**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 6. Práctica 2: Motor a pasos.  **Equipo:** Carlos Daniel Lozano Vázquez18300249

Ángel Alberto Rivas Álvarez 18100242 **Grupo y grado:** 7ºA1. **Fecha:** 05/09/2021.

**Objetivo:**

Mandar una secuencia de movimiento a un servomotor, para poder posicionar en cualquier ángulo, de entre 0 a 180 grados.

**Descripción:**

Realizar una secuencia de movimientos en un servomotor. El usuario del programa debe indicar desde una hasta nueve posiciones del motor, indicando los ángulos en que se debe colocar el servo motor, el cual debe tener algún elemento para indicar la posición (un pequeño brazo o palanca). Una vez hecha la indicación de ángulos, la secuencia debe ser visible en el servomotor.

Recordar que cada secuencia es un ángulo al que se desplazará el servomotor, y dicho desplazamiento será entre 0 y 180 grados, y que deberá quedarse entre tres y cinco segundos en dicha secuencia, para poder verlo.

**Resumen:**

Un servomotor es un dispositivo eléctrico autónomo que gira partes de una máquina con alta eficiencia, es decir, tiene un consumo de energía reducido y con gran precisión.

El eje de salida de este motor se puede mover a un ángulo, posición y velocidad particulares que un motor normal no tiene.

Utiliza un motor normal y lo acopla con un sensor para retroalimentación posicional.

El controlador es la parte más importante del servomotor diseñado y utilizado específicamente para este propósito.

Un servomotor es, en otras palabras, un motor especial al que se le ha añadido un sistema de control, un potenciómetro y un conjunto de engranajes. Lo cual nos permite controlar su posicionamiento y velocidad en 360 grados.

Partes de un servomotor

● Motor

● Círculo de control

Aplicaciones de servomotor

● Uniones de mecanismos electrónicos que requieren ángulos de movimiento precisos.

● Sistemas de enfoque óptico automático en cámaras.

● Sistemas de posicionamiento de antenas.

● Sistemas de dirección de aviones, helicópteros y barcos

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 eléctrico:**

**Diagrama, Esquemático

Descripción generada automáticamente**

**Diagrama del circuito:**

Imagen que contiene tabla, computadora, cuarto

Descripción generada automáticamente

**Programa:**

#include <Servo.h>

#include <Keypad.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 20, 4);

Servo servo1;

*char* keys\_p[4][4]

{

   {'1','4','7','\*'},

   {'2','5','8','0'},

   {'3','6','9','#'},

   {'A','B','C','D'}

};

*unsigned* *short* \*grados;

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);

*int* 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*);

}

*void* setup() {

  lcd.init();

  lcd.backlight();

  servo1.attach(11, 500, 2400);

}

*void* loop() {

*short* opc1 = 0, opc2 = 0, num = 0, cant = 0, \*grados;

*int* tiempo = 0;

    while(cant <= 0 || cant > 9){

        show("Posiciones: ", 0, 0);

        cant = (*short*)enter();

    }

    grados = new *short*[cant];

    for(*short* i = 0; i < cant; i++){

        show("Ingrese grados ", 0, 0);

        show(" Num: ", 0, 2);

        grados[i] = (*short*) enter();

    }

    for(*short* i = 0; i < cant; i++){

        lcd.clear();

        servo1.write(grados[i]);

        show("Grados: ", 0, 1);

        lcd.print(servo1.read());

        delay(5000);

    }

    lcd.clear();

    delete []grados;

}

*int* enter(){

*int* 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) + temp;

        X3++;*//se le suma uno a la coordenada X*

     }

     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.

Ahora se ingresa la cantidad de movimientos/direcciones que se quiera ingresar en un intervalo del 1-9, después de ingresar ese dato se hace un arreglo con dicha longitud y se llena con valores en grados de 0º a 180º.

Después se entra a un ciclo donde se imprimirá la posición actual durante 5seg y se moverá el servo.

**Observaciones:**

Es importante saber las herramientas que tiene Arduino, para facilitarnos las tareas de funcionamiento, aunque también hay que saber cómo funcionan los componentes.

**Conclusiones:**

Los servomotores son muy importantes para saber la posición de algo ,en sistemas de dirección ya que son muy precisos y podemos saber en que posición se encuentran, además de que estos pueden toleras bastantes kilos, sólo que aumenta su precio.