



Universidad Autónoma Chapingo



Departamento de Ingeniería Mecánica
Agrícola

Ingeniería Mecatrónica Agrícola

Tarea 3:

Control de motor a pasos

Asignatura:

Sistemas digitales embebidos

Profesor:

Luis Arturo Soriano Avendaño

Alumnos:

Cocotle Lara Jym Emmanuel

López Hernández Alfonso

Reyes Paredes Antonio

Ruiz Diaz Carlos Eduardo

Contenido

Introducción.....	2
Proceso de idealización.....	5
Comunicación del equipo	5
Lluvia de ideas	7
Asignación de roles.....	7
Paso a paso la toma de decisiones.....	7
Propuesta de solución	8

Introducción

El motor paso a paso es un tipo de motor muy utilizado en aplicaciones robótica, también permite transformar un impulso eléctrico en un movimiento angular.

Este tipo de motor es muy habitual en los dispositivos donde se desea realizar un control de velocidad o de posición en un bucle abierto, típico de los sistemas de posicionamiento. El uso más común y conocido por la mayoría es el de las impresoras conectadas a un ordenador.

En este proyecto con la ayuda de un microcontrolador PIC16f877A, se simulará el funcionamiento del motor paso a paso, controlando el sentido de giro con un potenciómetro.

Motor paso a paso

Un motor paso a paso es un dispositivo electromecánico que convierte una serie de pulsos eléctricos en desplazamiento angulares, lo que significa que es capaz de girar una cantidad de grados dependiendo de sus entradas de control.

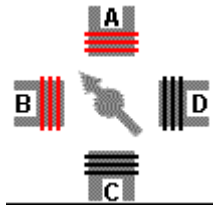
Principios de funcionamiento

El principio de funcionamiento esta basado en un estator construido por varios bobinados en un material ferromagnético y un rotor que puede girar libremente en el estator.

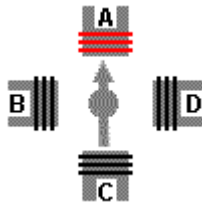
Estos diferentes bobinados son alimentados uno a continuación del otro y causan un determinado desplazamiento angular que se denomina “paso angular” y es la principal característica del motor.

Secuencias de manejo

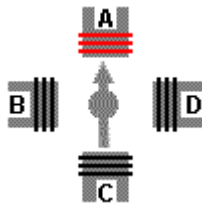
- Secuencia normal: con esta secuencia el motor siempre avanza un paso por vez debido a que siempre existen 2 bobinas activas, con esta secuencia se obtiene un alto torque de paso y retención.



- Secuencia wave drive (paso completo): en esta secuencia se activa solo una bobina por vez, lo que ocasiona que el eje del motor gire hacia la bobina activa.

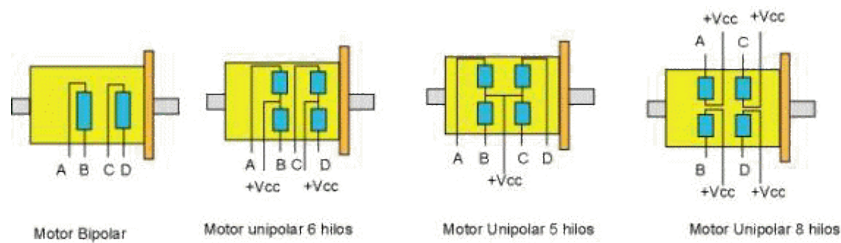


- Secuencia medio paso: en esta secuencia se activa las bobinas de tal manera que se combinan las secuencias, el resultado que se obtiene es un paso mas corto, primero se activan dos bobinas y posteriormente solo 1 y así sucesivamente.



Existen dos tipos de motor paso a paso:

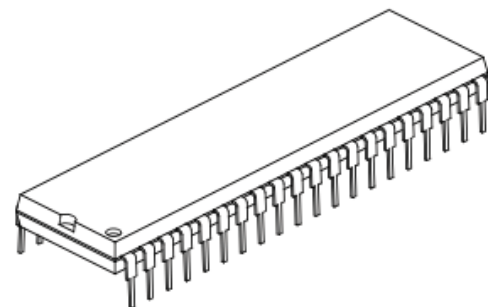
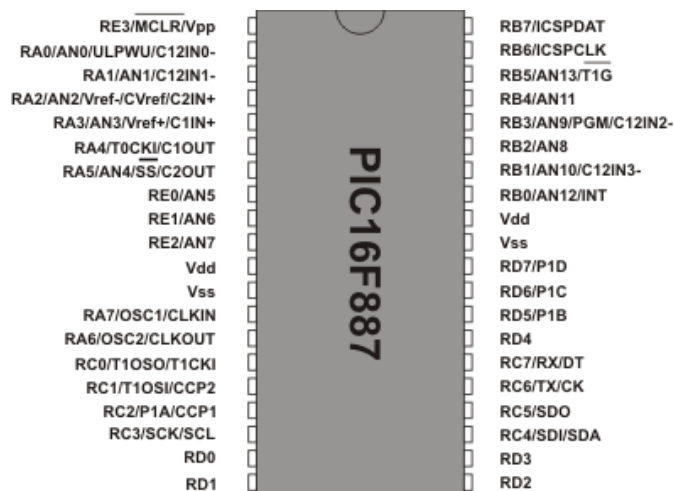
- Motores unipolares: este tipo de motor tiene dos bobinas en cada uno de los estatores y cada par de bobinas tienen un punto común, es decir, tienen 5 o 6 terminales.
- Motores bipolares: este tipo de motor tiene dos bobinas y no poseen puntos comunes, es decir tiene cuatro terminales. Para controlar este tipo de motor paso a paso bipoar es necesaria usar 8 transistores o circuitos integrados especiales.}



Microcontrolador

Un microcontrolador es como un ordenador en pequeño: dispone de una memoria donde se guardan los programas, una memoria para almacenar datos, dispone de puertos de entrada y salida, etc. A menudo se incluyen puertos seriales, conversores analógicos/digital, generadores de pulsos PWM para el control de motores, bus I2C.

Microcontrolador 16f877A.



Proceso de idealización

La tarea que se pide consiste en realizar el control de un motor a pasos por medio de un programa realizado en lenguaje C, posteriormente debe estar simulado en Proteus. Los parámetros para llevar a cabo el diseño son:

- Motor a paso unipolar
- Utilizar un microcontrolador
- Usar las entradas y salidas digitales
- Emplear el convertidor analógico digital

Las funciones que debe realizar son:

- Mediante un potenciómetro variar la velocidad de giro
- Por medio de un botón cambiar el sentido de giro

Comunicación del equipo

Para establecer una comunicación entre los integrantes del equipo, se creó un chat entre los mismos en Microsoft Teams, posteriormente se realizó una reunión por medio de esta plataforma, la comunicación fue satisfactoria, de esta forma se realizó la lluvia de ideas que se presenta a continuación, de igual manera se estableció el método de trabajo.



Ilustración 1. Equipo de trabajo

Lluvia de ideas

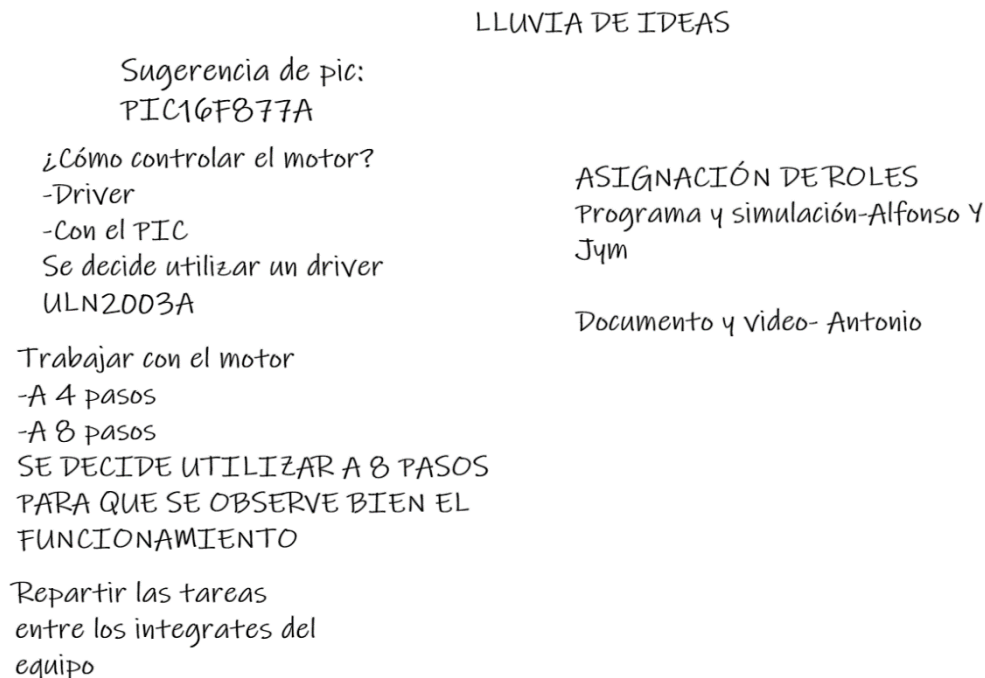


Ilustración 2. Lluvia de ideas

Asignación de roles

Para asignar los roles que debe realizar cada integrante del equipo se consideraron las habilidades de cada uno y las herramientas con las que cuenta. De esta manera se lograron distribuir las actividades de la mejor forma.

Paso a paso la toma de decisiones

1. Se inicio con una reunión por medio de Teams, donde se llevó a cabo la lluvia de ideas, y se acordó la forma de trabajo.
2. Después se realizó una breve investigación acerca de lo que se va a desarrollar, los componentes a utilizar y la manera de llevarlo a cabo.

Código en C

```
// Universidad Autónoma Chapingo

//      DIMA

// Ingeniería Mecatrónica Agrícola

//      Equipo 1

//   Integrantes del equipo:

// Alfoso López Hernández

// Antonio Reyes Paredes

// Carlos Eduardo Ruíz Díaz

// Jym Emmanuel Cocotle Lara


#include <Tarea3.h>

// Habilitamos el puerto b

#define standard_io(b)

// Habilitamos el puerto c

#define standard_io(c)


// Establecemos una variable de tipo entero y lo igualamos a 0

int contador = 0;

// Establecemos una variable de tipo entero de 16 bits

int16 lectura_ADC;

// Establecemos una variable de tipo entero de 16 bits

int16 millis;


// ClockWise

// Establecemos el arreglo para el giro en sentido horario

byte const CW[8] = {0b1000, 0b1100, 0b0100, 0b0110,

                   0b0010, 0b0011, 0b0001, 0b1001};

// Counter ClockWise

// Establecemos el arreglo para el giro en sentido antihorario

byte const CCW[8] = {0b1001, 0b0001, 0b0011, 0b0010,

                    0b0110, 0b0100, 0b1100, 0b1000};


// Función que establece el sentido y velocidad del motor

void motor(int contador){

    // Si el botón esta presionado, el motor gira en sentido horario

    if(input_state(PIN_C0)){//CW

        // Paso a la derecha

        output_b(CW[contador]);

    }

}
```

```

// De lo contrario gira en sentido antihorario
else{//CCW

    // Paso a la izquierda
    output_b(CCW[contador]);

}

// Control de velocidad
// Al valor del regulador se le resta 1 hasta llegar a 0
// con un retardo de 1 milisegundo
while(millis > 0){
    delay_ms(1);
    millis--;
}

// Varía dependiendo el equipo por la sobrecarga de la simulación
delay_ms(400);
}

void main(){
    // Establecemos la entrada 0 como analógica
    setup_adc_ports(AN0);

    // ADC trabaja con el reloj interno
    setup_adc(ADC_CLOCK_INTERNAL);

    // Establecemos el puerto b como salidas
    set_tris_b(0x00);

    // Establecemos el puerto c0 como entrada
    set_tris_c(0x01);

    while(TRUE)
    {
        // Control de posición del motor

        // Declaramos un bucle el cual sirve para leer los arreglos
        // que declaramos anteriormente
        for(contador = 0; contador <= 7; contador++){
            set_adc_channel(0); // Declaración del pin A0 como ADC

            delay_us(20); // Retardo de estabilización

            lectura_ADC = read_ADC(); // Lectura del ADC

            // Establecemos la velocidad

            millis = lectura_ADC; // Relación del ADC con la velocidad

            motor(contador); // Llamado de la función
        }
    }
}

```

