



Escuela: Centro de enseñanza técnica industrial – Colomos

Carrera: Desarrollo de software

Materia: Sistemas embebidos 2.

Tema: Act3 Prácl Control de motor a CD.

Equipo: Carlos Daniel Lozano Vázquez 18300249

Ángel Alberto Rivas Álvarez 18100242

Grupo y grado: 7ºA1.

Fecha: 26/08/2021.

Objetivo:

Variar la velocidad de un motor a CD, utilizando el módulo de PWM de un sistema digital, proporcionando el tiempo de aceleración y desaceleración.

Descripción:

Utilizar una tarjeta de desarrollo electrónico para realizar un control de velocidad incremental y decremental para un motor a CD por medio de la modulación de pulso y el uso de una interfaz de potencia. El motor deberá acelerar desde cero hasta la velocidad máxima en un tiempo estipulado y después desacelerar hasta llegar a cero nuevamente.

Utilizar el teclado para pedir el tiempo, la pantalla para mostrar tanto las preguntas necesarias y el porcentaje de aceleración del motor.

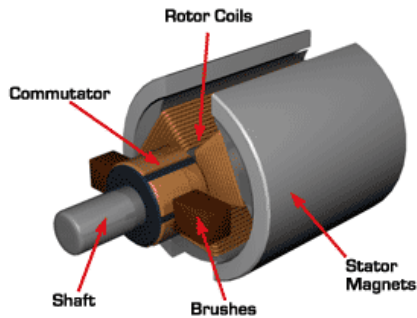
Resumen:

En esta práctica emplearemos uso del motor a corriente directa el cual es una máquina que convierte la energía eléctrica en energía mecánica, provocando un movimiento rotatorio en un eje, gracias a la acción que se genera del campo magnético.

Existen varios tipos de motores de corriente directa por su construcción y por ende el cómo se generan los campos electromagnéticos, entre los cuales podemos mencionar los siguientes:

- Motor serie
- Motor en paralelo
- Motor compuesto
- Motor de imán permanente
- Motor paso a paso

En este caso se usará un motor DC de 5V el cual tiene la siguiente composición:



También se empleará el uso de un transistor NPN para la interfaz de potencia y hacer girar el motor de 5v ya que se requiere una fuente de alimentación externa.

ON CHARACTERISTICS

DC Current Gain (Note 2.) ($I_C = 0.1 \text{ mA}$, $V_{CE} = 1.0 \text{ Vdc}$) ($I_C = 1.0 \text{ mA}$, $V_{CE} = 1.0 \text{ Vdc}$) ($I_C = 10 \text{ mA}$, $V_{CE} = 1.0 \text{ Vdc}$) ($I_C = 50 \text{ mA}$, $V_{CE} = 1.0 \text{ Vdc}$) ($I_C = 100 \text{ mA}$, $V_{CE} = 1.0 \text{ Vdc}$)	2N3903	h_{FE}	20	—	—
	2N3904		40	—	—
	2N3903		35	—	—
	2N3904		70	—	—
	2N3903		50	150	—
	2N3904		100	300	—
Collector-Emitter Saturation Voltage (Note 2.) ($I_C = 10 \text{ mA}$, $I_B = 1.0 \text{ mA}$) ($I_C = 50 \text{ mA}$, $I_B = 5.0 \text{ mA}$)	2N3903	$V_{CE(sat)}$	30	—	Vdc
	2N3904		60	—	
	2N3903		15	—	
Base-Emitter Saturation Voltage (Note 2.) ($I_C = 10 \text{ mA}$, $I_B = 1.0 \text{ mA}$) ($I_C = 50 \text{ mA}$, $I_B = 5.0 \text{ mA}$)	2N3904	$V_{BE(sat)}$	30	—	Vdc
	2N3903		0.65	0.85	
	2N3904		—	0.95	

ON Semiconductor™

<http://onsemi.com>

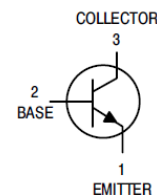
MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector-Emitter Voltage	V_{CEO}	40	Vdc
Collector-Base Voltage	V_{CBO}	60	Vdc
Emitter-Base Voltage	V_{EBO}	6.0	Vdc
Collector Current – Continuous	I_C	200	mA
Total Device Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	625 5.0	mW mW/ $^\circ\text{C}$
Total Device Dissipation @ $T_C = 25^\circ\text{C}$ Derate above 25°C	P_D	1.5 12	Watts mW/ $^\circ\text{C}$
Operating and Storage Junction Temperature Range	T_J, T_{stg}	–55 to +150	$^\circ\text{C}$

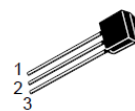
THERMAL CHARACTERISTICS (Note 1.)

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Ambient	$R_{\theta JA}$	200	$^\circ\text{C/W}$
Thermal Resistance, Junction to Case	$R_{\theta JC}$	83.3	$^\circ\text{C/W}$

1. Indicates Data in addition to JEDEC Requirements.

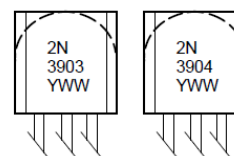


STYLE 1



TO-92
CASE 29
STYLE 1

MARKING DIAGRAMS



Y = Year
WW = Work Week

Este transistor se usa conectando el colector al motor, la base va con el pin PWM del Arduino y entre estas 2 va una R de 320 ohms, por último, el emisor va a GND del Arduino y de la fuente de 5v (Matricula: 2N3904).

Otro dispositivo del cual se hará uso es de un teclado matricial 4x4 de uso rudo, el cual es menos frágil que el de membrana, este teclado se usara para ingresar el intervalo de segundos que queremos en el cual el motor alcance su máxima velocidad y su mínima velocidad.

Este teclado es un arreglo de botones los cuales envían una señal dependiendo del botón que se trate, por ende, en el Arduino se tiene que hacer un mapeo del teclado.

Ya, por último, se usa una LCD de 20x4, es decir 20 columnas por 4 filas, en la cual se mostrará el tiempo en segundos y la velocidad del motor en porcentaje.

Esta pantalla LCD es un dispositivo que permite mostrar diversos caracteres, lo cual es muy útil ya que no tenemos que emplear el uso de display ni otros circuitos integrados.

Diagrama a bloques:

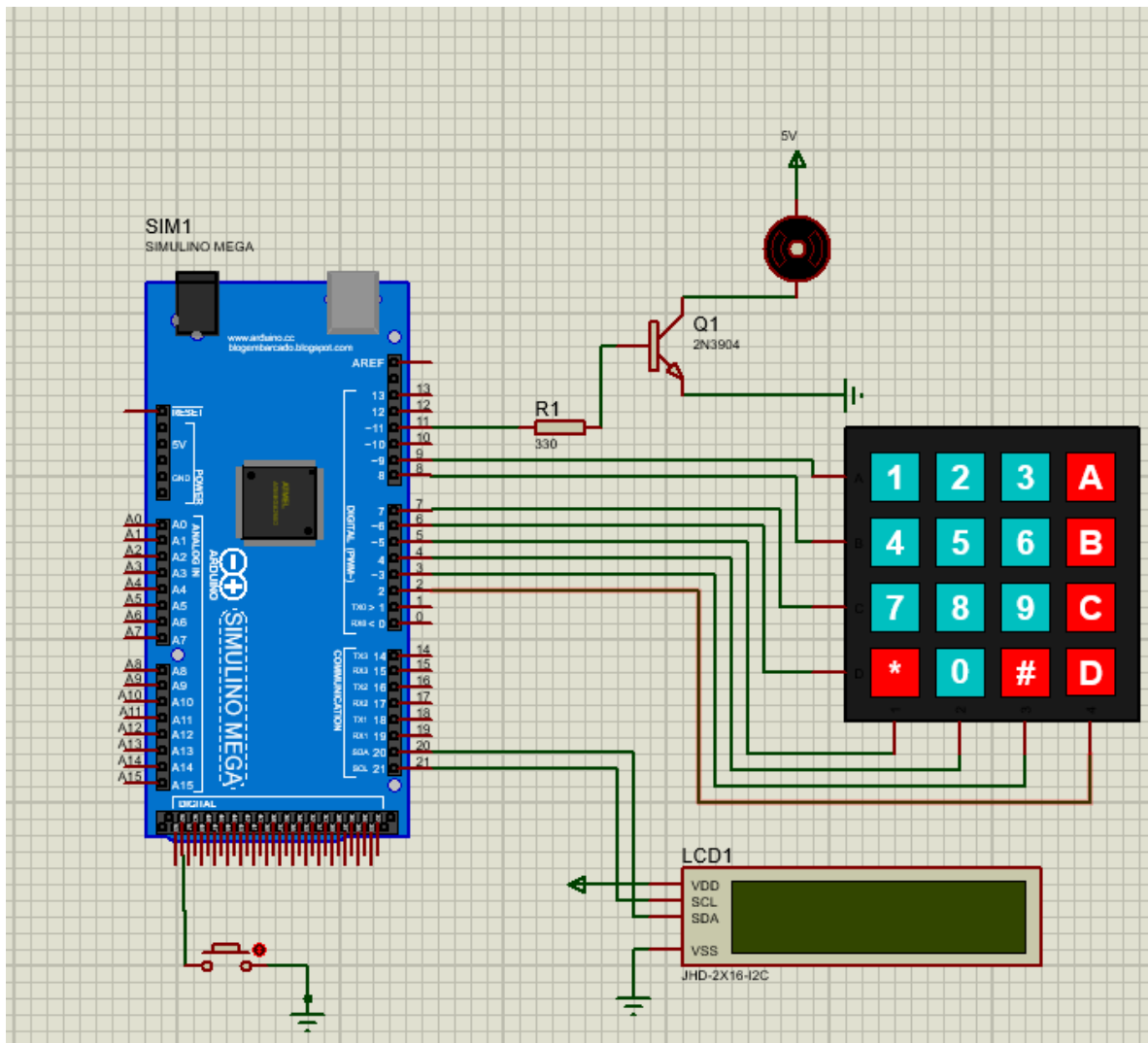
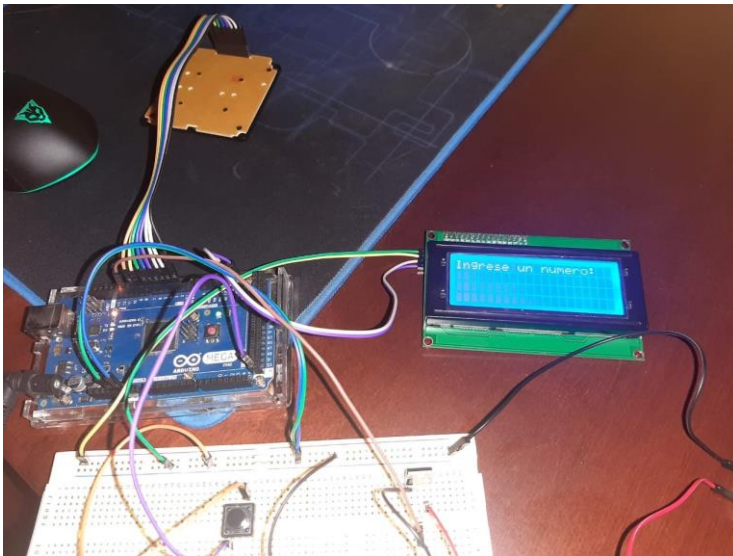
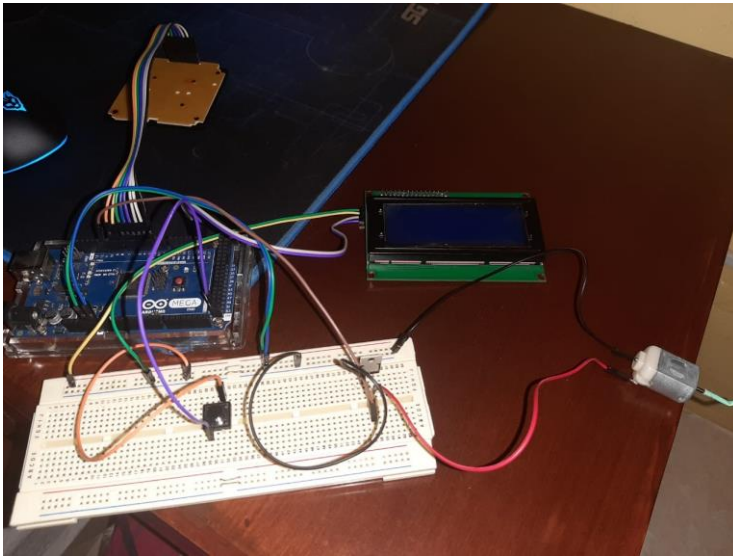


Diagrama del circuito simulado:



Programa:

```
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);

//Variables para asignar pines
short button = 52, Motor = 11;
const byte Fila = 4;
const byte Columna = 4;
char keys_p[Fila][Columna]
{
    {'1', '4', '7', '*'},
```

```

    {'2','5','8','0'},
    {'3','6','9','#'},
    {'A','B','C','D'}
};

byte pin_f[Fila] = {9, 8, 7, 6};
byte pin_c[Columna] = {5, 4, 3, 2};
//mapeo del teclado
Keypad TechHex = Keypad(makeKeymap (keys_p), pin_f, pin_c, Fila, Columna);
double enter();

//funcion para imprimir texto
void show (const char* str, int CoorX = 0, int CoorY = 0) {
    //coordenadas para inicializar el cursor
    if (CoorX >= 0 && CoorY >= 0)
        lcd.setCursor(CoorX, CoorY);
    //imprime la string
    lcd.print(str);
}

//funcion para imprimir numeros
void show (double num, int CoorX = 0, int CoorY = 0) {
    if (CoorX >= 0 && CoorY >= 0)
        lcd.setCursor(CoorX, CoorY);
    lcd.print(num);
}

//funcion para obtener rango de 2 puntos diferentes (ecuacion de la recta)
inline double conversion( double X, double int_min, double int_max, double Out_min, double Out_max){
    return (X-int_min) * (Out_max-Out_min)/(int_max - int_min) + Out_min;
}

void setup() {
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0,0);
    pinMode(Motor, OUTPUT);
    pinMode(button, INPUT_PULLUP);//boton para secuencia
}

void loop() {
    double num = 0, temp = 0, incremento = 0, porcentaje = 0;
    int tiempo = 0;
    lcd.print("Ingrese un numero: ");
    num = enter();

```

```

system("cls");
show("Tiempo: ", 1, 0);
show(tiempo, 9, 0);
show("Porcentaje: ", 1, 1);
show(porcentaje, 13, 1);
lcd.print("%");
//obtener razón de crecimiento.
incremento = 256/num;
//se le asigna la razón de crecimiento
temp = incremento;
for(; temp < 256; temp += incremento){
    delay(1000);
    lcd.clear();
    tiempo ++;
    //obtiene el punto
    porcentaje = conversion( temp, 0, 255, 0, 100);
    show("Tiempo: ", 1, 0);
    show(tiempo, 9, 0);
    show("Porcentaje: ", 1, 1);
    show(porcentaje, 13, 1);
    lcd.print("%");
    analogWrite(Motor, temp);
}
//en caso de que el valor supere 255
if(temp + incremento >= 256 && tiempo < num){
    delay(1000);
    lcd.clear();
    //se analiza si el tiempo transcurrido es menor al ingresado, si
    es así se incrementa en 1
    tiempo ++;
    porcentaje = 100;
    temp = 255;
    show("Tiempo: ", 1, 0);
    show(tiempo, 9, 0);
    show("Porcentaje: ", 1, 1);
    show(porcentaje, 13, 1);
    analogWrite(Motor, 255);
    lcd.print("%");
}
tiempo = 0;
temp-=incremento;
for(; temp >=0; temp -= incremento){
    delay(1000);
    lcd.clear();
    tiempo ++;

```



```

    porcentaje = conversion( temp, 0, 255, 0, 100);
    show("Tiempo: ", 1, 0);
    show(tiempo, 9, 0);
    show("Porcentaje: ", 1, 1);
    show(porcentaje, 13, 1);
    lcd.print("%");
    analogWrite(Motor, temp);
}
if(temp - incremento <= 0 && tiempo < num){
    delay(1000);
    lcd.clear();
    tiempo ++;
    porcentaje = 100;
    show("Tiempo: ", 1, 0);
    show(tiempo, 9, 0);
    show("Porcentaje: ", 1, 1);
    show(porcentaje, 13, 1);
    analogWrite(Motor, 0);
    lcd.print("%");
}
while(digitalRead(button) == HIGH){} //ciclo para apreciar resultado
delay(200);
lcd.clear();
}

double enter()
{
    double value = 0;
    char c = '\0';
    int X3 = 10, temp=0;//coordenada X y valor temporal
    c = TecHex.getKey();

    while( c =='\0' || c < 58 || c > 47)
    {
        delay(1);
        c = TecHex.getKey();
        if( c !='\0' && c < 58 && c > 47){

            lcd.setCursor(X3,2);
            lcd.print(c);
            temp = String(c).toInt(); //se le resta a c el valor ascii de 48 --
            > '0'
            value = (value * 10.00) + temp; //se le ingresa el valor ingresado y
            se multiplica x10 para que funcione

```

```

//ejemplo (0*10)+1---
> (1*10)+2, esto quiere decir que se ingreso primero el 1 y luego el 2 y te
da 12
    X3++; //se le suma uno a la coordenada X
    }
    if(c == '#')
        break;
    }
    lcd.clear();
    return value;
}

```

Explicación:

Se inicializa los pines del teclado, botón, del PWM/Motor, al igual que se hace un mapeo del teclado y también se inicializa el objeto de la LCD, por último, se inicializan las variables globales.

Ya al poner los pines correspondientes como entrada y salida, en el loop se imprime unos valores en la lcd, después de eso se le solicita al usuario ingresar un número, en este caso se validan que no se ingrese letras y solamente cuando se presione el carácter '#', se romperá el ciclo y se tendrá el valor ingresado en segundos para alcanzar la velocidad del motor.

Teniendo este valor, lo que sigue será sacar una taza de crecimiento dividiendo 256/el valor ingresado, ya teniendo dicha taza se le asigna a un valor temporal para poder usarlo en un ciclo iterativo, dicho ciclo servirá para aumentar la velocidad por medio de esa taza, con la función conversión la cual usa la ecuación de la recta como base para su funcionamiento se obtendrá el porcentaje de velocidad, a su vez se imprimirán los valores de tiempo y porcentaje en la LCD.

Otra cosa es que se debe de usar analogWrite(), para poder enviar la señal de trabajo de un pin PWM.

Después de esto se usa un if para aumentar al 100% la velocidad del motor ya que el ciclo ignora ese caso (en algunas ocasiones).

Se hace lo mismo que el ciclo pasado, la única diferencia es que es regresiva.

Al terminar el 2do ciclo se debe de presionar el botón para poder volver a seguir con el ciclo, dicho botón se inicializo como INPUT_PULLUP ya que así me ahorro una Resistencia y siempre estará en alto su estado lógico, por ende, el push cambiará su estado a 0 o bajo para romper el ciclo while.

Observaciones:

Se me dificultó la parte de conectar el motor DC con su interfaz de potencia porque creí que al combinar GND del Arduino y de la fuente iba a hacer un corto, pero no fue así.

Conclusiones:

Es muy interesante ver como el PWM y acelerar un motor de 5v, ya que se tiene que usar la función `analogWrite()` para que funcione de manera correcta, y como es un pin digital solo tiene un rango de 8 bits, es decir de 0 a 255 en decimal.