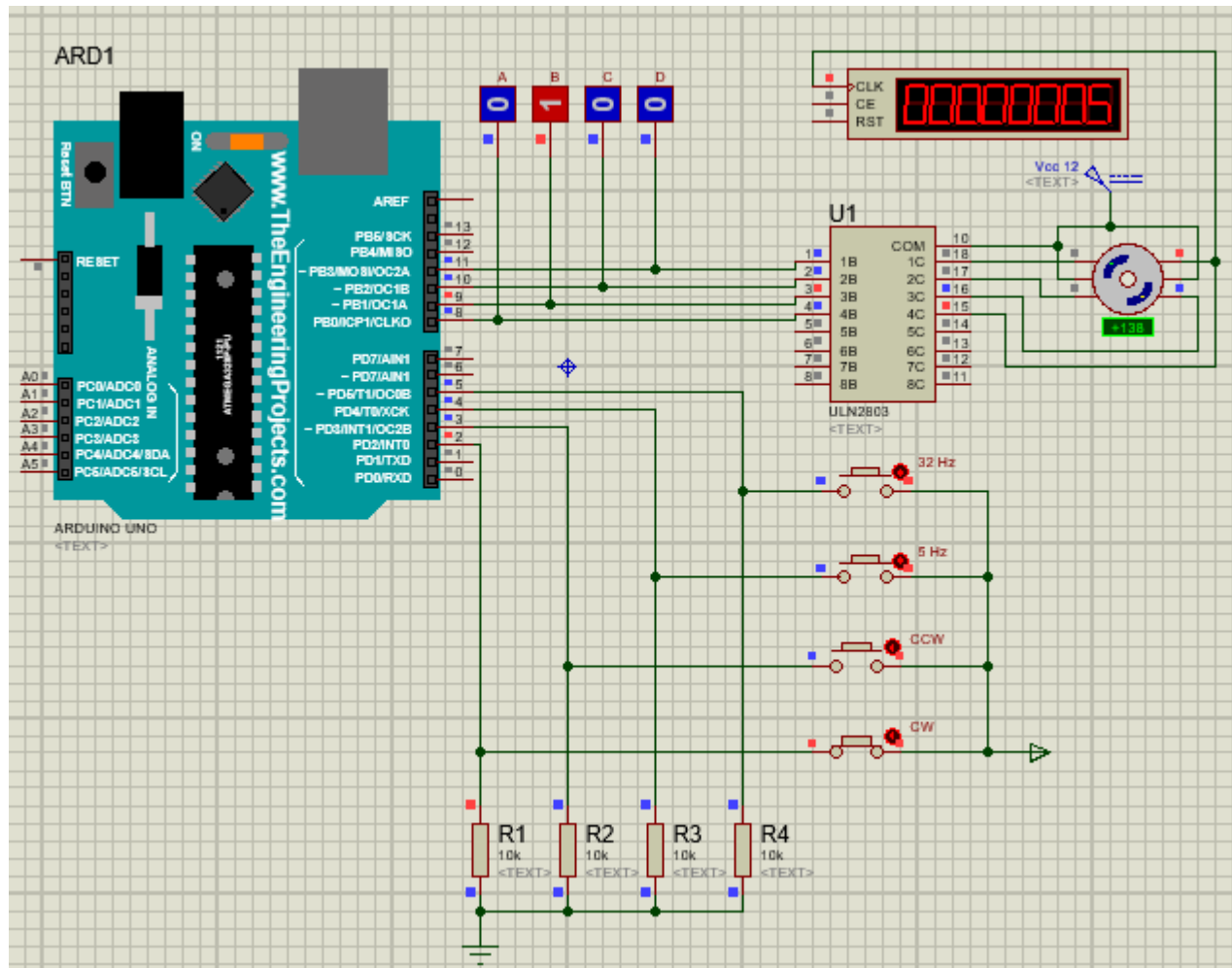


4. Por último, se plasmó el código de Arduino:

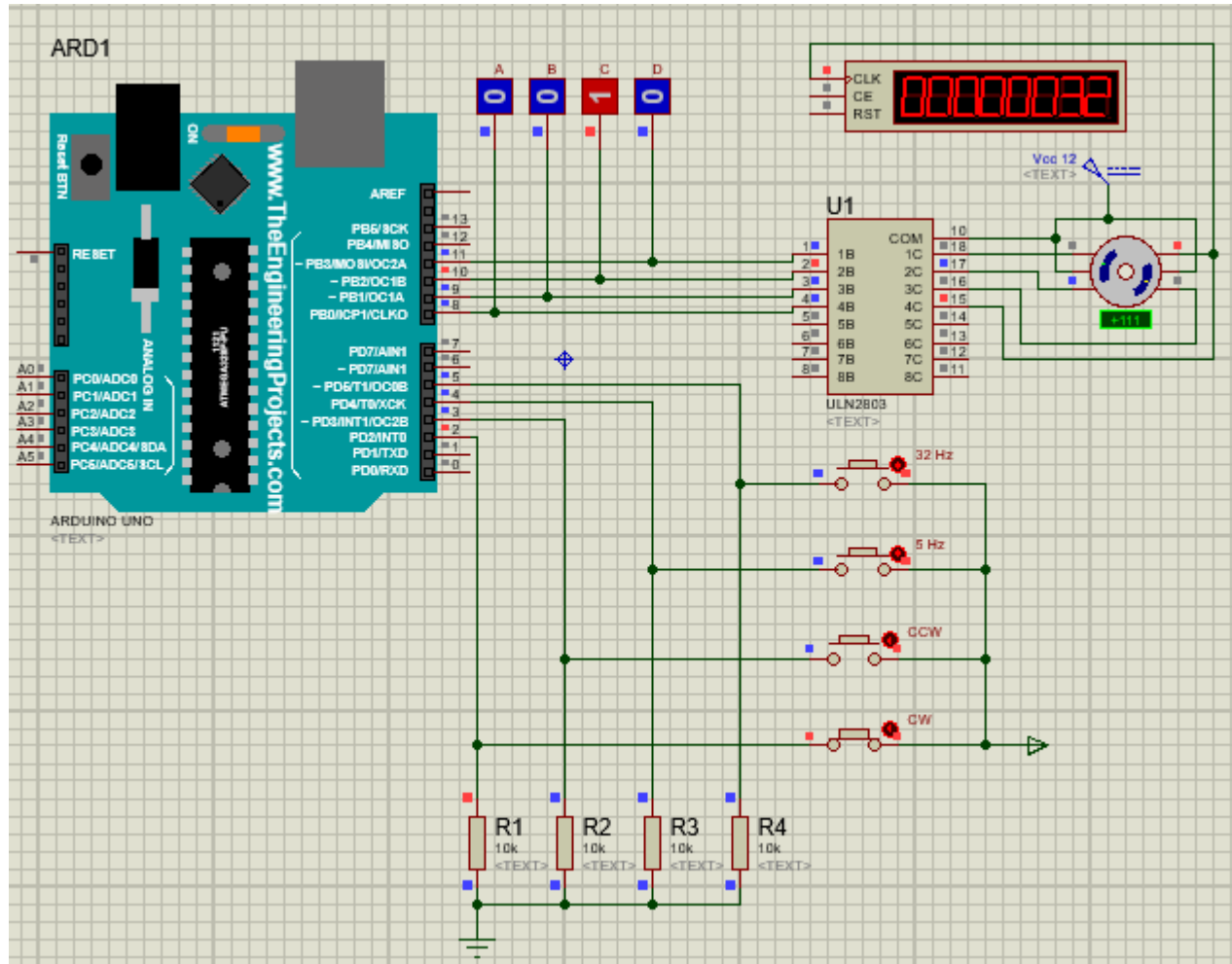
```
1 #define IN1 8
2 #define IN2 9
3 #define IN3 10
4 #define IN4 11
5 #define CW 2
6 #define CCW 3
7 #define Frec5 4
8 #define Frec32 5
9
10 int Demora;
11 int Pasos [4][4] = {
12     {1, 0, 0, 0},
13     {0, 1, 0, 0},
14     {0, 0, 1, 0},
15     {0, 0, 0, 1}};
16
17 void setup() {
18     pinMode(IN1, OUTPUT);
19     pinMode(IN2, OUTPUT);
20     pinMode(IN3, OUTPUT);
21     pinMode(IN4, OUTPUT);
22
23     pinMode(CW, INPUT_PULLUP);
24     pinMode(CCW, INPUT_PULLUP);
25     pinMode(Frec5, INPUT_PULLUP);
26     pinMode(Frec32, INPUT_PULLUP);
27 }
28
29 void loop() {
30     if(digitalRead(Frec5)==HIGH) {
31         Demora = 50;
32     }
33     if(digitalRead(Frec32)==HIGH) {
34         Demora = 8;
35     }
36
37     if(digitalRead(CW)==HIGH) {
38         for (int j = 0; j < 4; j++){
39             digitalWrite(IN1, Pasos[j][0]);
40             digitalWrite(IN2, Pasos[j][1]);
41             digitalWrite(IN3, Pasos[j][2]);
42             digitalWrite(IN4, Pasos[j][3]);
43             delay(Demora);
44         }
45     }
46
47     if(digitalRead(CCW)==HIGH) {
48         for (int j = 3; j >= 0; j--){
49             digitalWrite(IN1, Pasos[j][0]);
50             digitalWrite(IN2, Pasos[j][1]);
51             digitalWrite(IN3, Pasos[j][2]);
52             digitalWrite(IN4, Pasos[j][3]);
53             delay(Demora);
54         }
55     }
56 }
```

CAPTURAS DE PANTALLA

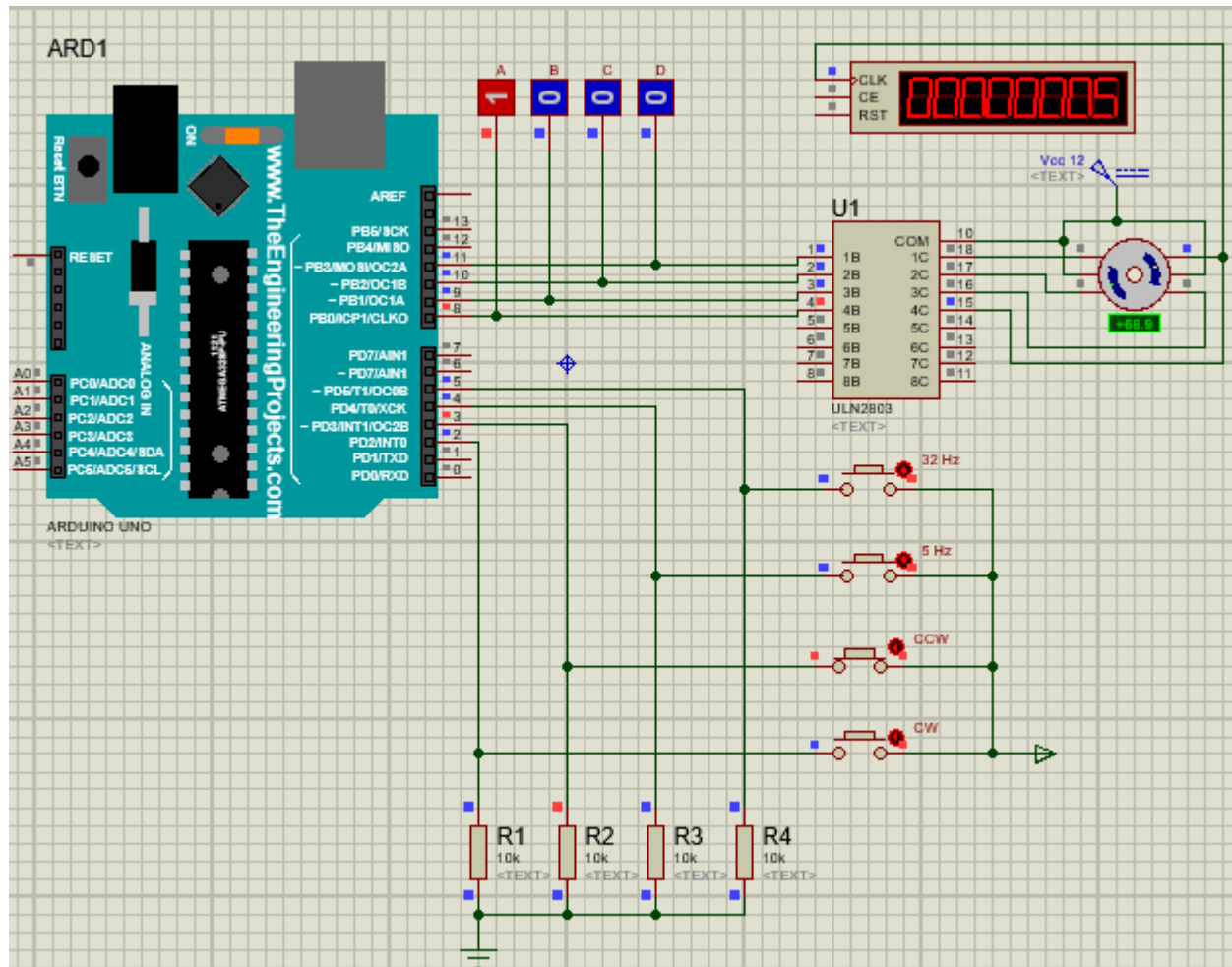
Motor trabajando en sentido de las manecillas del reloj a 5 Hz:



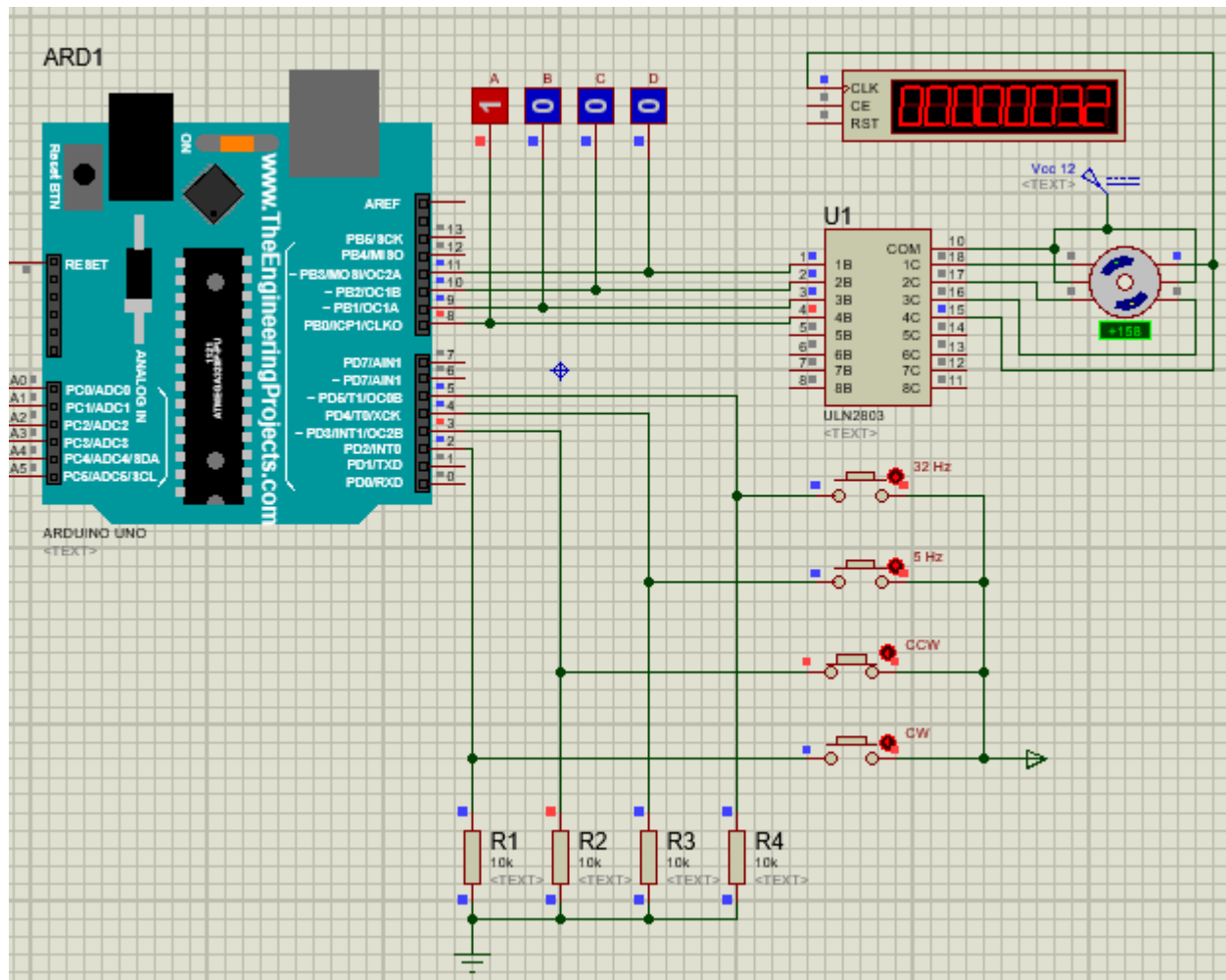
Motor trabajando en sentido de las manecillas del reloj a 32 Hz:



Motor trabajando en sentido opuesto de las manecillas del reloj a 5 Hz:



Motor trabajando en sentido opuesto de las manecillas del reloj a 32 Hz:



CONCLUSIONES

En conclusión, se lograron en su totalidad los objetivos de la práctica:

- Diseño de un motor paso a paso unipolar
- Control del sentido de giro del motor mediante botones
- Control de la velocidad del motor (5 Hz y 32 Hz) mediante botones
- Programación de la tarjeta de adquisición (Arduino Uno)
- Simulación del circuito en Proteus

Cabe hacer las siguientes aclaraciones sobre la simulación:

- Botón (32 Hz) Modifica la velocidad del motor a 32 Hz
- Botón (5 Hz) Modifica la velocidad del motor a 5 Hz
- Botón (CCW) Modifica la dirección de giro del motor en contra de las manecillas del reloj
- Botón (CW) Modifica la dirección de giro del motor en sentido de las manecillas del reloj
- LED A: Se enciende cuando la bobina A está encendida
- LED B: Se enciende cuando la bobina B está encendida
- LED C: Se enciende cuando la bobina C está encendida
- LED D: Se enciende cuando la bobina D está encendida
- El voltaje de alimentación del motor es de 12V
- El medidor de frecuencia nos muestra la velocidad de giro del motor en Hz
- El encapsulado ULN2803 es el encargado de controlar el encendido y apagado de cada una de las bobinas de nuestro motor paso a paso

Con esto concluye la práctica.