

"Año del Diálogo y Reconciliación Nacional"

ASCENSOR INTELIGENTE

CARRERA:

Técnicas en Ingeniería Electrónica

ALUMNO:

- Magh

2018

I) OBJETIVOS

El objetivo de este proyecto es diseñar un control eléctrico y electrónico que controle e integre todos los elementos propios de un ascensor.

Creamos un ascensor que, por mediante una aplicación de voz de nuestro teléfono pueda subir y bajar al destino deseado.

Para cumplir estos objetivos, el proyecto abarca multitud de ámbitos de ingeniería como son:

- Electrónica: Microcontroladores, sistemas lógicos, transmisión de datos.
- Electricidad: Máquinas eléctricas, electrotécnica, líneas de conducción, baja tensión.
- Automática industrial: Autómatas programables, Grafcets de control.

II) MATERIALES

- **Rodajes:** Cojinete que consta de dos cilindros concéntricos entre los que va colocado un juego de rodillos o bolas que pueden girar libremente.



- **Engranajes:** Se denomina engranaje al mecanismo utilizado para transmitir potencia mecánica de un componente a otro. Los engranajes están formados por dos ruedas dentadas, de las cuales la mayor se denomina corona y la menor piñón. Un engranaje sirve para transmitir movimiento circular mediante el contacto de ruedas dentadas.



- **Angulo de fierro una pulgada:** Producto no plano de sección transversal formada por dos alas de igual longitud en ángulo recto y que se obtienen por Laminación de palanquillas de Acero Estructural, previamente calentadas hasta una temperatura del orden de los 1250°C.



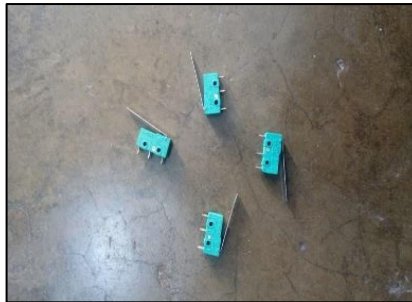
- **Hilo de pescar:** Este material es usado para dar a nuestros sedales la propiedad de ser o más bien, parecer invisibles a los ojos de los peces debajo del agua.



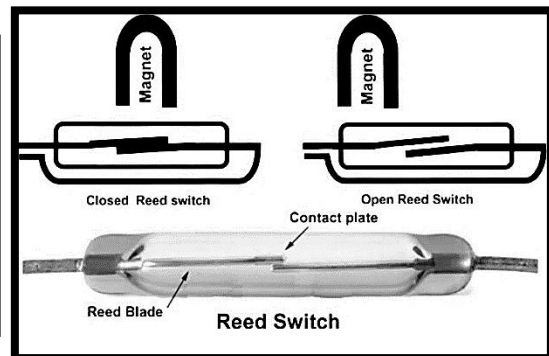
- **Madera:** es un material ortótropo, con distinta elasticidad según la dirección de deformación, encontrado como principal contenido del tronco de un árbol.



- **Fines de carrera:** Son dispositivos electrónicos, neumáticos o mecánicos situados al final del recorrido o de un elemento móvil, como por ejemplo una cinta transportadora, con el objetivo de enviar señales que puedan modificar el estado de un circuito.



- **Red switch:** Un interruptor de lengüeta o reed switch o relé reed es un interruptor eléctrico activado por un campo magnético. Cuando los contactos están normalmente abiertos se cierran en la presencia de un campo magnético; cuando están normalmente cerrados se abren en presencia de un campo magnético.



- **Lector de CD:** Para leer discos compactos (CD) se necesita un lector de CD-ROM, que es un dispositivo dependiente del ordenador y controlado por éste. Estos lectores admiten tanto discos compactos musicales como CD-ROM.



- **Motorreductor:** Los reductores de velocidad con sistemas formados por engranajes que hacen que los motores eléctricos funcionen a distintas velocidades. Los reductores o motorreductores son necesarios para toda clase de máquinas y aparatos de uso industrial que precisan reducir de forma segura su velocidad.



III) MARCO TEÓRICO

Un ascensor o elevador es un sistema de transporte vertical diseñado para movilizar personas o bienes entre diferentes niveles. Puede ser utilizado ya sea para ascender o descender en un edificio o una construcción subterránea. Se conforma con partes mecánicas, eléctricas y electrónicas que funcionan conjuntamente para lograr un medio seguro de movilidad. Si se considerara un medio de transporte, sería el segundo más utilizado después del automóvil.

Los ascensores empezaron como simples cuerdas o cadenas a modo de montacargas. Un ascensor es básicamente una plataforma que es tirada o empujada por medios mecánicos.

El año 1823, en Londres, Burton y Horner construyen el llamado cuarto ascendente que tenía capacidad para elevar hasta 20 personas a una altura de 37 metros.

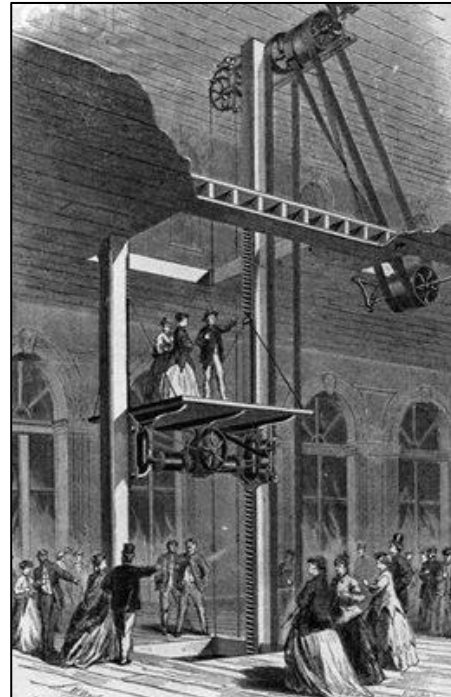
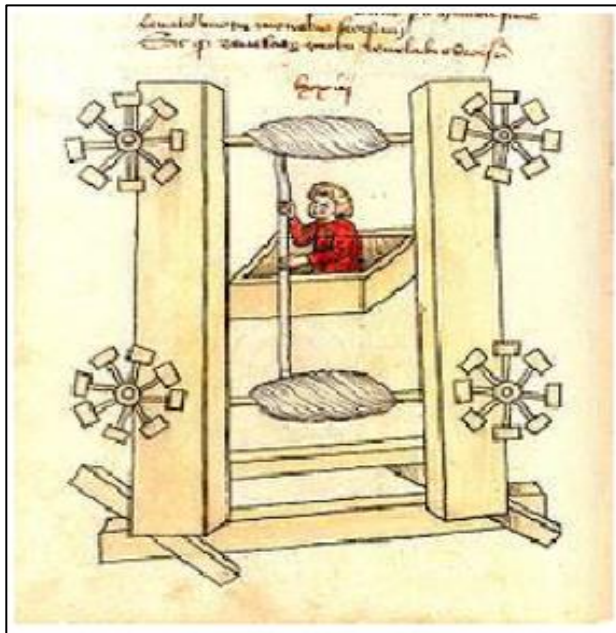
En 1853, Elisha Graves Otis inventa los primeros frenos de emergencia para ascensores, lo cual le lleva a construir elevadores seguros, que venían con un dispositivo de seguridad en caso que el cable de sujeción se rompiera.

El diseño de seguridad de Otis, con algunas diferencias, es similar a uno de los tipos de seguridad que se emplean actualmente. Este dispositivo consistía en una serie de rodillos preparados de tal forma que cuando el ascensor tomaba una velocidad demasiado rápida se bloqueaba el descenso mediante un control automático.



Elisha Graves Otis

Fundador de la empresa OTIS se considera el padre del ascensor moderno y es el precursor de la mayoría de los sistemas de seguridad que se utilizan actualmente



IV) PROCEDIMIENTOS

- Nos dividimos para comprar los materiales, preferimos fierro por ser más rígido y conveniente para el ascensor.
- Soldamos los fierros de un metro, y lo fijamos a la base de madera para hacer empezar con el armado del eje.
- El eje estará formado por:
 - o 2 engranajes. Uno es para que la faja haga girar el eje, el segundo es para poder subir por medio del giro al ascensor.
 - o 2 rodajes. Servirán para sujetar el eje con la parte superior del ascensor. Por medio de ello el eje girará en perfectas condiciones.
- Al fijar los rodajes y los engranajes en el eje. Comenzamos a poner el hilo de pescar trenzado para que aguante el peso del ascensor.
- Soldamos un par de fierros a la parte inferior izquierda de los soportes del ascensor para poder poner el motorreductor de 24v, a una altura determinada para poder tensar la faja que mediante ello hará movimiento al eje que podrá elevar y bajar el ascensor.
- Mediante ello colocamos en la parte inferior atornillándolo con autorroscantes, hubo inconvenientes al ponerlo, porque al elevarlo chocaba con una cierta parte de la estructura que tuvimos que lijarse.
- La parte de la carcasa lo que tapa los soportes, los sujetamos con ángulos para mayor agarre y diseño en el proyecto.





V) PROGRAMACIÓN PARA EL ESP32:

```
/*Programa para compilar al ESP32 Dev Module

Código escrito y modificado por: MAGH

Conexiones para Display LCD:
PIN 21 = SDA
PIN 22 = SCL

*/

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

void resetear_LCD();
int piso_destino(); int piso_actual();
void display_LCD(int actual, int destino);
void puerta();

const int LCD_reset = 2;
const int motor_baja = 12; const int motor_sube = 13;
const int puerta_abre = 14; const int puerta_cierra = 27;
const int fc_puerta_abierta = 18; int fc_abierto = 0; int abrir = 0;
const int fc_puerta_cerrada = 19; int fc_cerrado = 0; int cerrar = 0;
const int piso_1 = 34; int p1 = 0;
const int piso_2 = 35; int p2 = 0;
const int piso_3 = 32; int p3 = 0;
const int piso_4 = 33; int p4 = 0;
int cnx_establecida = 0;
int rst_LCD; int sLCD = 0;
int pisoactual = 0; int pisodestino = 0; int piso = 0; int subiendo = 0; int bajando = 0;
uint8_t downarrow[8] = {0x0A, 0x04, 0x04, 0x04, 0x04, 0x15, 0x0E, 0x04};
uint8_t uparrow[8] = {0x04, 0x0E, 0x15, 0x04, 0x04, 0x04, 0x04, 0x0A};
byte dato;
int temporizador = 0;
LiquidCrystal_I2C lcd(0x27, 20, 4);
BluetoothSerial ESP_BT;

void setup()
{
    // Serial.begin(115200); //Inicia el monitor serial
    ESP_BT.begin("ESP32_MAGH"); //Nombre para el dispositivo Bluetooth y lo inicia
    lcd.begin(); //Inicia el LCD
    lcd.backlight();
    lcd.createChar(0, downarrow); lcd.createChar(1, uparrow);
    lcd.home();
    //Serial.println("Dispositivo iniciado, listo para emparejar.");
    pinMode(motor_sube, OUTPUT);
    pinMode(motor_baja, OUTPUT);
    pinMode(puerta_abre, OUTPUT);
    pinMode(puerta_cierra, OUTPUT);
    pinMode(LCD_reset, INPUT);
    pinMode(piso_1, INPUT);
    pinMode(piso_2, INPUT);
    pinMode(piso_3, INPUT);
```

```
pinMode(piso_4, INPUT);
pinMode(fc_puerta_abierta, INPUT);
pinMode(fc_puerta_cerrada, INPUT);
digitalWrite(motor_sube, HIGH);
digitalWrite(motor_baja, HIGH);
digitalWrite(puerta_abre, HIGH);
digitalWrite(puerta_cierra, HIGH);
}

void loop()
{
    //Resetea el LCD
    resetear_LCD();
    //Detecta si recibio alguna señal por Bluetooth y si es así, hace su lógica.
    temporizador = millis();
    if(millis() >= temporizador+5000) //Cada 5 segundo lo tomará como una desconexión
        //automática por inactividad y mostrará de nuevo los mensajes.
    {
        temporizador = millis();
        cnx_establecida = 0;
    }
    while (!ESP_BT.available() && cnx_establecida == 0)
    {
        lcd.setCursor(1, 1);
        lcd.print("Esperando conexion");
        lcd.setCursor(1, 2);
        lcd.print("Bluetooth...");
        delay(200);
        lcd.clear();
    }
    if (ESP_BT.available())
    {
        if (cnx_establecida == 0)
        {
            lcd.setCursor(3, 1);
            lcd.print("Conexion");
            lcd.setCursor(10, 2);
            lcd.print("establecida...");
            delay(500);
            lcd.clear();
            lcd.clear();
            lcd.setCursor(3, 1);
            lcd.print("Dispositivo");
            lcd.setCursor(10, 2);
            lcd.print("conectado");
            delay(500);
            lcd.clear();
            cnx_establecida = 1;
        }
        piso_destino(); piso_actual();
        piso = pisoactual - pisodestino;
        if (piso < 0)
        {
            piso = -piso;
            puerta(); subiendo = 1;
            while (piso < pisodestino)
```

```
{
    digitalWrite(motor_sube, LOW);
    piso_actual(); piso = pisodestino - pisoactual;
    display_LCD(pisoactual, pisodestino);
}
digitalWrite(motor_sube, HIGH);
subiendo = 0; puerta();
}
else if (piso > 0)
{
    puerta(); bajando = 1;
    while (piso > pisodestino)
    {
        digitalWrite(motor_baja, LOW);
        piso_actual(); piso = pisoactual - pisodestino;
        display_LCD(pisoactual, pisodestino);
    }
    digitalWrite(motor_baja, HIGH);
    bajando = 0; puerta();
}
}
display_LCD(pisoactual, pisodestino);
//delay(100);
}

void resetear_LCD()
{
    //Resetea el LCD
    rst_LCD = digitalRead(LCD_reset);
    if (rst_LCD == 0 && sLCD == 1)
    {
        lcd.clear();
        sLCD = 0;
    }
    if (rst_LCD == 1 && sLCD == 0)
    {
        lcd.begin(); lcd.backlight();
        lcd.setCursor(1, 1); lcd.print("SENATI-EST-uC-Deza");
        lcd.setCursor(0, 2); lcd.print("Ascensor Inteligente");
        sLCD = 1;
        delay(1000);
        lcd.clear();
    }
}

int piso_destino()
{
    dato = ESP_BT.read();
    switch (dato)
    {
        case 1: pisodestino = 1;
            break;
        case 2: pisodestino = 2;
            break;
        case 3: pisodestino = 3;
            break;
        case 4: pisodestino = 4;
            break;
    }
    return pisodestino;
}
```

```
int piso_actual()
{
    p1 = digitalRead(piso_1);
    p2 = digitalRead(piso_2);
    p3 = digitalRead(piso_3);
    p4 = digitalRead(piso_4);
    if (p1 == 1)
    {
        pisoactual = 1;
    }
    if (p2 == 1)
    {
        pisoactual = 2;
    }
    if (p3 == 1)
    {
        pisoactual = 3;
    }
    if (p4 == 1)
    {
        pisoactual = 4;
    }
    return pisoactual;
}

void display_LCD(int actual, int destino)
{
    lcd.clear(); lcd.setCursor(0, 0);
    lcd.print("Ascensor Inteligente");
    if (subiendo == 1)
    {
        lcd.setCursor(4, 1);
        lcd.write(0);
        lcd.print(" Subiendo ");
        lcd.write(0);
    }
    if (bajando == 1);
    {
        lcd.setCursor(4, 1);
        lcd.write(1);
        lcd.print(" Bajando ");
        lcd.write(1);
    }
    if (actual != destino)
    {
        lcd.setCursor(7, 3);
        lcd.print("Piso destino ");
        lcd.print(actual);
        lcd.setCursor(7, 3);
        lcd.print("Piso ");
        lcd.print(destino);
    }
    else
    {
        lcd.print("Piso ");
        lcd.print(destino);
    }
}

void puerta()
{
    fc_abierto = digitalRead(fc_puerta_abierta);
    fc_cerrado = digitalRead(fc_puerta_cerrada);
    if (fc_cerrado == 1 && (subiendo == 0 || bajando == 0))
    {
```



```

    lcd.clear(); lcd.setCursor(5, 1);
    while (fc_abierto == 0)
    {
        digitalWrite(puerta_abre, LOW);
        lcd.print("Abriendo puerta");
    }
    digitalWrite(puerta_abre, HIGH);
    delay(10); lcd.print("Puerta abierta");
}
else if (fc_abierto == 1 && (subiendo == 0 || bajando == 0))
{
    lcd.clear(); lcd.setCursor(5, 1);
    while (fc_cerrado == 0)
    {
        digitalWrite(puerta_cierra, LOW);
        lcd.print("Cerrando puerta");
    }
    digitalWrite(puerta_cierra, HIGH);
    delay(10); lcd.print("Puerta cerrada");
}
}

```

VI) PROGRAMACION PARA ANDROID:

