**Logotipo, nombre de la empresa

Descripción generada automáticamente**

**Escuela:** Centro de enseñanza técnica industrial – Colomos  
**Carrera:** Desarrollo de software  
**Materia:** Sistemas embebidos 2.  
**Tema:** Actividad 8. Práctica 4: Relevadores.  **Equipo:** Carlos Daniel Lozano Vázquez18300249

Ángel Alberto Rivas Álvarez 18100242 **Grupo y grado:** 7ºA1. **Fecha:** 07/09/2021.

**Objetivo:**

Usar relevadores para el cambio de giro de un motor a CD y elegir una velocidad de giro.

**Descripción:**

Usar dos relevadores, para poder hacer el cambio de giro de un motor a CD; deberán preguntar el sentido de giro del motor y la cantidad de energía (porcentaje de velocidad) que se le entregara a dicho 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: Diagrama

Descripción generada automáticamente

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.

Tabla

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente con confianza media

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.

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.

Se usarán 2 relés para poder usar un puente H con ellos, el relé tiene una bobina de activación para poder ser usado como interruptor y tiene 3 puertos, el comúnmente cerrado, abierto y el común, el comúnmente abierto cuando se activa el relé se hace comúnmente cerrado teniendo contacto con el común.

**Diagrama eléctrico:**

Diagrama, Esquemático

Descripción generada automáticamente

**Diagrama del circuito:** Imagen que contiene computadora, monitor, luz, oscuro

Descripción generada automáticamente

**Programa:**

#*include* <Keypad.h>

#*include* <LiquidCrystal\_I2C.h>

#*define* *GIRO\_IZQ* 50

#*define* *GIRO\_DER* 51

#*define* *PWM\_* 11

LiquidCrystal\_I2C *lcd*(0x27, 20, 4);

typedef unsigned short us;

char keys\_p[4][4]

{

   {'1','4','7','\*'},

   {'2','5','8','0'},

   {'3','6','9','#'},

   {'A','B','C','D'}

};

*byte* pin\_f[4] = {9, 8, 7, 6};

*byte* pin\_c[4] = {5, 4, 3, 2};

//*mapeo del teclado*

Keypad TecHex = *Keypad*(*makeKeymap* (keys\_p), pin\_f, pin\_c, 4, 4);

bool State = LOW;

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);

}

void *setup*(){

    lcd.*init*();

    lcd.*backlight*();

//*se inicializan los pines*

*pinMode*(GIRO\_DER, OUTPUT);

*pinMode*(GIRO\_IZQ, OUTPUT);

*pinMode*(PWM\_, OUTPUT);

*digitalWrite*(GIRO\_DER, HIGH);

*digitalWrite*(GIRO\_IZQ, HIGH);

}

void *loop*(){

    us Opc = 0, porcentaje = 0;

*while*(Opc < 1 || Opc > 3)

    {

*show*("1.- Giro IZQ ", 1, 0);

*show*("2.- Giro DER ", 1, 1);

*show*("3.- Stop", 1, 2);

*show*(" Opc: ", 1, 3);

//*opcion del menu*

        Opc = *enter*();

    }

*while*((porcentaje < 1 || porcentaje > 100) && Opc != 3)

    {

        lcd.*clear*();

*show*("Veloicdad: ", 1, 0);

//*velocidad ingresada en porcentaje*

        porcentaje = *enter*();

    }

*if*(Opc == 3)

    {

*digitalWrite*(GIRO\_IZQ, HIGH);

*delay*(10);

*digitalWrite*(GIRO\_DER, HIGH);

*analogWrite*(PWM\_, 0);

    }

*else*

    {

//*el estado para girar a izquierda o derecha se le asigna un 1 o 0 logico*

        State = (Opc == 1)? LOW : HIGH;

*digitalWrite*(GIRO\_IZQ, State);

*delay*(10);

//*se le pone el estado negado para que pueda fluir la corriente*

*digitalWrite*(GIRO\_DER, !State);

//*se manda la señal PWM al pin*

*analogWrite*(PWM\_,  *map*(porcentaje, 0, 100, 0, 255));

    }

*delay*(10);

}

us *enter*(){

  int value = 0, temp = 0;

  char c = '\0';

  c = TecHex.*getKey*();

*while*( c =='\0' || c < 58 ||c > 47)

  {

*delay*(1);

     c = TecHex.*getKey*();

*if*( c !='\0' && c < 58 && c > 47){

        lcd.*print*(c);

        temp = *String*(c).*toInt*();//*se le resta a c el valor ascii de 48 --> '0'*

        value = (value \* 10) + temp;

     }

*if*(c == '#')

*break*;

  }

  lcd.*clear*();

*return* value;

}

**Explicación:**

Se inicializa los pines del teclado, el objeto de la clase del motor a pasos, 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.

Al poner los pins 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.

Se debe ingresar una opción, si es 1 state es LOW y en Giro\_Izq será Low por ende se activa el 2do relé y el Giro\_der es HIGH por lo que el 1er relé esta desactivado, en 3 se ponen ambos pines en HIGH

**Observaciones:**

Me costo un poco el poner la interfaz de potencia para que giraran los relés y lo solucione poniéndolo en el comúnmente cerrado con ambas gnd del Arduino y de la fuente

**Conclusiones:**

Es importante ver como trabaja la polarización de un motor para poder trabajar de manera efectiva con un motor a DC, además de poder acelerarlo.