UNIVERSIDAD NACIONAL DE AVELLANEDA

Arquitectura de Sistemas de Elaboración de Datos II

Trabajo Final

"Ascensor Montacargas"

Alumnos:
Napoleone Luciano, Reádigos Pablo, Serra Tomás

Profesor:
D'angiolo Federico

Presentación:
13 de Diciembre de 2017

Índice

Introducción

Idea Original

En el trascurso de la cursada de Arquitectura de Sistemas de Elaboración de Datos II, dentro de la carrera de Ingeniería en informática de la Universidad de Avellaneda (UNDAV) se propuso como forma de evaluar en segunda instancia, realizar un proyecto en donde incluya algunos de los temas vistos y estudiados en clase y/o en la bibliografía. Al ofrecernos este desafío, sabíamos que la condición era juntarse en grupos de no más de tres o cuatro personas y debíamos elegir un tema que incluya intereses en común, ya que es muy difícil trabajar en algo en lo cual no estaríamos conformes. Conformamos el grupo y entre actualizar los proyectos de materias anteriores entre otras ideas que no nos convencían, surgió el pensamiento de realizar un tipo de **"Ascensor"**, La facilidad de realizar este proyecto radica en que uno de los estudiantes del grupo trabaja en una empresa de ascensores, por lo cual podemos adquirir información privilegiada del funcionamiento de los mismos y seguir normas y estándares para simular y acercarnos un poco más a un esquema real.

Después de analizar las variables que podía tener este vehículo vertical según código de edificación de la Ciudad Autónoma de Buenos Aires y Ordenanza Municipal N° 49308, notamos que por problemas de diseño y arquitectura no podíamos tener las seguridades que requería el ascensor, por lo que tuvimos que adaptarlo a un Ascensor Montacargas, lo cual este equipo fabricado no podría transportar personas, sino solamente objetos o cosas.

Objetivo del Proyecto

En cuanto a los objetivos planteados para este proyecto, podemos identificar dos. Uno general y uno específico.

Con respecto al objetivo general, queda determinado por Introducir a los alumnos y a los lectores de este informe a la generación de un producto orientado a un "Sistema Embebido", teniendo como marco teórico, la materia Arquitectura de Sistemas de Elaboración de Datos II de la UNDAV.

El objetivo específico de este proyecto es en base a la idea del Ascensor Montacargas, construir una plataforma que se adapte y sea funcional a un "hueco de Ascensor", construir una cabina, una "sala de máquinas", además de construir y diseñar el circuito y la lógica del funcionamiento del ascensor.

El ascensor deberá cumplir con los siguientes requisitos.

- Subir y bajar controladamente a través de una botonera.
- Controlar y saber en qué piso se encuentra el montacargas.
- Observar a través de LED'S o display, la posición del ascensor.
- Controlar la velocidad del motor.
- Indicar con dos leds si la maniobra es de subir o es de bajar.
- Realizar varias llamadas en simultáneo.

Desarrollo de Trabajo

Presentación de los elementos utilizados.

Programador PIC: Esta plaqueta fue construida por uno de los integrantes del grupo, por lo que resulto ser la clave para que elijamos PIC como micro controlador para el proyecto, resulto de gran ayuda porque lo utilizamos fuera del ámbito universitario, por lo que no dependíamos estar en la facultad para poder programar, por lo que pudimos hacer pruebas en nuestras casas. La plaqueta fue construida en el trascurso de las clases de la materia.

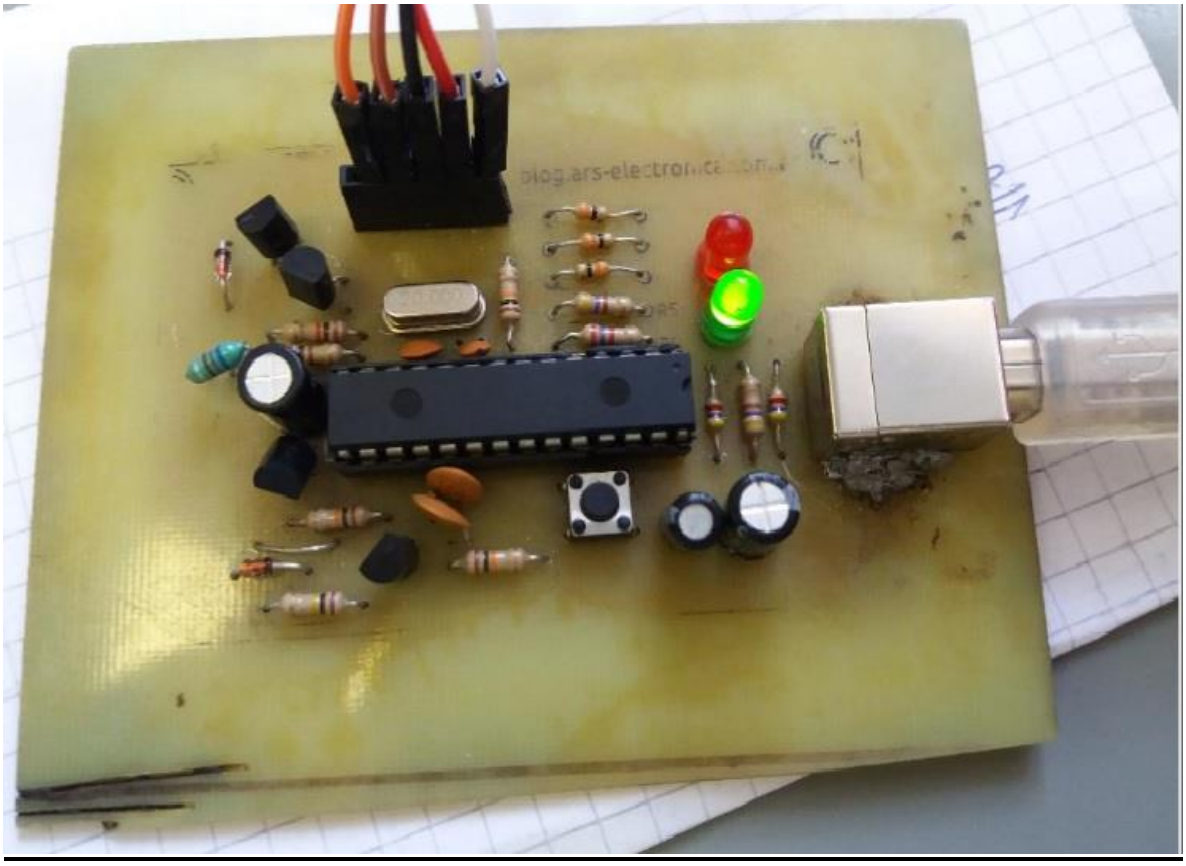


Imagen 1 - Programador PicKit Construido en UNDAV

Por abajo se puede observar el circuito impreso. (Imagen 2)

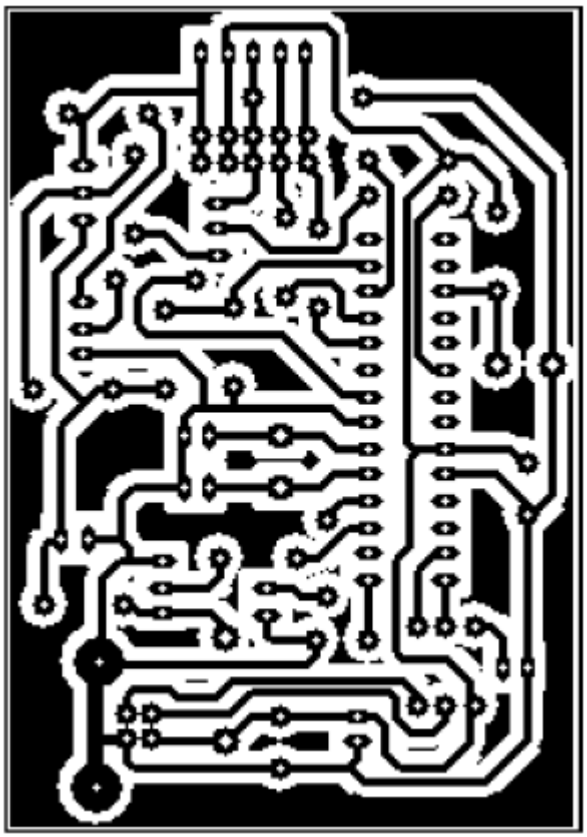


Imagen 2 - Circuito Impreso Programador PicKit

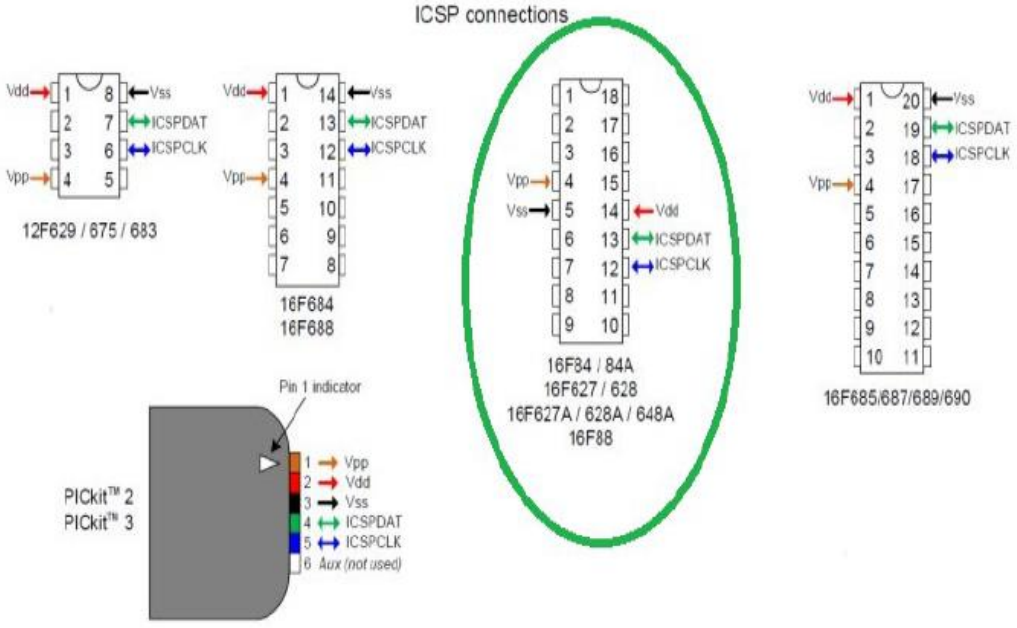


Imagen 3 - Conexionado del PIC al Programador PicKit

PIC 16F628A: Utilizamos este micro controlador ya que fue el que utilizamos durante el trascurso de la materia, la elección se debió exclusivamente al dominio que tenemos sobre este circuito integrado.

Imagen 4 - PIC 16F628A

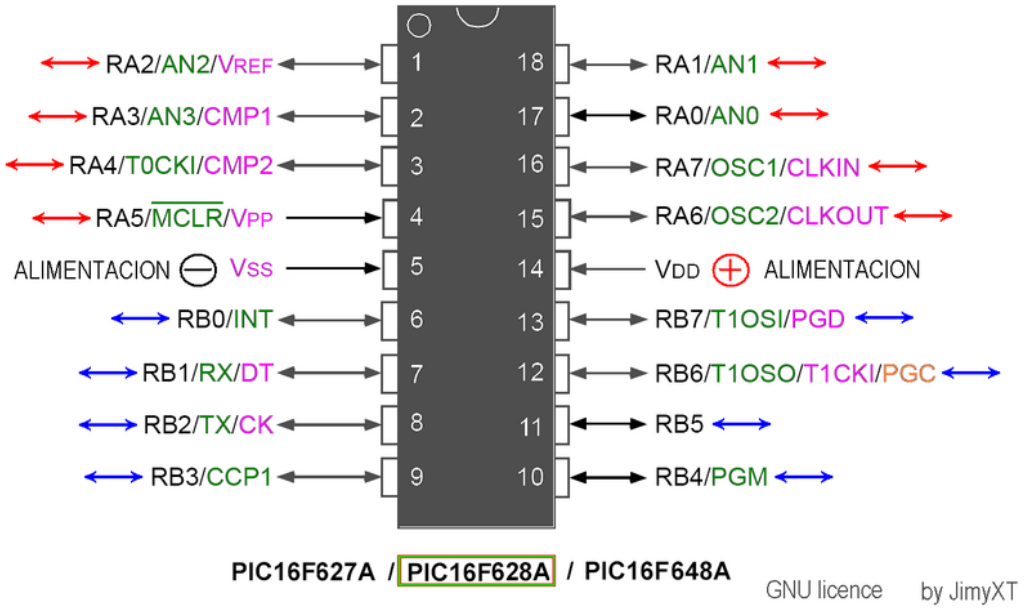


Imagen 5 - Puertos PIC 16F628A

Puente H: Para poner en funcionamiento el motor es necesario utilizar este integrado, el cual contiene cuatro resistencias, cuatro transistores y cuatro diodos. Según mostramos en la figura.

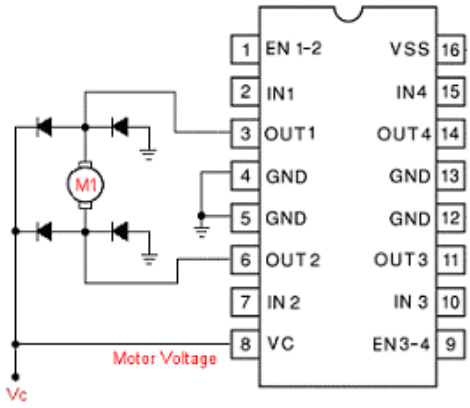


Imagen 6 - Conexión Puente H.

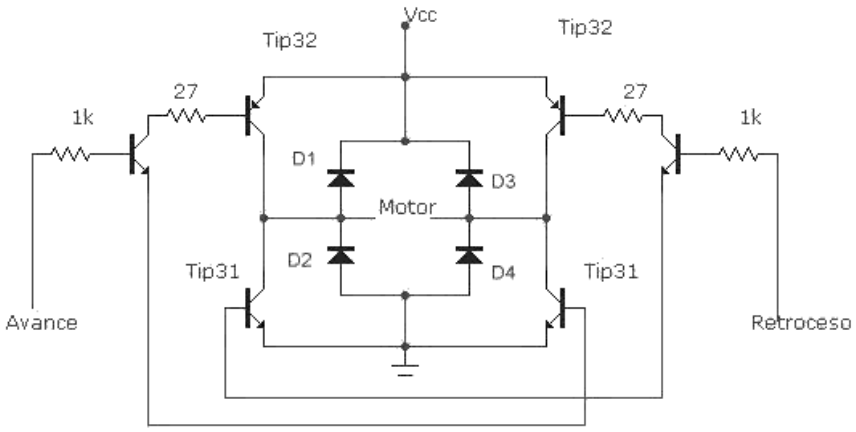


Imagen 7 - Arquitectura Puente H.

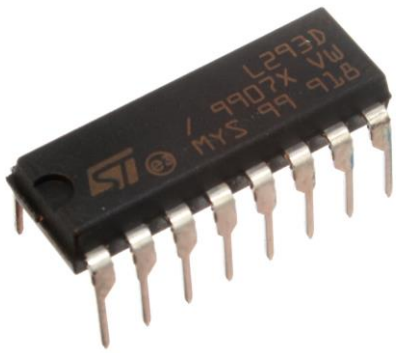


Imagen 8 - Puente H- L293D

Compuertas AND: Al tener un Pic 16F628A como Microprocesador, sabemos que tenemos solo un puerto PWM (Para poder controlar la velocidad del motor a través del ancho de pulso de la señal) – El puerto RB3. Es por eso que tuvimos que utilizar dos de estas compuertas para armar la lógica para el manejo de la maniobra SUBIR y BAJAR. Básicamente, lo que utilizamos, fue la salida del puerto RB3 como una entrada en cada una de la compuertas AND, por otra parte, la segunda entrada de cada compuerta estará afectada con la salida del PIC que realizan la alternancia 0 ó 1, para indicar sentido de giro horario o anti-horario y/o realizar la detención de motor.

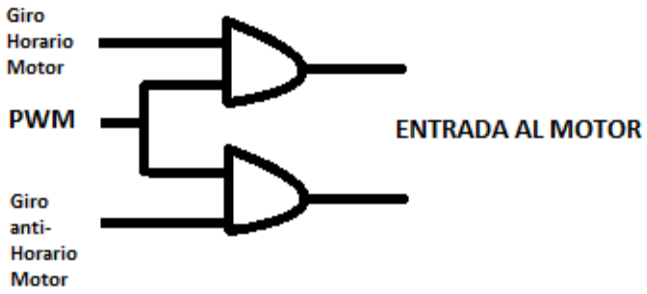


Imagen 9 - Esquema de Funcionamiento

Según podemos observar en la figura del esquema de funcionamiento, la lógica para darle energía al motor, debe llegarle un 1 y un 0 ó un 0 y un 1. Si recibe dos 1 ó dos 0 el motor se detendrá automáticamente. Vemos como cuando activamos el PWM con un valor alto, la lógica depende pura y exclusivamente de las salidas del PIC en las cuales usamos el sentido Giro Horario Motor y Giro Anti-Horario Motor. Con eso obtendremos la ENTRADA AL MOTOR y podremos controlarlo.

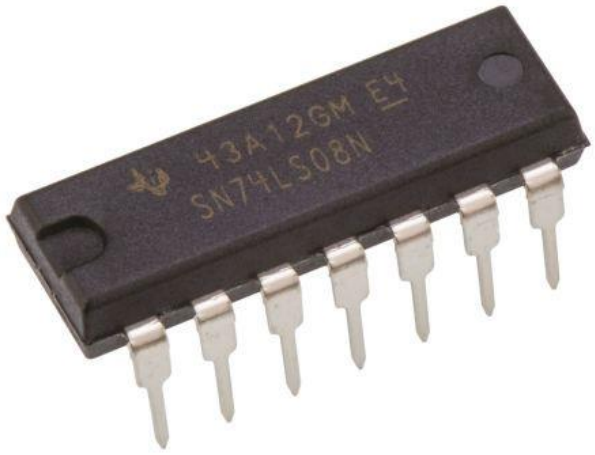


Imagen 10 - Compuerta AND SN74LS08N

Botones mecánicos: Los utilizamos para realizar las llamadas de los pisos desde la botonera de llamada y envíos. Cuatro Botones. Uno para acceder a cada uno de los tres pisos. Y el último para realizar una parada de emergencias en caso de algún desperfecto mecánico y/o eléctrico.

Imagen 11 - Botones Mecánicos

En el desarrollo del trabajo tuvimos que utilizar el concepto Resistencias Pull Down que aprendimos durante el trascurso de la materia. Al trabajar con botones mecánicos aprendimos que para este caso, se necesitaba armar un circuito con este tipo de conexionado de resistencias, pero para interiorizarnos en el concepto, explicaremos brevemente que significa PULL DOWN y PULL UP con un ejemplo de un LED y explicaremos el porqué de la elección PULL DOWN.

Resistencia Pull down

En la **configuración pull down**, cuando el circuito está **en reposo** como se muestra en la imagen 12, la caída de tensión en la resistencia es prácticamente **0V (LOW)**, en cambio sí **pulsamos P1**, dejará pasar la corriente y tendremos una diferencia de potencial de **5V (HIGH)**. Este es el uso normal del estado LOW y HIGH.

Resistencia Pull up

Por el contrario, en la **configuración pull up**, cuando el circuito está **en reposo**, P1 sin pulsar, la caída de tensión es de **5V (HIGH)**, en cambio cuando **pulsamos P1** se deriva toda la corriente a masa y la caída de tensión es **0V (LOW)**.

Normalmente **las resistencias que se utilizan en estos casos son de 10K**. Como hemos comprobado, **estas dos configuraciones nos evitarán que en estado de reposo midamos**

un valor erróneo eliminando la influencia de factores externos sobre nuestras mediciones como el ruido eléctrico.

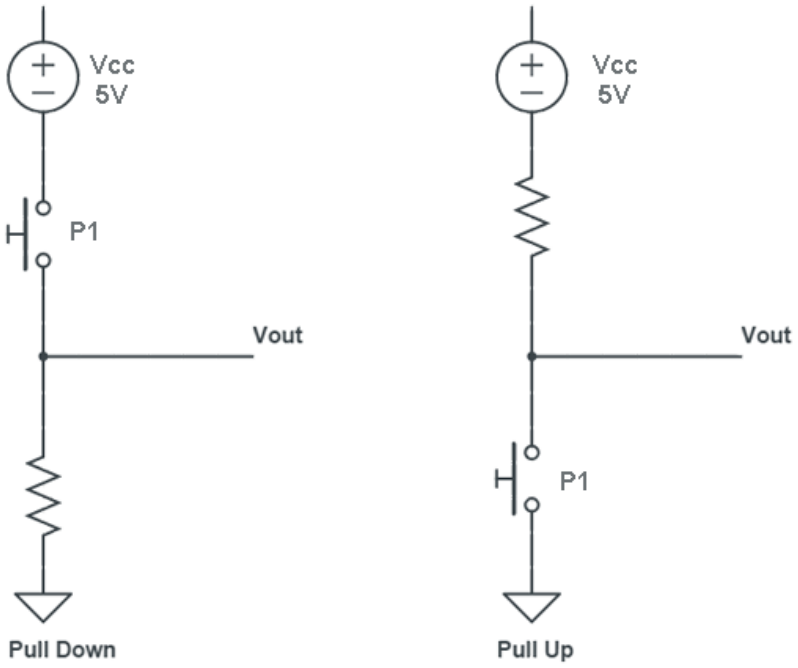


Imagen 12 - Resistencias Pull Down y Pull Up

El pulsador tiene cuatro patitas que están conectadas a pares, Cuando pulsamos el interruptor se cierra el circuito y dejamos pasar la corriente. Esto nos permite, por ejemplo, controlar un LED, un motor o cualquier otro elemento. La duda surge cuando dejamos la patita donde está conectado ese elemento al aire es decir, sin estar conectado a nada.

En este caso tiene un comportamiento inestable por no estar conectado a ninguna tensión, esto se conoce como **alta impedancia**. Para conseguir que el LED quede en un estado determinado cuando el pulsador esté abierto, debemos hacer uso de las resistencias **Pull-up ó Pull-down**.

Estos dos tipos de resistencias nos aseguran que cuando el interruptor esté abierto tengamos un valor HIGH (5V) o LOW (0V), pero ¿cómo lo hacen?

Empecemos por el caso de la resistencia pull-down. En este caso el LED (en la imagen 12 se muestra como Vout) tiene dos vías de escape. Por un lado podemos tomar como referencia la conexión que está al aire en el pulsador, de alta impedancia, o puede ir a la resistencia que está conectada a tierra.

Hay que pensar que en este caso, al tener una alta impedancia, sería como un muro a la hora de decidir el camino, por lo tanto se verá obligado a tomar como referencia el camino de la resistencia, el que tiene menor diferencia de potencial, está conectado a tierra (0V).
Esta es la configuración recomendada cuando queremos tener un valor HIGH al presionar el pulsador y un valor LOW al dejar de presionar el pulsador, el uso común.
Por otro lado, en la resistencia pull-up ocurre lo contrario. Mientras el pulsador esté sin presionar, la referencia que tomará el LED será de 5V, cuando se presiona el pulsador, la referencia que toma el LED es 0V debido a que está conectado a tierra.
La elección de Pull Down se debe por lo expuesto en los últimos dos párrafos. Queremos presionar un botón y que el ascensor funcione y se dirija hasta la posición deseada, el análisis es análogo con el LED, nos pareció que era claro explicarlo con un ejemplo práctico.

Carretel con cabezal de Impresora: Lo utilizamos como estructura del ascensor, aprovechamos el sistema de guías y el cabezal que simula ser la cabina del montacargas. Además la correa que corre a través del motor es lo que realiza el movimiento de arriba hacia abajo y viceversa (Simularía ser los cables de acero de tracción de los asensores). En el bajo recorrido posee otra polea para hacer fluido el movimiento de la correa.

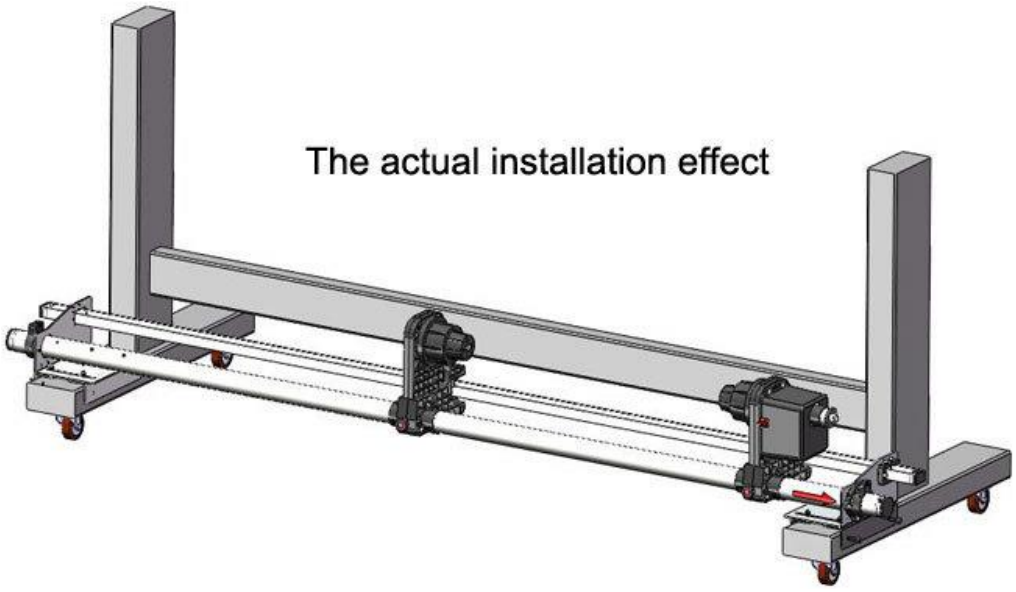


Imagen 13 - Carretel de impresora (Imagen Ilustrativa)

Motor DC: El motor utilizado estaba incorporado en el carretel de la impresora. Se trata de un motor de Corriente Continua convencional.

Imagen 14 - Motor DC (Imagen a modo ilustrativo)

Características técnicas:

- Voltaje de funcionamiento: 5V
- Resistencia interna: 10 Ω
- Corriente consumida (en vacío): 16,8 mA

LED: Los utilizamos para informar si el ascensor está subiendo o bajando. Uno para cada función. Por otra parte indicamos en qué posición se encuentra. PB,1° o 2° piso.

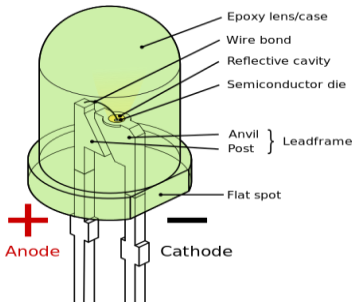


Imagen 15 - LED

Un LED es un componente optoelectrónico activo y, más concretamente, un diodo que emite luz.

LED fototransistor: En principio íbamos a utilizar este tipo de LED fototransistor para conocer la posición de la cabina del Montacargas. Luego se despreció su uso. (Se ampliara en el sector de Circuitos utiliados).

Imagen 16 - LED Fototransistor

Este fototransistor estaba pensado utilizarse como siempre activo, o sea que siempre está en 5v, en cuanto la cabina pasa por delante y le tapa la luz, se coloca en 0v y sabemos que se encuentra en ese piso o debe detener su marcha.

Módulo Display Catalex: Utilizamos este módulo para indicar la posición del ascensor. Es la interface que interactúa con el usuario, ya que te indica en donde se encuentra el ascensor, cuando ves por el visor. (Al estar reducido de puertos en el pic. Terminamos utilizando leds para indicar la posición del ascensor).

Imagen 17 - Módulo Display – Catalex

Cables varios: Los utilizamos para la conexión del PIC a los componentes respectivamente.

Imagen 18 - Cables varios

Resistencias: Las utilizamos para reducir tensiones y como PULL DOWN para los botones.

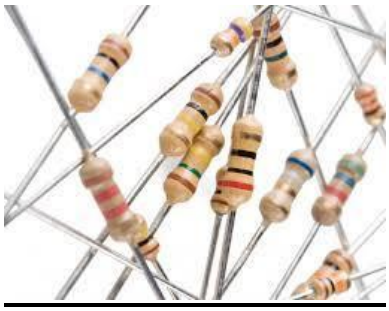


Imagen 19 - Resistencias Varias

Plomada de pesca: La utilizamos para que el ascensor tenga el contrapeso necesario para que el mismo quede estable en cada piso. (Se ampliara su uso en la Experiencia).

Imagen 20 – Plomada de pesca

Panel Solar: Fue probado para detecte la luz de 3 leds blancos ubicados al costado de la cabina del ascensor para obtener dos estados lógicos (0 Y 1) y así intentar detener el motor en el piso adecuado. (Se ampliara en el sector de Circuitos utilizados)

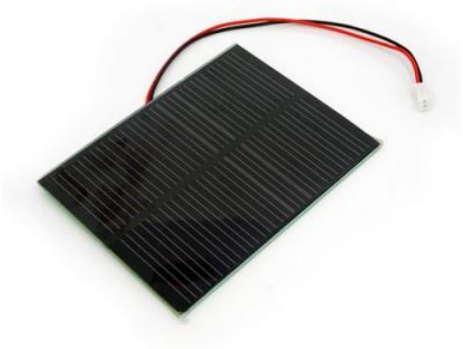


Imagen 21 – Panel Solar

Transistores: Los utilizamos para amplificar la tensión de salida del circuito del panel solar para que la entrada del PIC pueda detectar los dos estados lógicos (0 y 1).

a) T**ransistor NPN BC548**

Características técnicas:

- Tensión emisor-colector hasta 30V.
- Corriente de colector máxima 100mA.
- Disipación máxima 625mW.
- Tensión base-colector hasta 30V.
- Temperatura de operación ambiente: -55 a 150 ºC.

Imagen 21 – Transistor BC 548

b) T**ransistor PNP BC558**

Características técnicas:

- Tensión emisor-colector hasta -30V.
- Corriente de colector máxima -100mA.
- Disipación máxima 625mW.
- Tensión base-colector hasta -30V.
- Temperatura de operación ambiente: -55 a 150 ºC.

Imagen 22 – Transistor BC 558

Circuitos Utilizados Para Paradas

Durante las pruebas realizadas para que el ascensor suba y baje correctamente mediante los botones de piso y su respectiva parada de emergencia; utilizamos tres circuitos diferentes para lograr el control del motor y que se detenga correctamente en cada piso. El primero fue un circuito con fototransistores. Durante las pruebas llegamos a la conclusión de que el mismo no nos servía para el objetivo adecuado ya que su tiempo de respuesta era bastante lento. El segundo fue más sofisticado ya que con la utilización del panel solar los tiempos de respuesta del mismo eran casi instantáneos (al recibir la luz y la oscuridad); lamentablemente dicho circuito entregaba poca tensión a la salida, lo cual al utilizar el circuito con los dos transistores para amplificar la tensión se adicionaba mucho tiempo de retardo entre el valor lógico 1 y el valor 0 por lo tanto este circuito también quedo descartado.

Finalmente, luego de varios intentos fallidos, se nos ocurrió recurrir a un método clásico pero infalible. Realizando tres agujeros en la cabina para pasar un cable conectado a 5V con el filamento del mismo al descubierto y colocar otro cable en la cabina autoportante para que el mismo haga contacto con el cable de 5V, a medida que vaya subiendo o bajando para detectar los 0V y 5V y así saber en que piso esta ubicado para indicarle al PIC que el motor debe detenerse en ese lugar.

Encontramos la respuesta mecánica, luego de varios intentos y utilizando diferentes delay's, para encontrar la mejor alternativa y mejor nivelación en cada uno de los pisos.

a) Circuito con fototransistores:

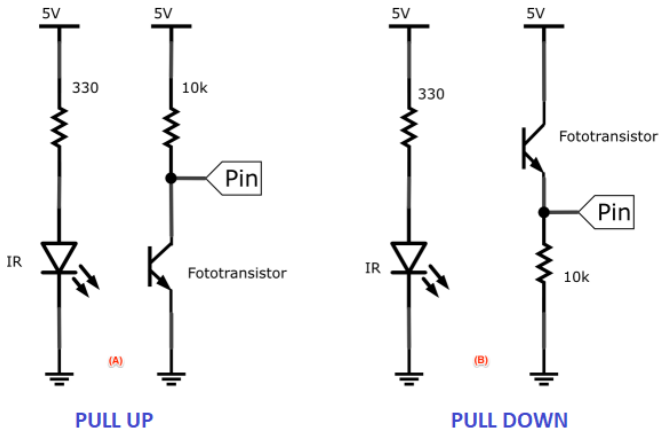


Imagen 23 – Circuitos con fototransistores

Existen dos tipos de configuración para utilizar el fototransistor PULL UP y PULL DOWN. En el primero el pin de entrada del PIC detecta un estado lógico 1 cuando el fototransistor se encuentra a oscuras y un estado lógico 0 cuando el fototransistor recibe luz ya que toda la corriente se va a masa (GND). En cambio en la segunda el pin de entrada del PIC detecta un 1 lógico cuando el fototransistor recibe la luz y un 0 cuando esta oscuro ya que la resistencia conectada a masa (GND).

Con la configuración PULL UP lo que hicimos fue colocar 3 fototransistores en el lado izquierdo de la cabina y colocar una cabina autoportante lo suficientemente ancha como para que, a su paso, logre oscuridad en el fondo de la cabina (donde se encuentran los 3 fototransistores). Luego de varias pruebas llegamos a la conclusión de que este sistema era demasiado lento para la aplicación de nuestro circuito así que quedó descartado.

b) Circuito con panel solar:

El siguiente circuito posee una pila AAA recargable de 1.2 V, un inductor de baja inductancia, un panel solar de baja tensión y un led testigo. Su funcionamiento es bastante sencillo, rápido y eficiente. Cuando el panel solar recibe la luz solar, el inductor y la batería se cargan con energía eléctrica y el led testigo permanece apagado. En cambio, cuando el panel solar recibe oscuridad, el inductor se descarga otorgándole la corriente necesaria al led para que se encienda.

Para la implementación de este circuito, colocamos 3 leds blancos de alta luminosidad en el costado izquierdo de la cabina y en la cabina autoportante pegamos el panel solar.

Las pruebas resultaron exitosas, el panel detectaba instantáneamente la luz del led y la oscuridad a medida que iba subiendo o bajando.

Este circuito solamente nos entregaba entre 0 y 1.2 V y según el datasheet de nuestro PIC 16F628A la tensión mínima que debe recibir una entrada del pic para detectar un 1 lógico es de 3.65V.

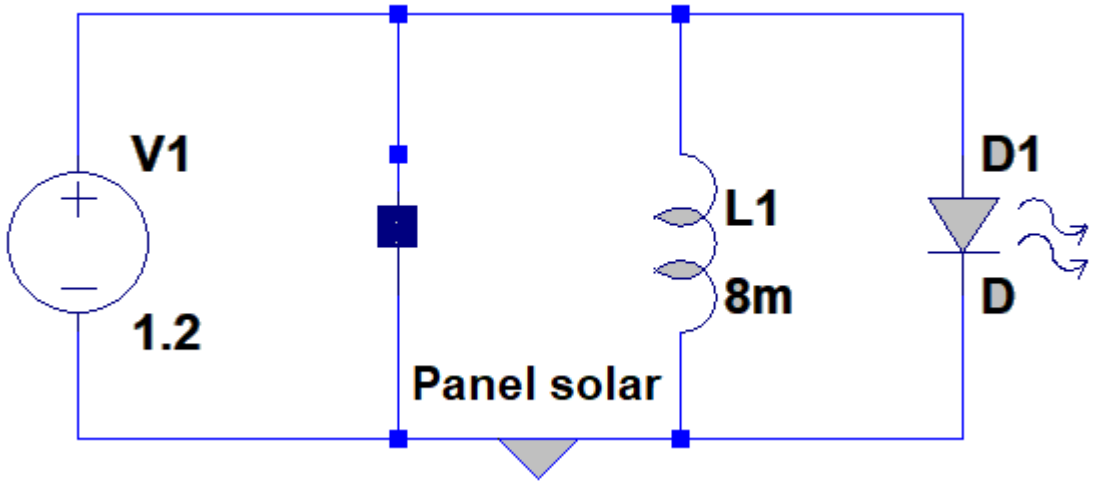


Imagen 24 – Circuito con panel solar

Para amplificar la tensión utilizamos el siguiente circuito:

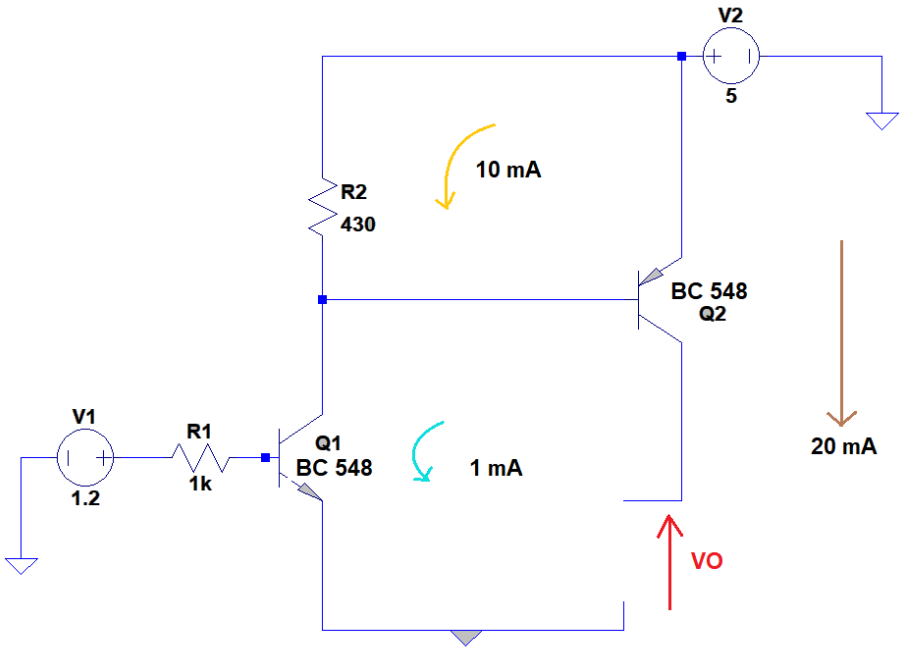


Imagen 25 – Circuitos amplificador de tensión

Según los datasheets de ambos transistores, el transistor Q1 necesita una corriente de base de 1 mA para que se sature y así cerrar la llave entre el colector y el emisor. Esta corriente es la que ingresa a la base del transistor Q2 haciendo que el mismo se sature brindando los 5V a la salida, necesario para nuestro pin de entrada del PIC. Cuando el led esta apagado hay 0V en V1 y una IB nula y por lo tanto en la salida del circuito hay 0V.

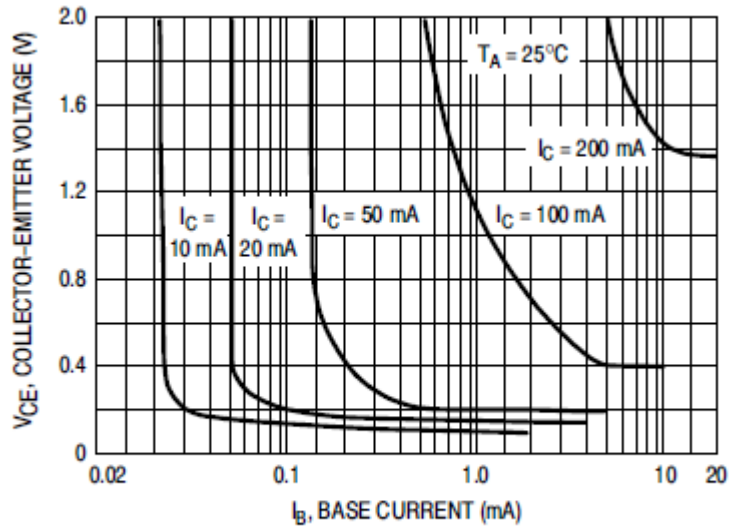


Figure 3. Collector Saturation Region

Imagen 26 – Grafico saturación BC 548

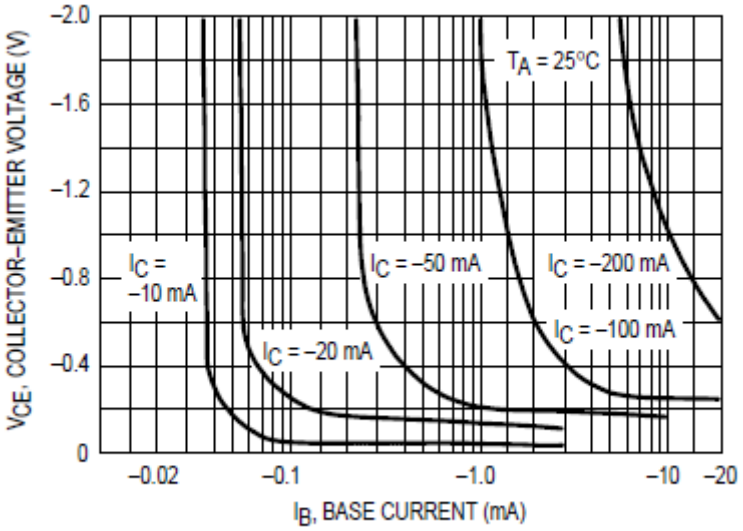


Figure 3. Collector Saturation Region

Imagen 27 – Grafico saturación BC 558

Este circuito finalmente tampoco nos sirvió ya que los transistores tienen muy baja performance para el uso del ascensor. El tiempo de retardo entre que los transistores pasan de corte a saturado es demasiado elevado, por lo que la respuesta que generaba para la parada del ascensor era excesiva y el ascensor se terminaba “pasando” de la parada

c) **Circuito con Filamento de Cables:**

Esta última prueba la hicimos perforando la cabina y colocando el cable con filamento en los tres pisos del ascensor. En la cabina autoportante colocamos un tarugo con otro cable con filamento para que el mismo haga contacto con el cable que esta conectado a 5V pegado a la cabina. Este sistema es utilizado en los ascensores “antiguos” con los sistemas de selectores verticales, es el mismo método realizado a escala. Cuando hay un contacto, hay un 1 instantáneo. Para que el sistema no detecte falsos 1 lógicos o falsos 0, tuvimos que agregar delays en el código fuente y con eso las pruebas realizadas resultaron satisfactorias.

Imagen 28 – Cabina con el cable con filamento

Consumos

Motor en vacío	16.8 mA
Compuerta AND sin carga	27 mA
Pico de Arranque Motor en Subida	487 mA
Pico de Arranque Motor en Bajada	183 mA
Reposo del circuito (Cabina Abajo)	61 mA
Reposo del Circuito (Cabina Arriba)	36 mA

Experiencia

Con respecto a la experiencia en este trabajo, como en cada uno de los que nos tocó encarar a lo largo de estos trayectos de electrónica, siempre se complica ensamblar todo el conjunto. Las conexiones de los circuitos por separado funcionan perfectamente, pero al armar todo el circuito nos vemos con complicaciones para que todo actuara en simultáneo. Por otra parte, teníamos todo diagramado en papel cada uno de los componentes que íbamos a utilizar, como íbamos a conectarlos, como diseñábamos el circuito, después por problemas físicos (Falta de puertos para el indicador de 7 segmentos) o por problemas de proveedores (No conseguimos el puente H conocido sino uno integrado), además del diseño de la estructura del hueco del ascensor. El armado de la estructura, en principio utilizamos una maqueta de cartón para poder hacer pruebas unitarias y probar los circuitos, después trasladar todo hacia la estructura realizada con la impresora 3D era desconectar y volver a conectar y atornillar nuevamente. Con la maqueta terminada, vinieron los problemas mecánicos, en las pruebas unitarias, se probó siempre el cabezal sin peso (cabina), por lo que al agregarle algunos gramos, las paradas en los pisos superiores se hacía inestable por lo que por la propia gravedad de la ahora cabina, se iba hacia abajo. Tuvimos que recurrir a implementar un contrapeso con un sistema básico de polea en “la terraza” de la estructura. Encontrar el peso exacto era imposible, por no contar con una balanza, por lo que tuvimos que ir realizando distintas mediciones básicas de pesaje (a la vieja usanza, como la balanza de la justicia), si de un lado pesa más que del otro es porque hay más peso, probamos con distintas piezas mecánicas, y varias al mismo tiempo, Finalmente pudimos encontrar una pesa que cumpla con la función correcta y hasta pudimos contar con que el contrapeso fuese un poco más pesado, para poder cargar la cabina con algún objeto y así quede balanceado. Una de las peores situaciones por las que pasamos fue la parada del ascensor en los pisos. Teníamos el circuito para controlar el motor, ya subía y bajaba, pero no podíamos realizar las paradas en los pisos, como contamos en el apartado anterior recurrimos a dos actos fallidos antes de concretar con la parada final. Nuestro problema no era encontrar una resolución, ya que obteníamos la logia 5 o 0 volt, pero el problema era físico, el tiempo de respuesta de los circuitos era lento para realizar la parada abruptamente. Si bien fue caótica cada una de las tareas que ejecutamos, lo disfrutamos y quedamos a gusto con la terminación del proyecto final.

Conclusiones

Al terminar con el proyecto, podemos concluir que aunque tengamos los recursos necesarios tanto teóricos, como técnicos (Parte electrónica), hay cosas que requieren de otras áreas como por ejemplo la mecánica, las cuales no tenemos muchos conocimientos más que las ganas de querer lograrlo, nos faltaban herramientas e ideas. El montacargas diseñado debería poseer puertas tipo guillotina con cerraduras y a través de la lógica electrónica controlar estas cerraduras y autorizar a realizar una apertura de puertas o un cierre seguro, claramente por la disposición de la abertura de puertas, sumado a que la escala es muy chica como para simular un dispositivo de esta envergadura es que tuvimos que dejar el montacargas sin puertas.

Por otra parte, aunque contamos con un buen tiempo para la realización del trabajo, la vorágine de querer hacer un gran proyecto se fue reduciendo hasta los días previos de la entrega del proyecto.

Una apartado no menor es que en la industria se trabaja en conjunto con colegas de otras disciplinas y claramente no dejaran que los ingenieros en informática realicen el diseño de la estructura o las conexiones mecánicas y/o eléctricas de los artefactos de un ascensor, por lo que nuestra misión estuvo en realizar una maqueta similar a la de un Ascensor Montacargas, pero creemos que deberíamos dejarle el diseño a los experimentados, entiéndase este párrafo a modo de chascarrillo en clara alusión a la maqueta presentada.

Con ese concepto y realzando nuestras falencias, tenemos que augurar nuestras fortalezas. Creemos que hemos realizado un código limpio y lógico para el proyecto. Tratamos de enfocar a darle una vuelta más de rosca en darle un lote de llamadas en el código para poder demostrar que por más que la materia apunte a la arquitectura, el software es el gran aliado y se puede hacer mucho. Siempre desde nuestra perspectiva, colaboraremos en conjunto con los colegas de otras disciplinas para lograr grandes objetivos.

Como conclusión final, podemos estar conformes de entregar un trabajo terminado, con los requisitos básicos planteados y concluidos, habiendo utilizado los conocimientos previos de las materias anteriores y agregándole lo de la actual. Entendemos que se ha desarrollado una rica experiencia, ya que las pruebas y errores fallidos, nos hicieron dar cuenta de los grandes conocimientos que poseemos y así poder nutrirnos de herramientas para finalmente

lograr objetivos. Claramente, la construcción de los ingenieros se basa por una parte en poseer herramientas teóricas, pero también y no menos importante la adquisición de experiencia para enfrentarse al mundo con un abanico de soluciones, para luego terminar implementando la adecuada a cada una de las situaciones de la vida cotidiana.

Por último y para cerrar el proyecto, se adjuntaran una serie de anexos para dar por concluido el trabajo.

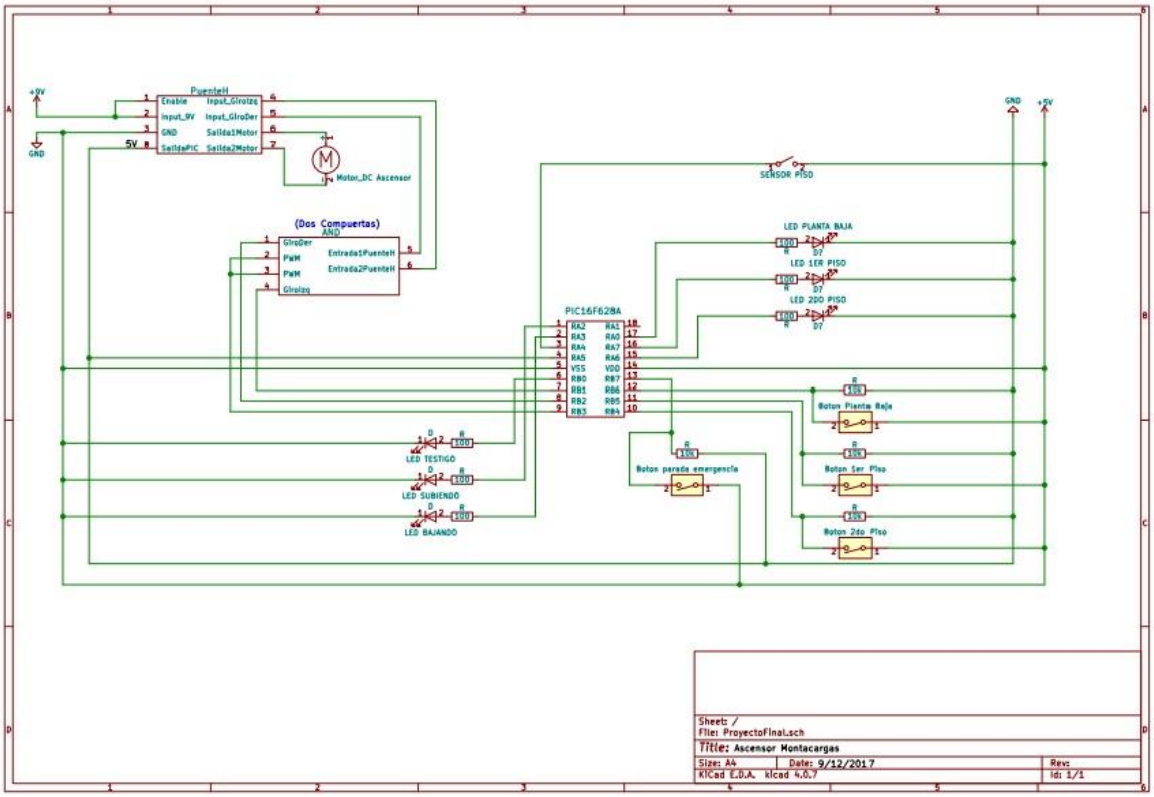
Mejoras a Desarrollar

Dentro de las mejoras a desarrollar en este proyecto podemos identificar las siguientes para convertirlo en un ascensor de trasporte de pasajeros:

- El agregado en la maqueta de un frente con pisos para que sea mas vistoso.
- El agregado de puertas tipo manuales, plegadizas o placas, cualquiera sea el caso con cerraduras electrónicas o puertas manejadas con servos electrónicos para apertura automática de puertas exteriores.
- Agregar una puerta en la cabina con cerradura (Puede ser plegadiza o automática).
- La conexión de tres cables de tracción como mínimo con una polea en el motor.
- La colocación de un regulador de velocidad mecánico con cable y polea en el bajo recorrido, con un contacto eléctrico para accionamiento del paracaídas (clavadas mecánicas) y corte de emergencia electrónico.
- La colocación de límites finales de carrera, para accionamiento de seguridades por corte mecánico.
- Colocación de botonera en cabina y en “los palieres”. (Deshacernos de la botonera de llamada y envío).
- Mejora en la circuitería y lógica electrónica, provocando movimientos menos bruscos en las paradas y aceleración del ascensor a través de aceleración en 2 velocidades o controlados con un variador de frecuencia.
- Cambiar la circuitería y lógica del programa para realizar nuevas maniobras, abandonar la usanza de maniobra simple (atiende una llamada por vez), por una maniobra colectiva en descenso o colectiva total (atiende todas las llamadas en descenso y/o más conveniente según los movimientos a realizar) – (Para este caso se deberían agregar más paradas en el ascensor para que se pueda evaluar y demostrar el comportamiento para el uso de muchos pisos). Para la maniobra total, se deberán agregar un botón más en todas las paradas intermedias y manejar la memoria del programa para tomar un total de llamadas “X”. tanto de la botonera de cabina como las exteriores.

- Colocar dispositivos de seguridad como pesadores de carga de cabina (balanza electrónica), barrera infrarroja de seguridad para apertura de puerta de cabina en caso de obstrucción (que un pasajero quiera ingresar a la cabina).
- Colocar dispositivos de sensor de parada de pisos más modernos para no depender de un contacto y cierre de circuito. En la industria de ascensores, se utilizan dos inductores magnéticos que viajan en la cabina y chapas e imanes en cada uno de los pisos para realizar la correcta nivelación.

Circuito Final



Bibliografía

- Fotocopia de libros proporcionados por la cátedra.
- Imágenes varias de www.google.com.ar
- Ordenanza Municipal N° 49308 - Ascensores
- Código de la edificación de la Ciudad de Buenos Aires.

Apéndice

Código Fuente Montacargas Ascensor

```
// CONFIG

#pragma config FOSC = INTOSCIO  // Seleccion de bits del oscilador
#pragma config WDTE = OFF       // Bit de habilitacion del watchdog
#pragma config PWRTE = ON       // Bit de habilitacion del timer de encendido
#pragma config MCLRE = ON       // Bit de seleccion de funcion RA5/MCLR/VPP
#pragma config BOREN = OFF      // Bit de deteccion de apagado
#pragma config LVP = OFF        // Low-Voltage Programming Enable bit (
#pragma config CPD = OFF        // Bit de proteccion de datos de memoria (Data memory code
protection off)
#pragma config CP = OFF         // Bit de proteccion de memoria (Code protection off)


#include <xc.h>

#define _XTAL_FREQ 12000000 // Indicamos a que frecuencia de reloj esta funcionando el micro
void configurarCicloTrabajo(int valor);
void llamaronPlantaBaja();
void llamaronPrimerPiso();
void llamaronSegundoPiso();
void paradaEmergencia();
void paradaPorPiso(int cantidadPisos);
void paradaPrimerPiso(int cantidadPisos);
void bajarAscensor(int cantidadSensores);
void subirAscensor(int cantidadSensores);
void agregarLlamadaACola(int pisoLlamada);
void inicializarColaLlamadas();
void procesarLlamadas();
int cantidadDeLlamadas();
void actualizarCola();
void bajar();
void subir();
void parar();
interrupt void interrupcion();

int pisoActual = 2;
int colaLlamadas[3];

int main() {

    TRISBbits.TRISB0 = 0; //LED INDICADOR DE INICIO PROGRAMA
    TRISBbits.TRISB1 = 0; //SALIDA 1 MOTOR
    TRISBbits.TRISB2 = 0; //SALIDA 2 MOTOR
    TRISBbits.TRISB3 = 0; //PWM MOTOR

    TRISBbits.TRISB7 = 1; //BOTON PLANTA BAJA
    TRISBbits.TRISB6 = 1; //BOTON PRIMER PISO
    TRISBbits.TRISB5 = 1; //BOTON SEGUNDO PISO
```

```
    TRISBbits.TRISB4 = 1; //PARADA DE EMERGENCIA

    TRISAbits.TRISA1 = 0; //Indicador PB
    TRISAbits.TRISA0 = 0; //Indicador 1
    TRISAbits.TRISA7 = 0; //Indicador 2

    TRISAbits.TRISA2 = 0; //LED SUBIR
    TRISAbits.TRISA3 = 0; //LED BAJAR

    TRISAbits.TRISA4 = 1; //SENSOR PISOS

    CCP1CON = 0x0C; // Configure CCP1 module in PWM mode
    PR2 = 0xFF; // Configure the Timer2 period
    T2CON = 0x01; // Set Prescaler to be 4, hence PWM frequency is set to 4.88KHz.
    parar();
    T2CON |= 0x04; // Enable the Timer2, hence enable the PWM.

    PORTBbits.RB3 = 1; // Se activa el PWM
    PORTBbits.RB0 = 1; // Se prende led testigo programa
    inicializarColaLlamadas();


    INTCON = 1001100; // Bit 7: Habilita las interrupciones, Bit 4: Habilita el rb0 Bit3: habilita RB7,
RB6, RB5 y RB4
    llamaronPlantaBaja();
    while (1) {
        procesarLlamadas();
    }
    return 0;
}

void llamaronPlantaBaja() {
    switch (pisoActual) {
        case 1:
            pisoActual = 0;
            bajar();
            paradaPorPiso(1);
            break;
        case 2:
            pisoActual = 0;
            bajar();
            paradaPorPiso(2);
            break;
    }
    PORTAbits.RA1=1;
    PORTAbits.RA0=0;
    PORTAbits.RA7=0;

}

void llamaronPrimerPiso() {
    switch (pisoActual) {
        case 0:
            pisoActual = 1;
            subir();
            paradaPorPiso(1);
```

```
            break;
        case 2:
            pisoActual = 1;
            bajar();
            paradaPrimerPiso(1);
            break;
    }
    PORTAbits.RA1=0;
    PORTAbits.RA0=1;
    PORTAbits.RA7=0;

}

void llamaronSegundoPiso() {
    switch (pisoActual) {
        case 0:
            pisoActual = 2;
            subir();
            paradaPorPiso(2);
            break;
        case 1:
            pisoActual = 2;
            subir();
            paradaPorPiso(1);
            break;
    }
    PORTAbits.RA1=0;
    PORTAbits.RA0=0;
    PORTAbits.RA7=1;
}

void paradaPrimerPiso(int cantidadPisos) {
    __delay_ms(100);
    while (cantidadPisos != 0) {
        if (PORTAbits.RA4) {
            cantidadPisos--;
        }
    }
    parar();
}

void paradaPorPiso(int cantidadPisos) {
    __delay_ms(100);
    while (cantidadPisos != 0) {
        if (PORTAbits.RA4) {
            __delay_ms(50);
            cantidadPisos--;
        }
    }
    parar();
}

void paradaEmergencia() {
    parar();
    inicializarColaLlamadas();
}
```

```
void subir() {
   PORTAbits.RA2 = 1;
   PORTAbits.RA3 = 0;
   configurarCicloTrabajo(825);
   PORTBbits.RB1 = 1;
   PORTBbits.RB2 = 0;
}

void bajar() {
   PORTAbits.RA2 = 0;
   PORTAbits.RA3 = 1;
   configurarCicloTrabajo(500);
   PORTBbits.RB1 = 0;
   PORTBbits.RB2 = 1;
}

void parar() {
   PORTAbits.RA2 = 0;
   PORTAbits.RA3 = 0;
   configurarCicloTrabajo(0);
   PORTBbits.RB1 = 0;
   PORTBbits.RB2 = 0;
}

void inicializarColaLlamadas() {
   for (int i = 0; i < 3; i++) {
      colaLlamadas[i] = -1;
   }
}

void agregarLlamadaACola(int pisoLlamada) {
   int i = cantidadDeLlamadas();
   colaLlamadas[i] = pisoLlamada;
}

int cantidadDeLlamadas() {
   int cantidadLlamadas = 0;
   for (int i = 0; i < 3; i++) {
      if (colaLlamadas[i] != -1) {
         cantidadLlamadas++;
      }
   }
   return cantidadLlamadas;
}

void procesarLlamadas() {

   if (cantidadDeLlamadas() != 0) {
      switch (colaLlamadas[0]) {
         case 0:
            llamaronPlantaBaja();
            break;
         case 1:
            llamaronPrimerPiso();
            break;
```

```
            case 2:
                llamaronSegundoPiso();
                break;
        }
        actualizarCola();
    }
}

void actualizarCola() {
    __delay_ms(300);
    colaLlamadas[0] = colaLlamadas[1];
    colaLlamadas[1] = colaLlamadas[2];
    colaLlamadas[2] = -1;
}

/*
 * Configurar el ciclo de trabajo (PWM), el valor debe estar entre 0 y 1024
 */
void configurarCicloTrabajo(int valor) {
    CCPR1L = valor >> 2; // Put MSB 8 bits in CCPR1L
    CCP1CON &= 0xCF; // Make bit4 and 5 zero
    CCP1CON |= (0x30 & (valor << 4)); // Assign Last 2 LSBs to CCP1CON
}

interrupt void interrupcion() { //Declaracion y llamado a la funcion:
    if (INTCONbits.RBIF == 1) {//Consulta el valor de RB 4 5 6 7
        if (!PORTBbits.RB4) {
            if (cantidadDeLlamadas() < 3) {
                if (PORTBbits.RB7) {
                    agregarLlamadaACola(0);
                }
                if (PORTBbits.RB6) {
                    agregarLlamadaACola(1);
                }
                if (PORTBbits.RB5) {
                    agregarLlamadaACola(2);
                }
            }
        } else {
            procesarLlamadas();
        }

    }
    INTCONbits.RBIF = 0;
}
```