**Logotipo, nombre de la empresa

Descripción generada automáticamente**

**Escuela:** Centro de enseñanza técnica industrial – Colomos  
**Carrera:** Desarrollo de software  
**Materia:** Sistemas embebidos 2.  
**Tema:** Act3 Prác1 Control de motor a CD.  **Equipo:** Carlos Daniel Lozano Vázquez18300249

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:**

**Diagrama a bloques:**

Diagrama, Esquemático

Descripción generada automáticamente

**Diagrama del circuito simulado:**

Imagen que contiene computadora, tabla

Descripción generada automáticamente

Imagen que contiene circuito

Descripción generada automáticamente

**Programa:**

#include <LiquidCrystal\_I2C.h>

#include <Keypad.h>

LiquidCrystal\_I2C lcd(0x27, 20, 4);

*//Variables para asitgnar 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 TecHex = 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 razon 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 aniliza 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 dificulto 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.