Universidad del cauca



Informe trabajo Ascensor de 3 pisos

Sebastián López Fernández

Juan Daniel Mosquera

Andrés Fernando Hoyos

Alumnos Tecnología en Telemática

Johana Andrea Hurtado Sánchez

Docente, Circuitos Digitales II

Departamento de Telemática

Facultad de ingeniería Electrónica y Telecomunicaciones

Popayán, 25 de noviembre de 2017

Índice

1. Requerimiento del trabaja
2. Introducción
3. Circuito esquemático y explicación de las conexiones
4. Diagrame de flujo
5. Explicación del código en ensamblador
6. Fotos de la maqueta
7. Conclusiones
8. Anexo
9. Bibliografía
10. El requerimiento para nuestro trabajo es el siguiente:

Se necesita un ascensor de 3 pisos, el cual va a estar por defecto en el piso 1, solo va a atender una llamada, si se presiona más de un botón, solo va a efectuar la acción del primer pulsador, va a tener 4 leds, uno de ellos va a estar encendido, indicando que el ascensor espera recibir un llamado a alguno de los pisos, otro led se va a encender en el momento en el cual el ascensor este subiendo, otro led se va a encender en el momento en que el ascensor se detenga, indicando que se están abriendo las puertas y finalmente un último led se va encender indicando que el ascensor esta bajando.

Adicionalmente un display 7 segmentos va a indicar el piso en el que se encuentra el ascensor, en caso de que el ascensor este en un piso, ya sea el primero, segundo o tercero y se presione el botón del piso en que se encuentre actualmente a encender el led, que indica que se abren las puertas.

1. Introducción

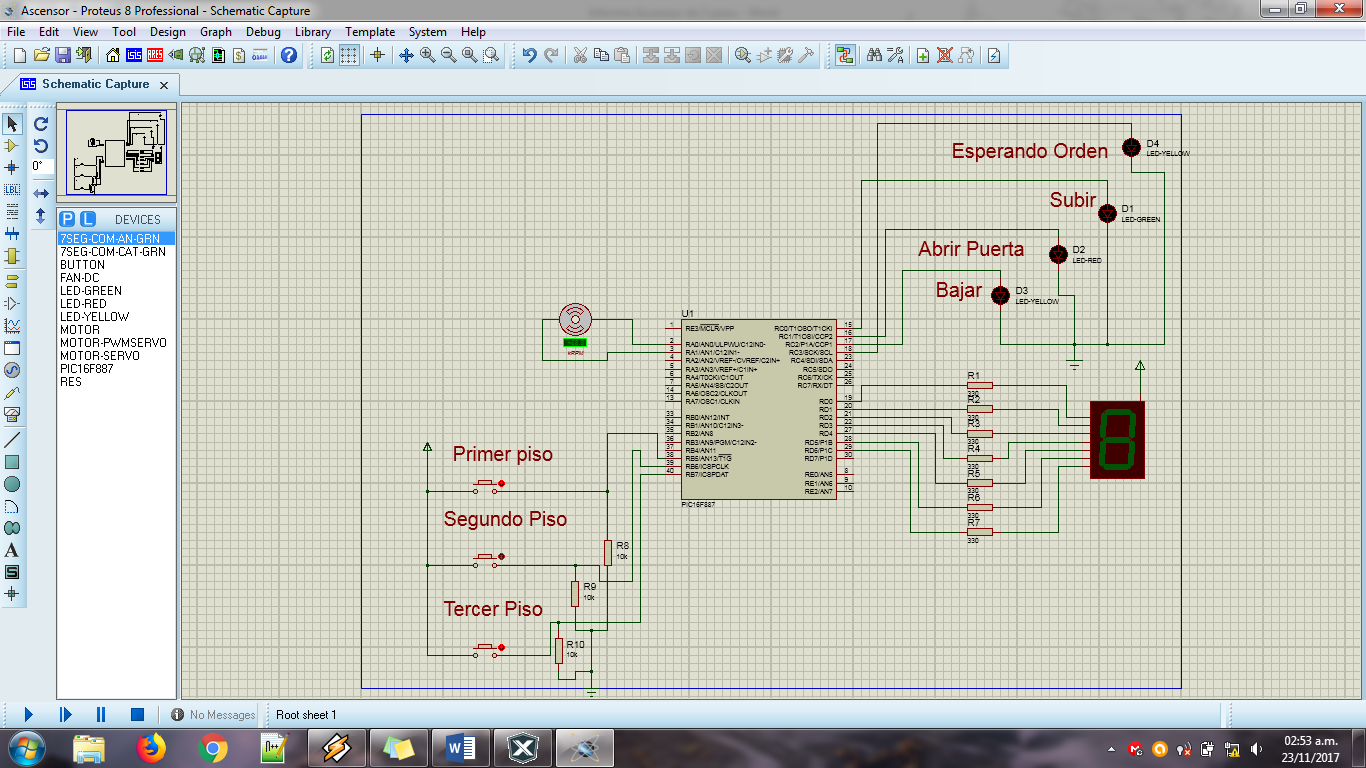
El siguiente informe es realizado para dar una explicación e idea, de cómo construir un ascensor de 3 pisos, claramente teniendo en cuenta el curso de circuitos digitales II, en donde se estará controlando el micro controlador PIC16F887 por medio del lenguaje de programación ENSAMBLADOR.

En principio se mostrará la conexión en el micro controlador ya mencionado, en cada uno de sus puertos, cuáles son sus conexiones, que dispositivos electrónicos distintos a los vistos en el curso de Circuitos Digitales 2 se van a utilizar, y finalmente la explicación del código desarrollado en el lenguaje ensamblador.

1. Circuito esquemático y explicación de las conexiones

Para hacer el diseño del ascensor primero hacemos el diseño de cómo va a quedar en la vida real, dicho de otra manera vamos a ver el circuito esquemático, para este proyecto usamos el simulador “Proteus”, el cual es una herramienta muy buena ya que este simulador se aproxima mucho a la vida real.

Este es el circuito esquemático que diseñamos para nuestro proyecto



En este circuito y en la protoboard vamos a conectar los pulsadores en el puerto B, en los bits más significativos es decir en los pines RB7, RB6 y RB5, conectados con resistencias de 10Kohmios.

En el puerto A vamos a conectar el servomotor, en el pines RA0 y RA1, para nuestro caso el servomotor va a ser un futaba 3003s, que gira 180 grados, pero debido a nuestra necesidad de que gire más de 180 grados, con unas modificaciones, hechas al servomotor, se ha logrado que tenga una rotación de más 360 grados, es decir se ha logrado que el servomotor puede dar más de una vuelta sin ninguna complicación.

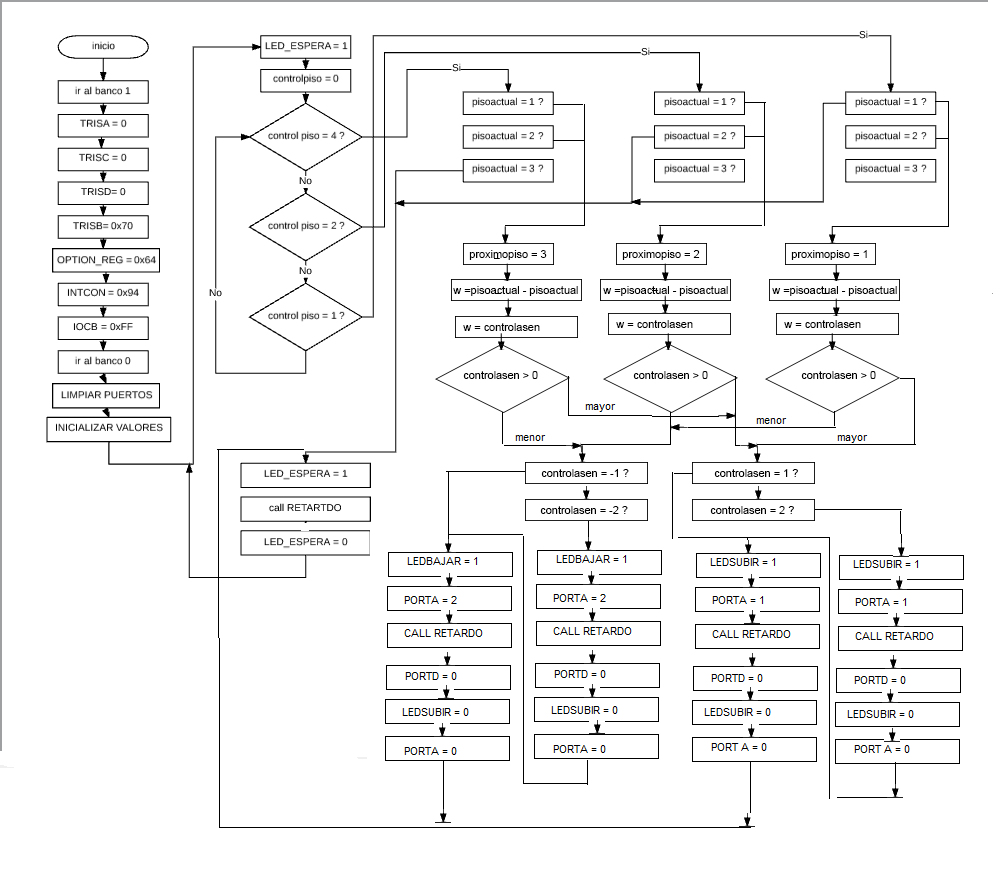
Para este caso, que se necesita que un ascensor suba o baje la cabina se va a desarrollar un programa para que a través del micro controlador PIC16F887 se manden pulsos al servomotor para que pueda girar en un sentido o en otro, dependiendo si quiere subir o bajar.

En el puerto C se van a tener conectados 4 leds, en los cuales se va a indicar si el ascensor está ascendiendo o descendiendo o en que caso de que haya llegado a su destino se va encender otro led que indica que se abren las puertas y finalmente va a estar otro led que se enciende cuando el proceso de ascender o descender haya terminado, este último led se va a encender indicando que el ascensor está listo para recibir una nueva orden.

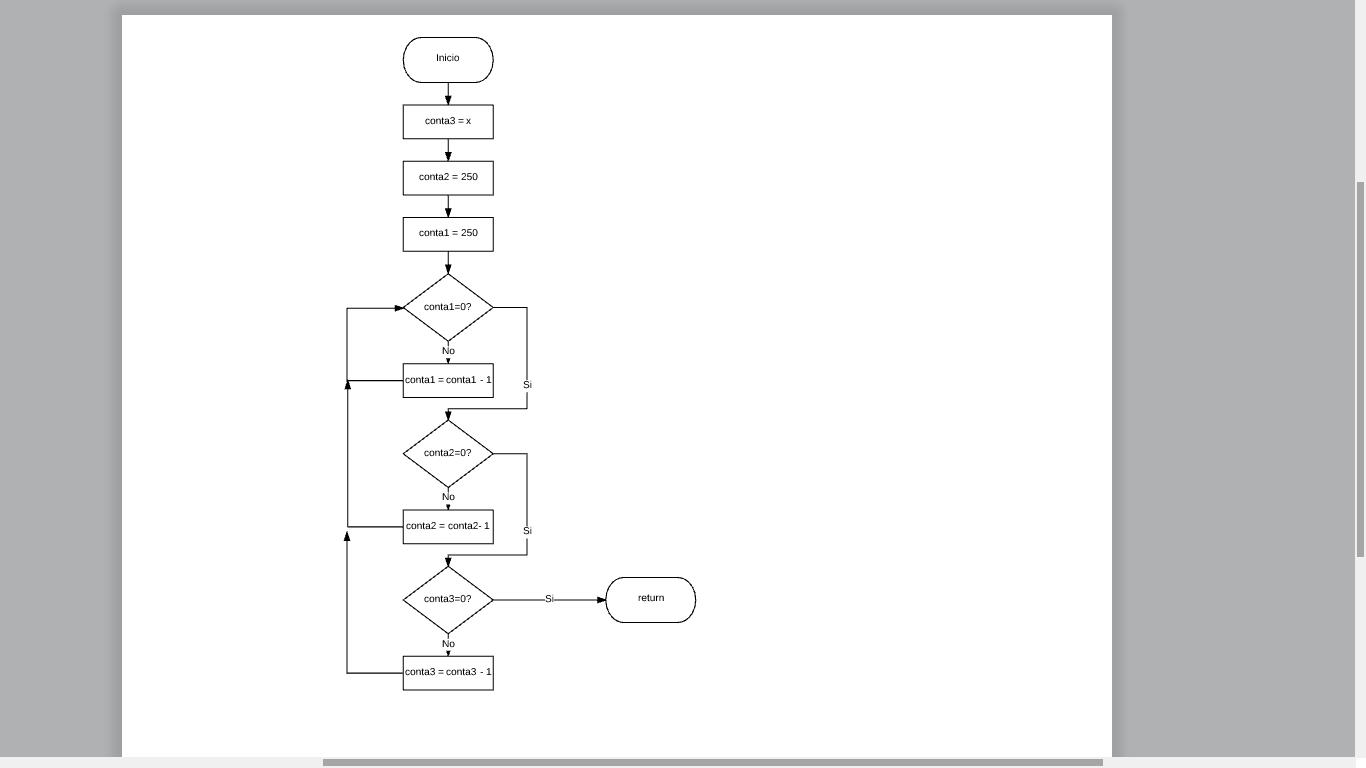
Finalmente en el puerto D va a estar conectado en display 7 segmentos, para nuestro caso va a ser un display de ánodo común, el cual al iniciar el programa va a mostrar la letra “A”, siendo esta la inicial de ascensor, después de pulsar un botón, el display 7 segmentos a indicar el piso en el que esta y al llegar al destino va a poner el número del piso en que se encuentra.

Adicionalmente se va a conectar un puente h entre el micro controlador y el servomotor. Esto es con el objetivo de que los pulsos que se le manda del pic, mediante los pulsadores al servomotor, los pueda hacer sin problema alguno, es decir que el servomotor, pueda girar a favor o en contra de las manecillas del reloj y también se pueda detener.

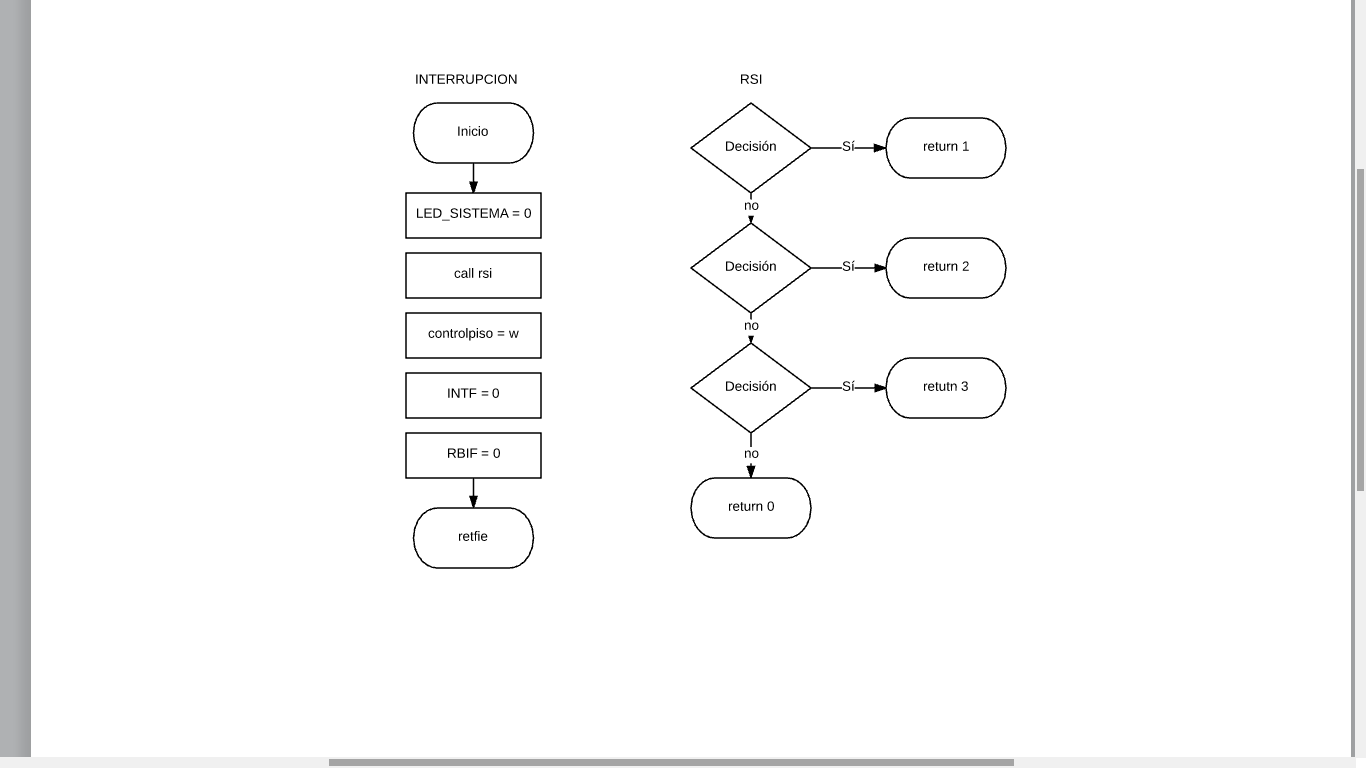
1. Diagrame de flujo



Retardo

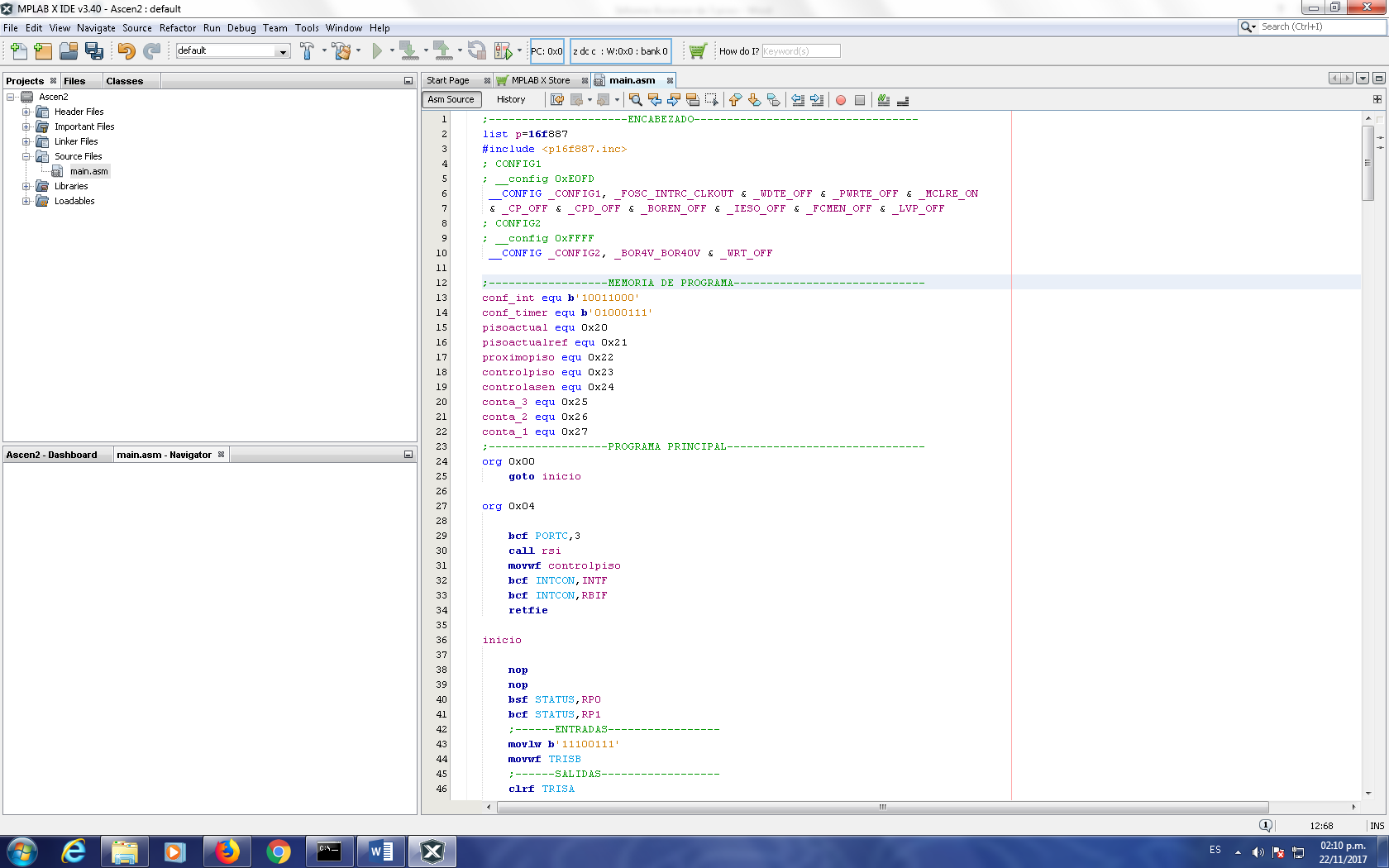


Interrupciones

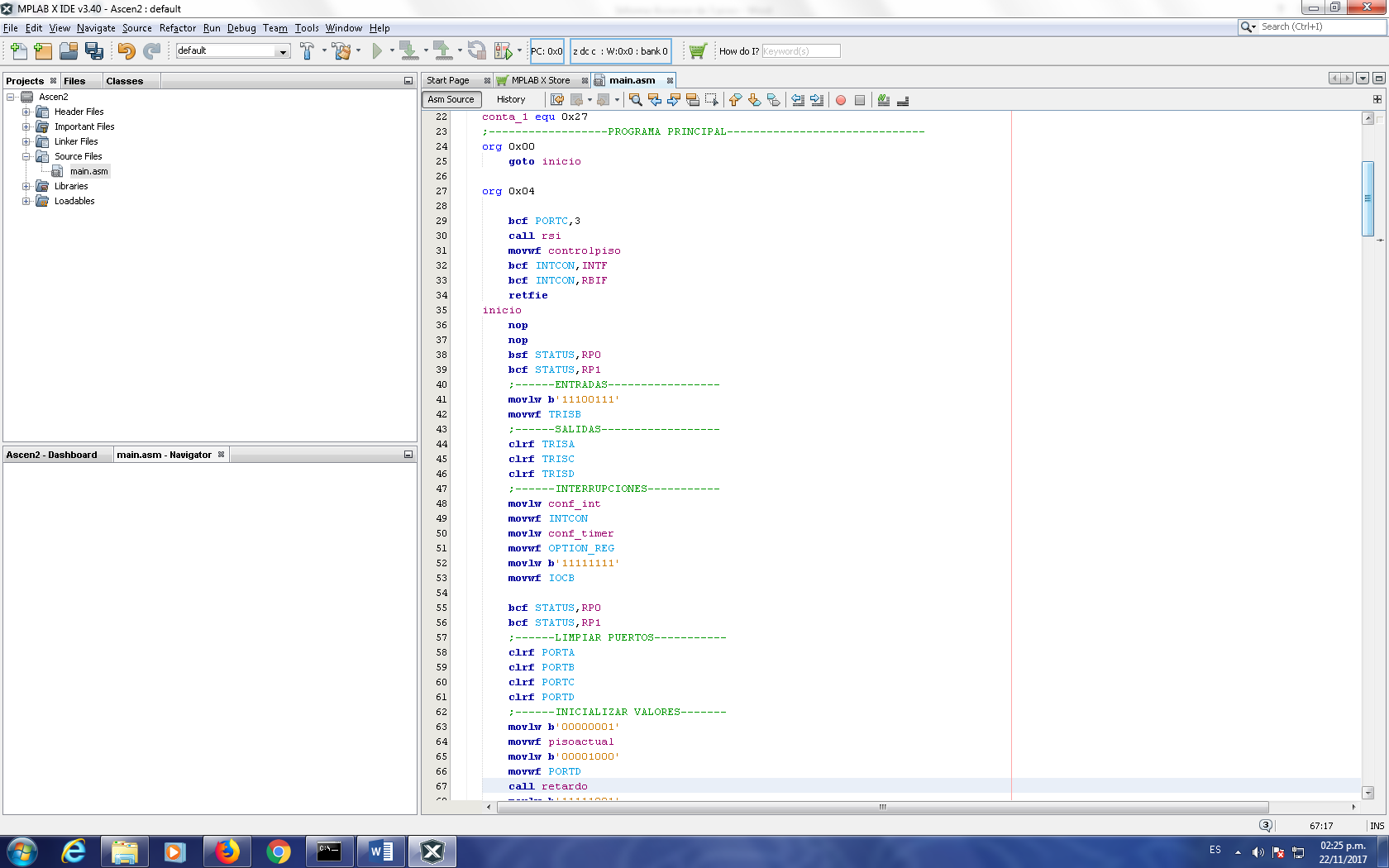


1. Explicación del código en lenguaje ensamblador

Para empezar el código se hace el encabezado, es decir listar en pic que se va a quemar el ascender de 3 pisos, también se va a generar el código, en el cual se le indica al quemador que el PIC16F887 tiene un reloj interno, para este caso va a ser de 4 MHz.



Después de esto se va a hacer el programa principal, en el cual se va a ingresar al banco 1, declarar los puertos de salida, que serán los puertos A, C y D y el puerto que será entrada, que será el puerto B. Junto la declaración de puertos se va a configurar una interrupciones, en las cuales se va a configurar los registros INTCON el registro OPTION\_REG y el IOCB

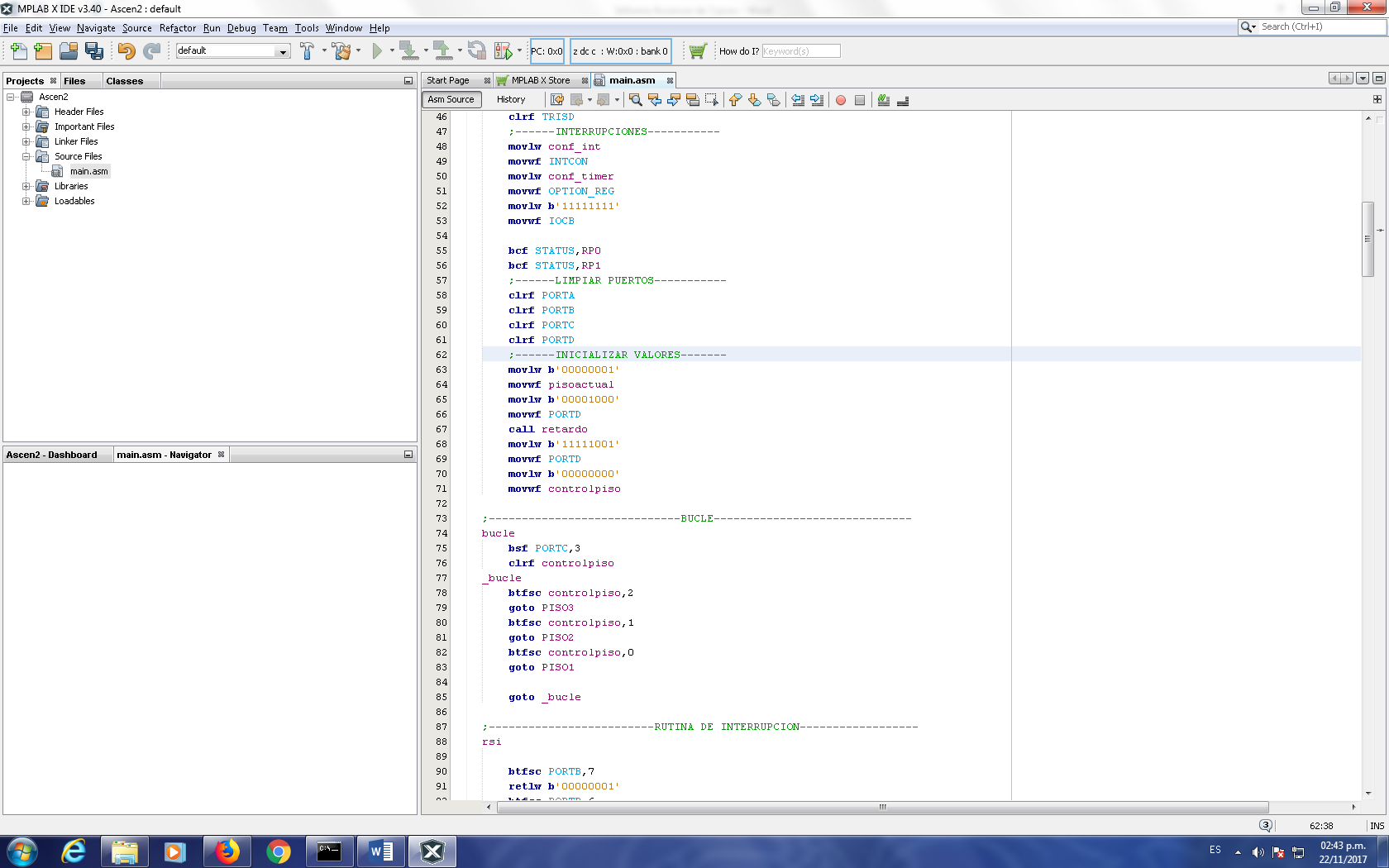


Después de declarar los puertos se regresa al banco 0.

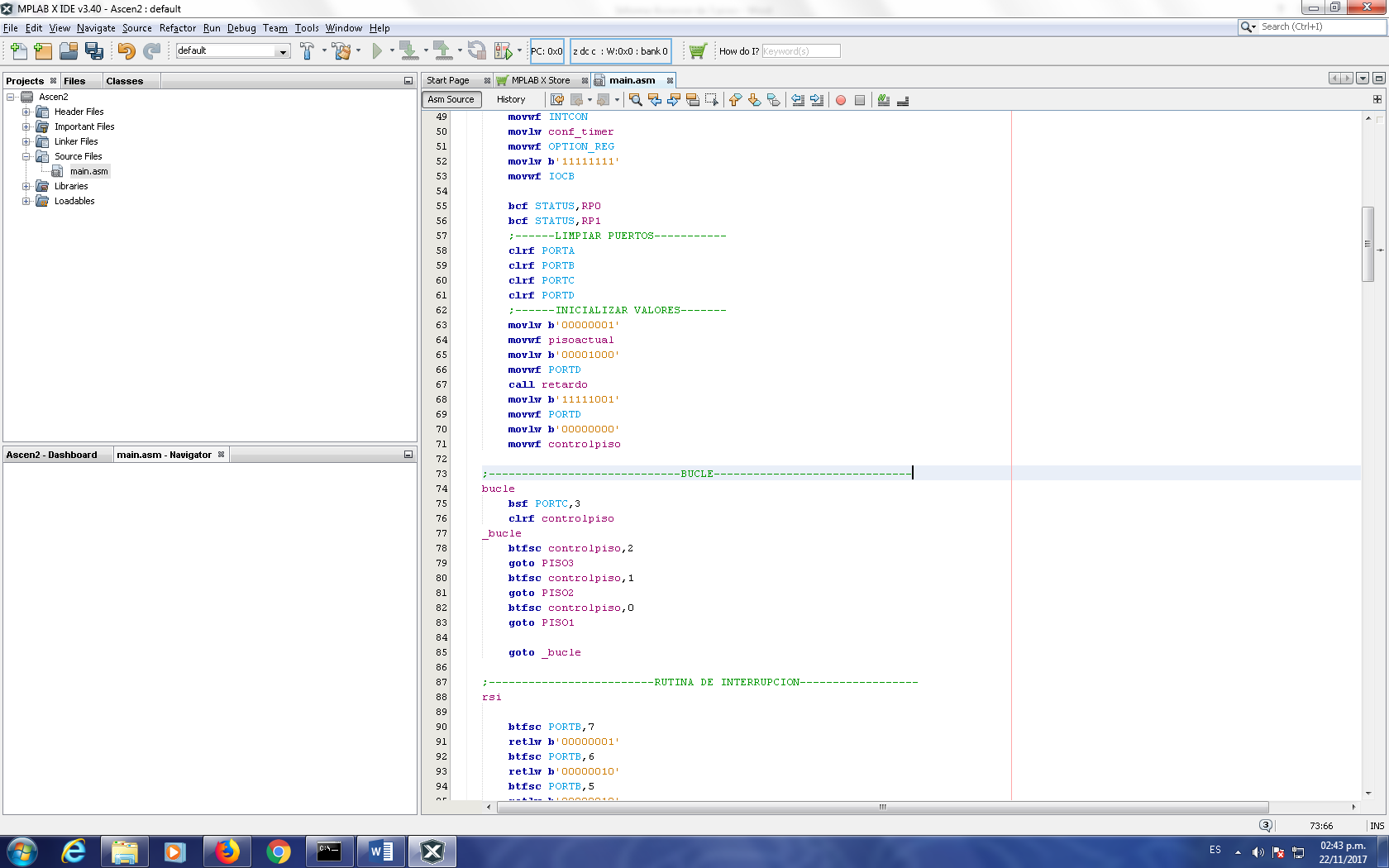
Org 0x04, es el vector de interrupción por defecto en los pic’s diseñados por microchip.

IOCB: Con el propósito de habilitar que los pulsadores causen una interrupción, el bit apropiado del registro IOCB debe estar a uno. Gracias a estas características, los pines del puerto PORTB se utilizan con frecuencia para comprobar los botones de presión en el teclado ya que detectan cada apretón de botón infaliblemente.

A continuación se limpia los puertos declarados.



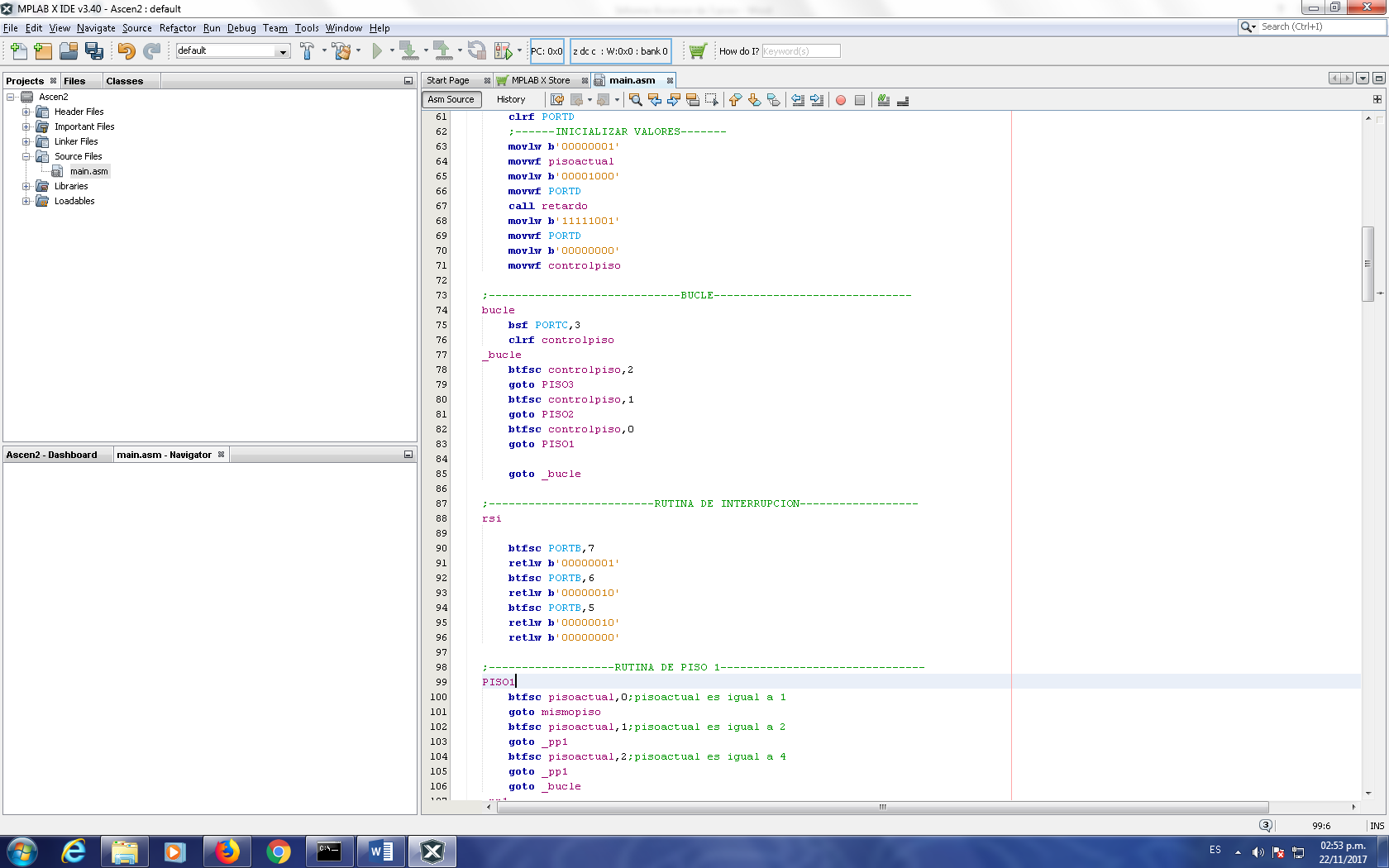
Se inicializan los valores



En las líneas donde dice movlw b’00000001’

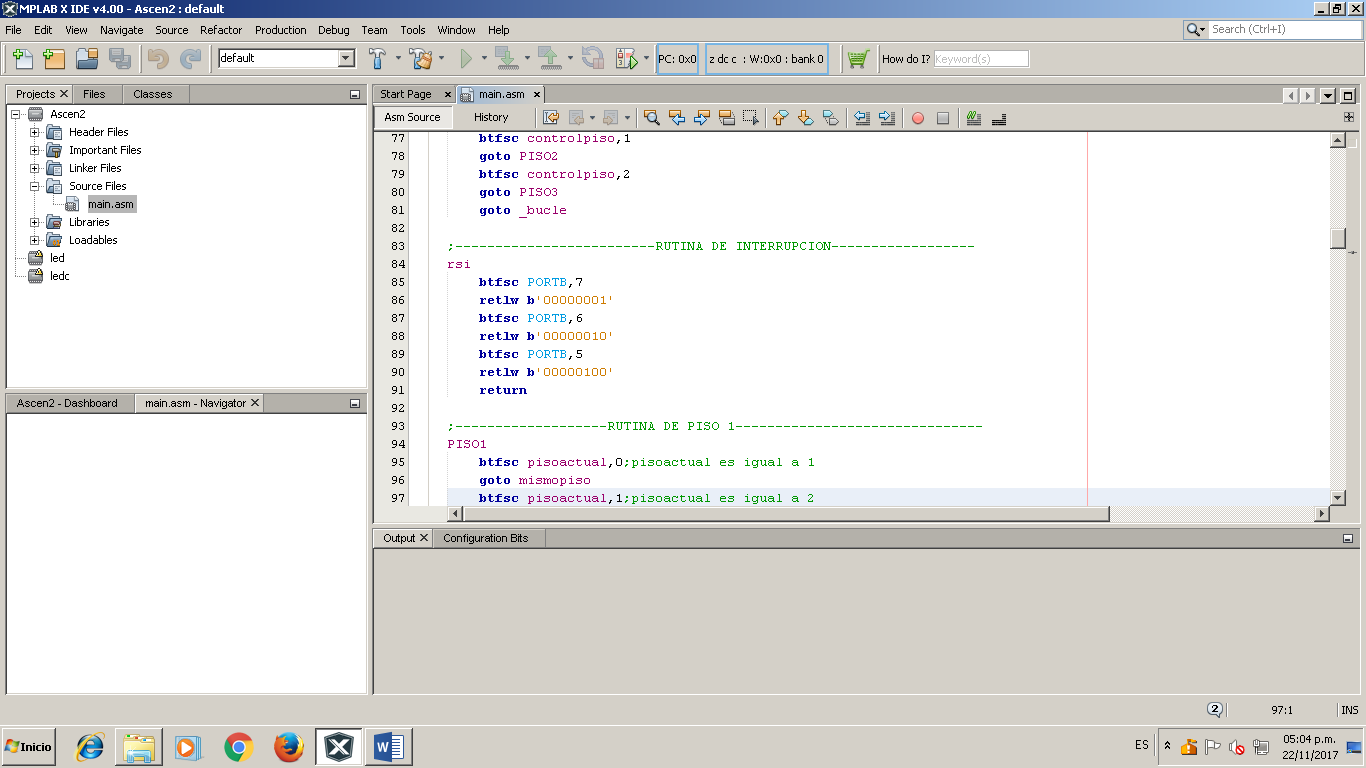
movwf pisoactual, es para que por defecto el ascensor inicie en el piso 1.

Ahora se va a hacer el bucle, el cual es la repetición o las veces que se va a ejecutar el programa, en otras palabras, el bucle infinito

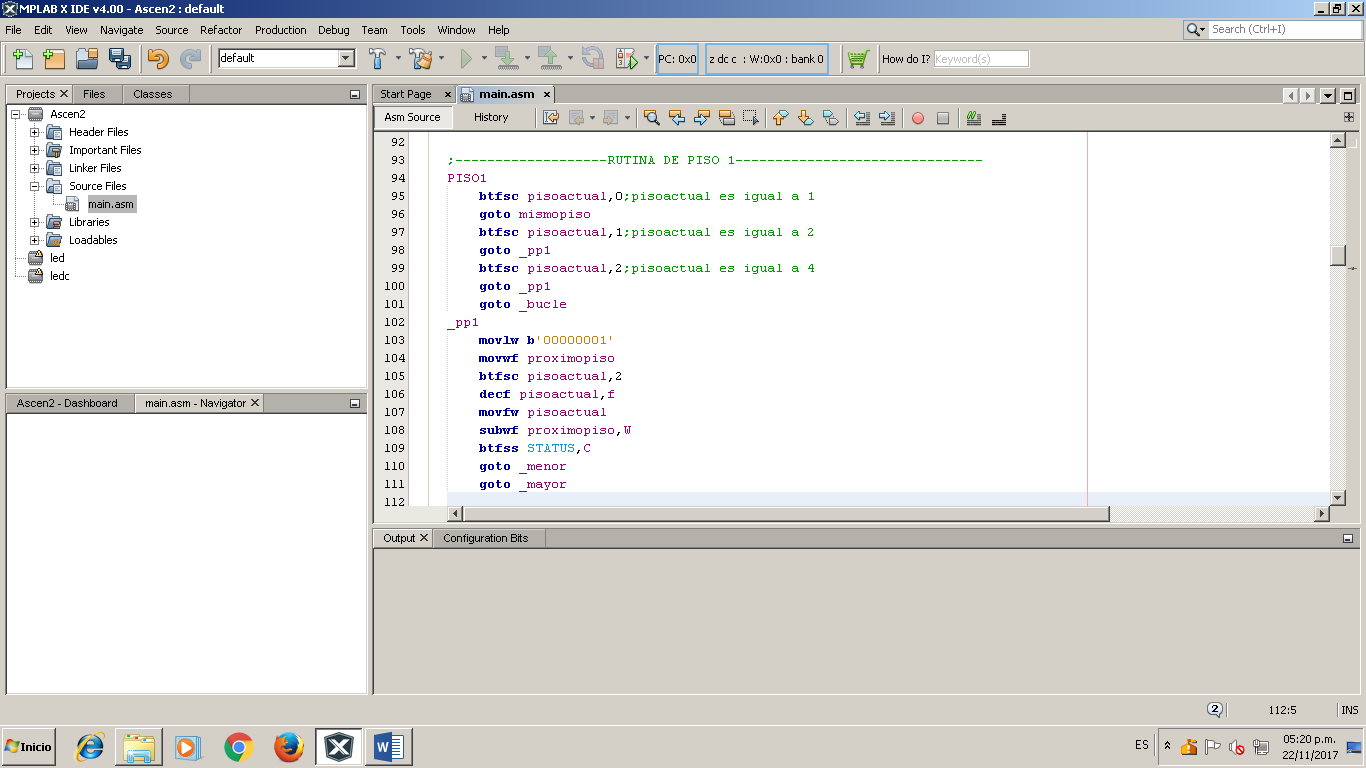


En este bucle se va a mandar los pulsos mediante los pulsadores a cual piso se quiere o se necesita ir, es decir que mediante los pulsadores se va a escoger a cuál de los 2 pisos se necesita subir o bajar. Mediante la bifurcación btfsc se lee cual piso es el presionado para que suba o baje.

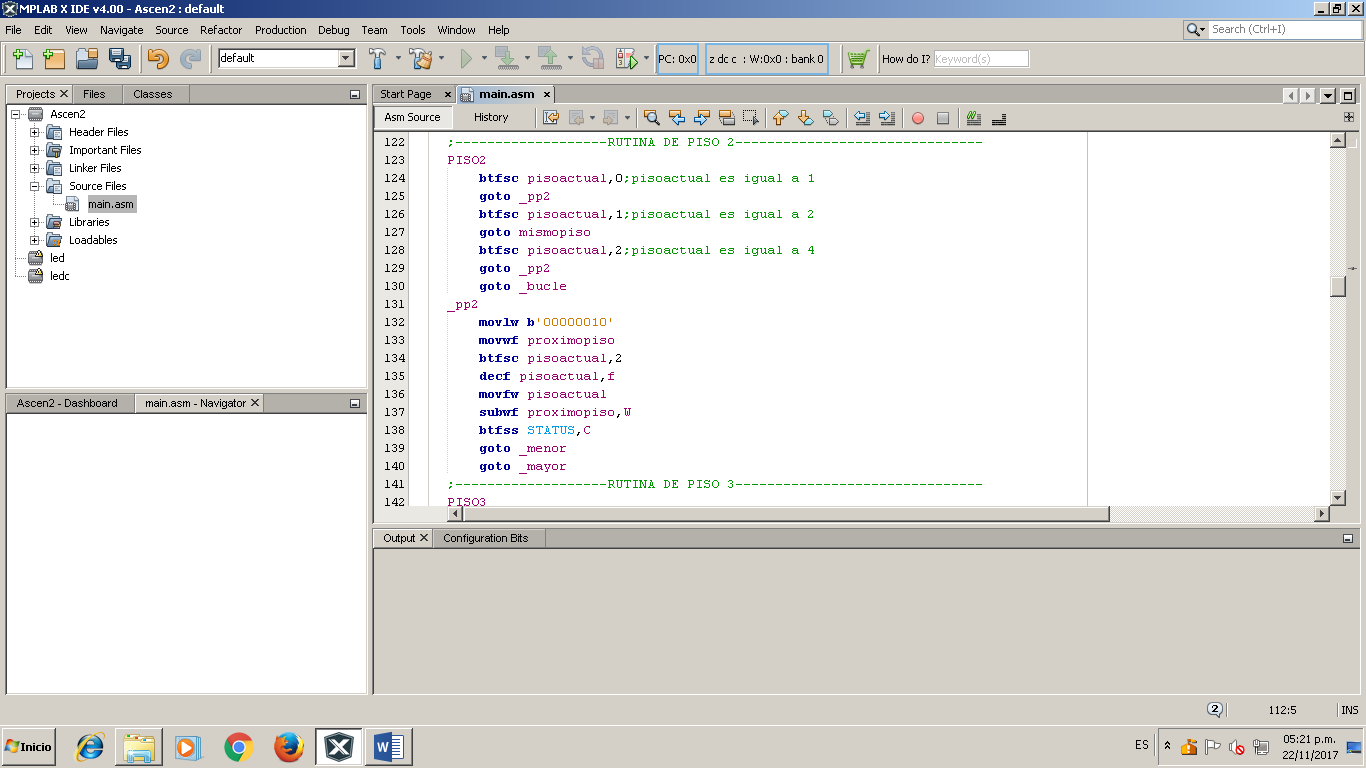
Después vamos a la rutina de interrupción, en la cual se le va a asignar un valor a cada piso, es decir al piso 1, se le asigna el valor 1, al piso 2 se le asigna el valor 2 y al piso 3 se le asigna el valor 4, en este último no se le asigna el valor de 3, debido a que se necesita un solo bit, en otras palabras el 3 en sus últimos 3 bits es de 011, mientras que el de 4 es de 100.



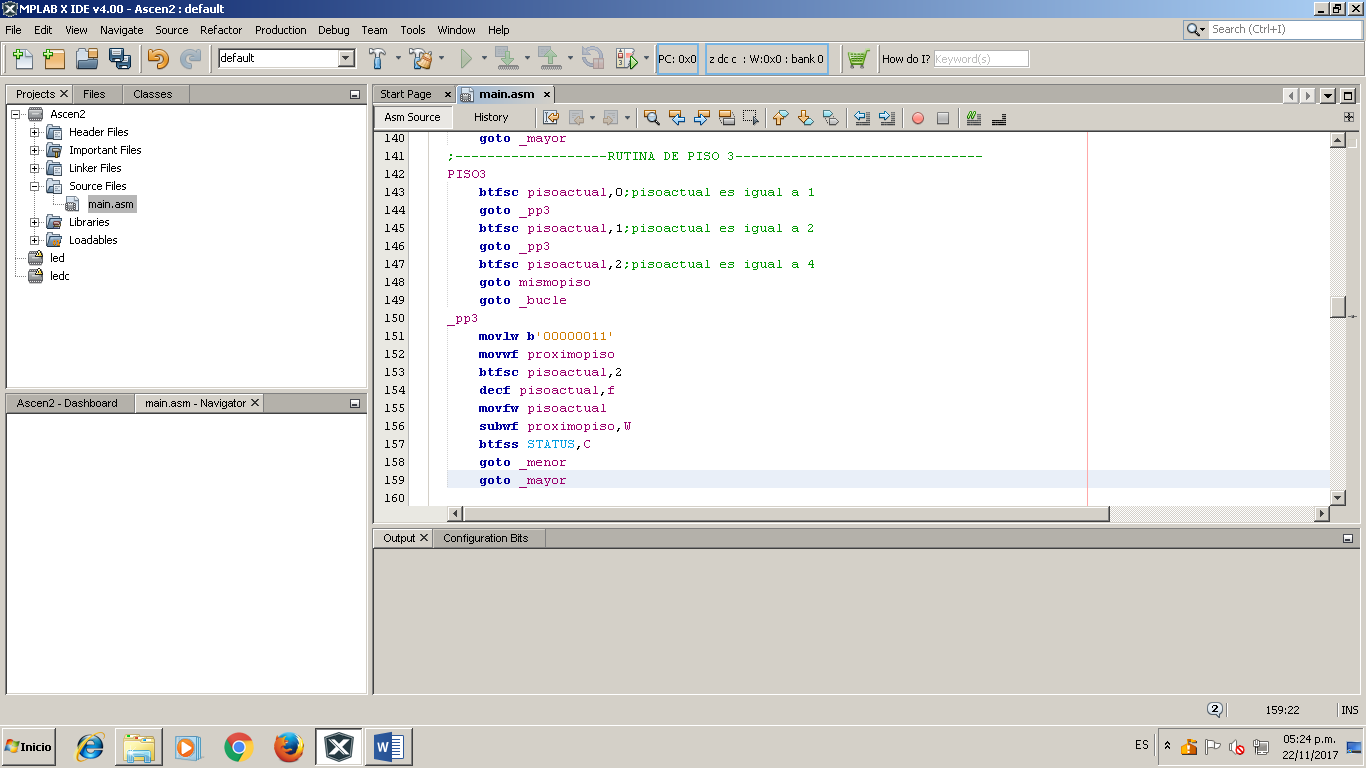
Se empieza con la subrutina del primer piso, en la cual se lee la orden para que el ascensor suba 1 o 2 piso.



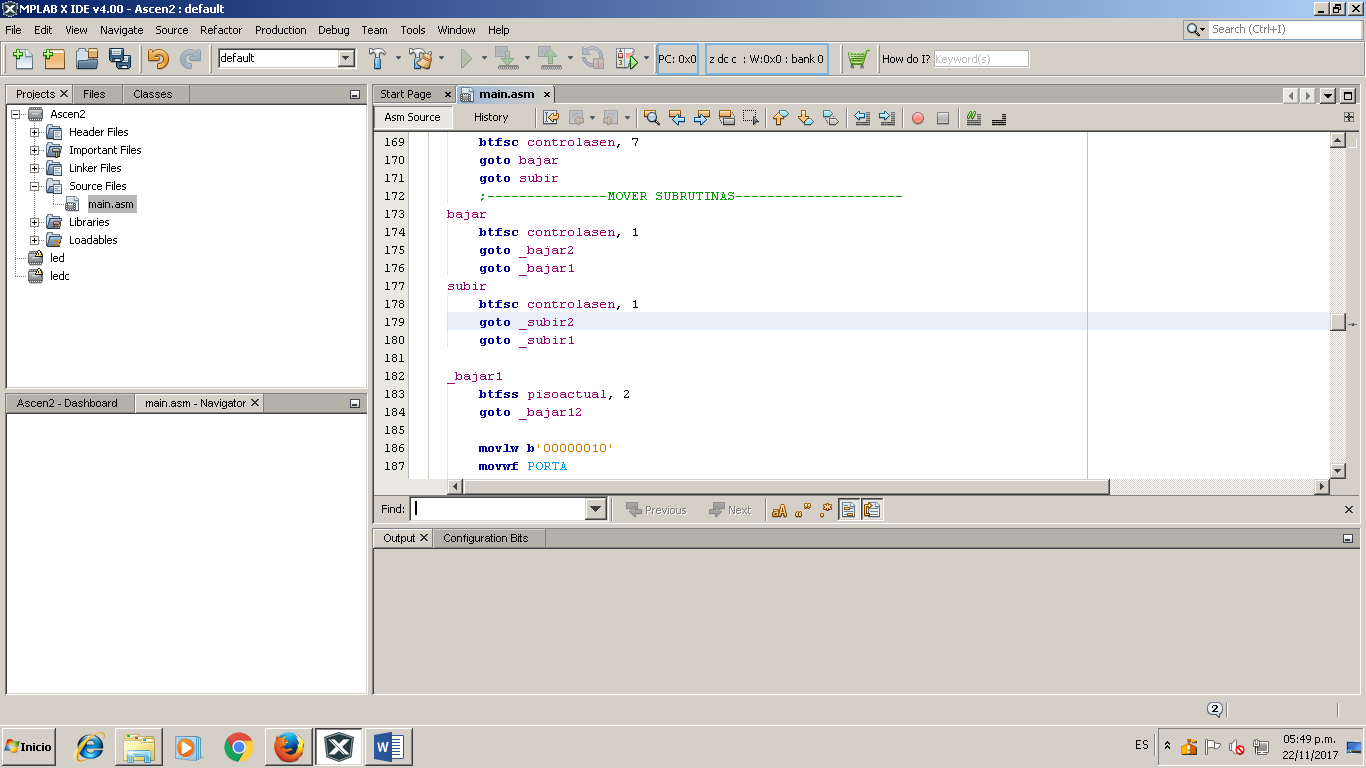
En la subrutina del segundo piso, se debe elegir cual es la orden del usuario, ya que aquí también hay 2 posibilidades, pero a diferencia de la subrutina del primer piso. Cuando el ascensor está en el segundo piso, aquí está la posibilidad de subir y la de bajar un piso.



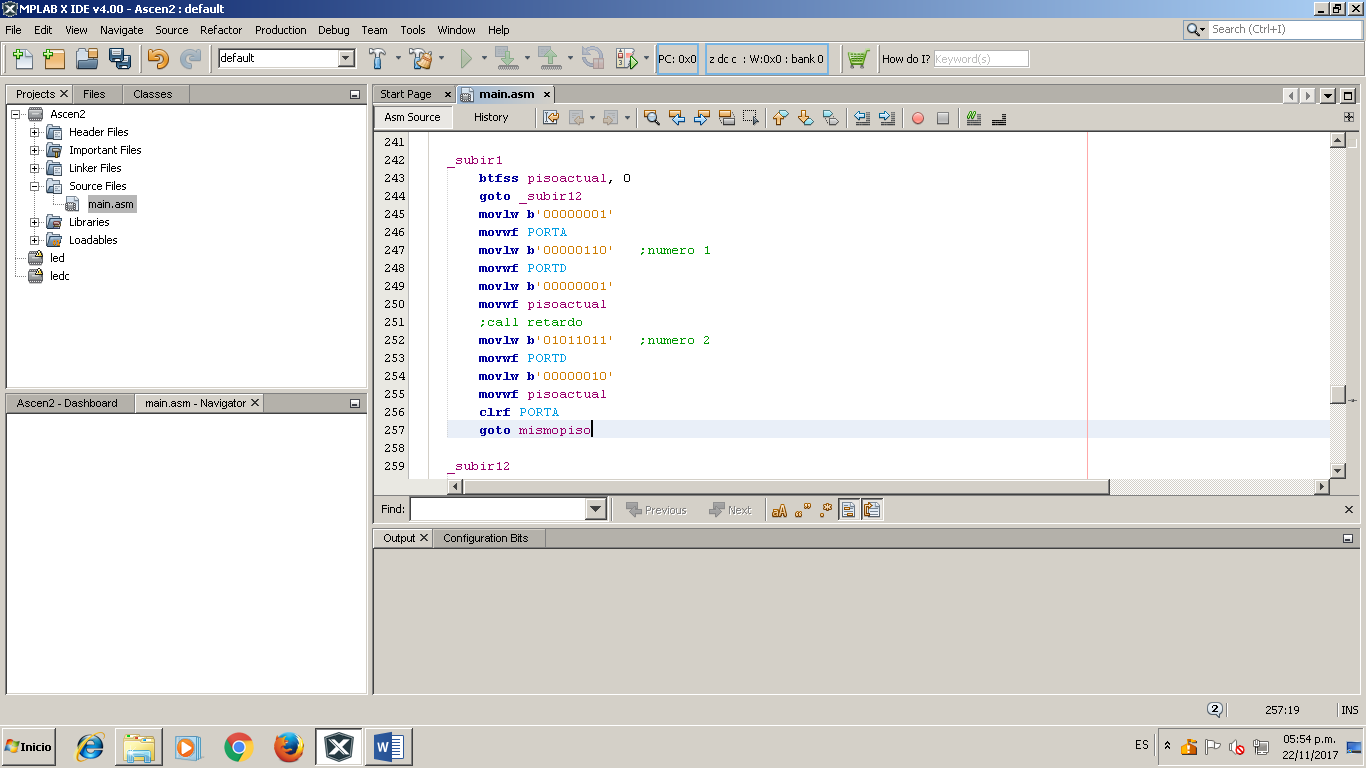
Finalmente cuando el ascensor se encuentra en el tercer piso, están las posibilidades de que el ascensor baje 1 o 2 pisos.



Ahora bien, dependiendo de cuál sea la orden dada al pic, el ascensor va a tener que subir o bajar como ya se ha mencionado, aquí entra a “jugar” el servomotor, y este es código para subir, que a la vez se divide en 2 subir, el de subir un piso y subir 2 pisos



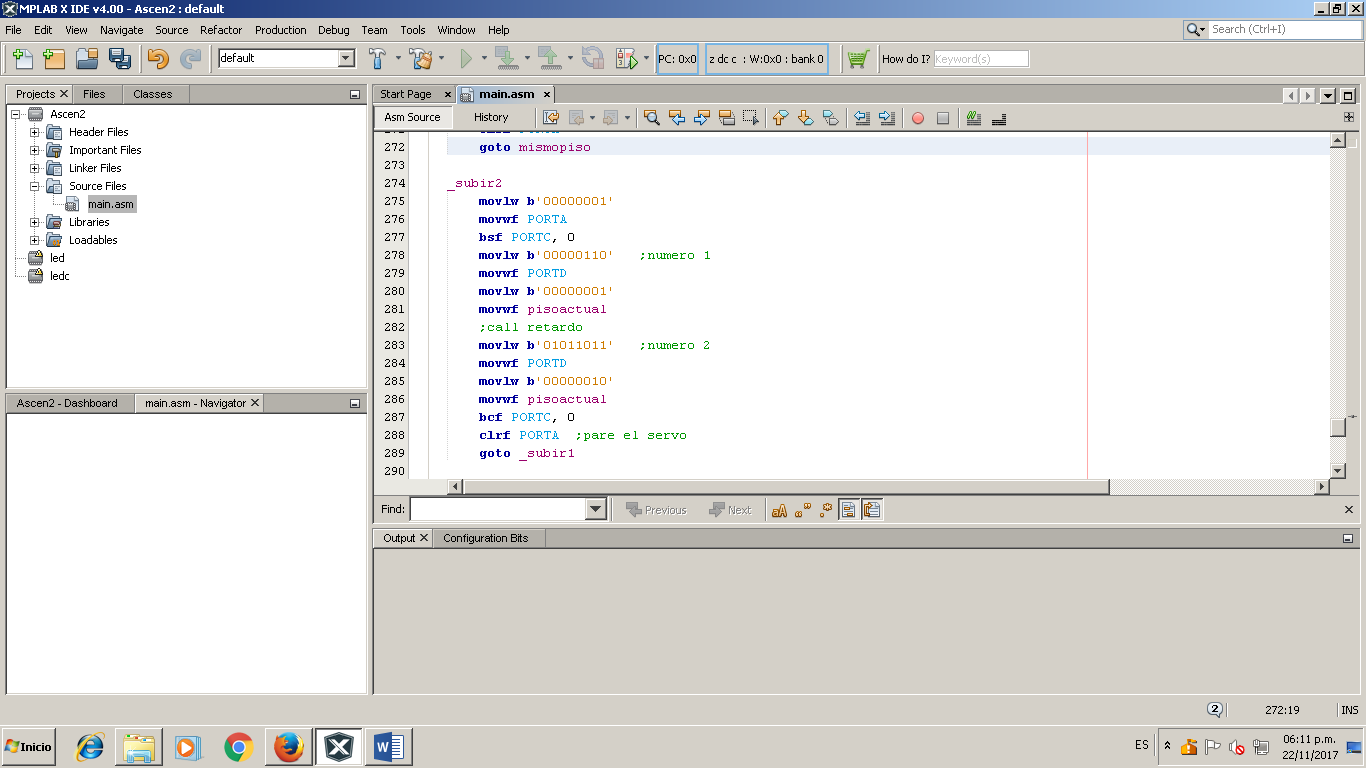
La subrutina \_subir1 es usada para subir del primer al segundo piso, esta es la subrutina.



La subrutina \_subir12 es para poder subir del segundo al tercer piso, esta es la subrutina.



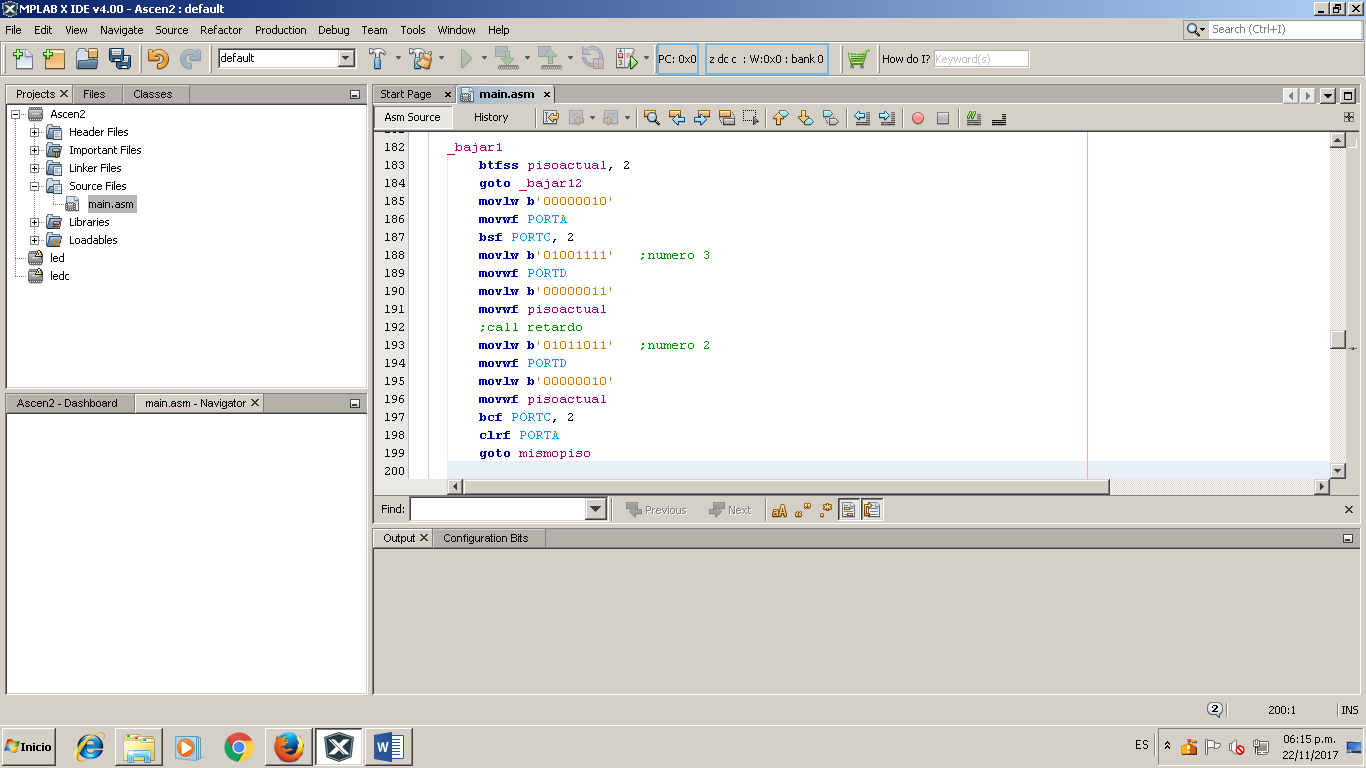
Ahora, la subrutina \_subir2 es utilizar una nueva rutina con la anterior para ir del primer piso, al segundo y después llegar al tercer piso, aquí están las subrutinas.



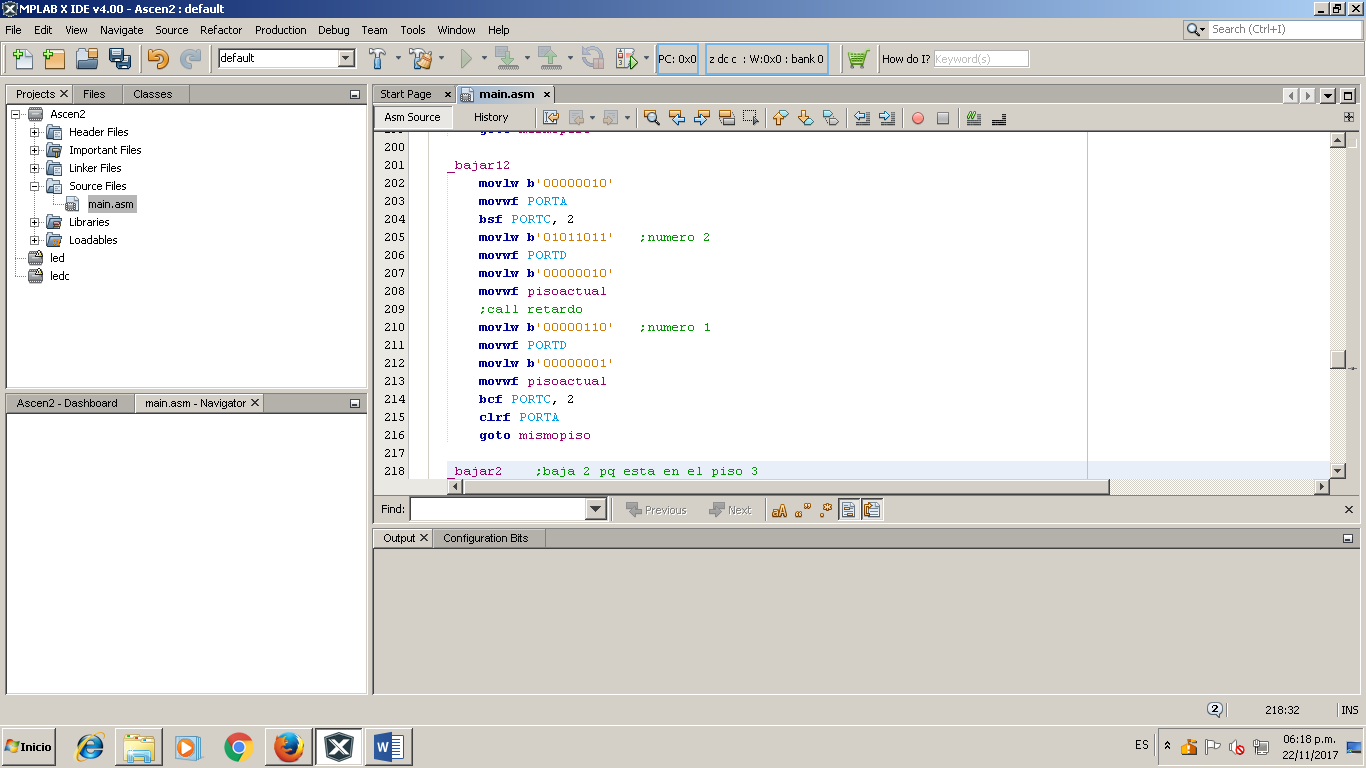


Ahora se va a mostrar el código para la bajada, al igual que el de subir tiene 3 opciones, bajar 1 que se subdivide 2 veces (en \_bajar1 y \_bajar12) y bajar 2.

Esta es la subrutina para bajar del tercer al segundo piso.

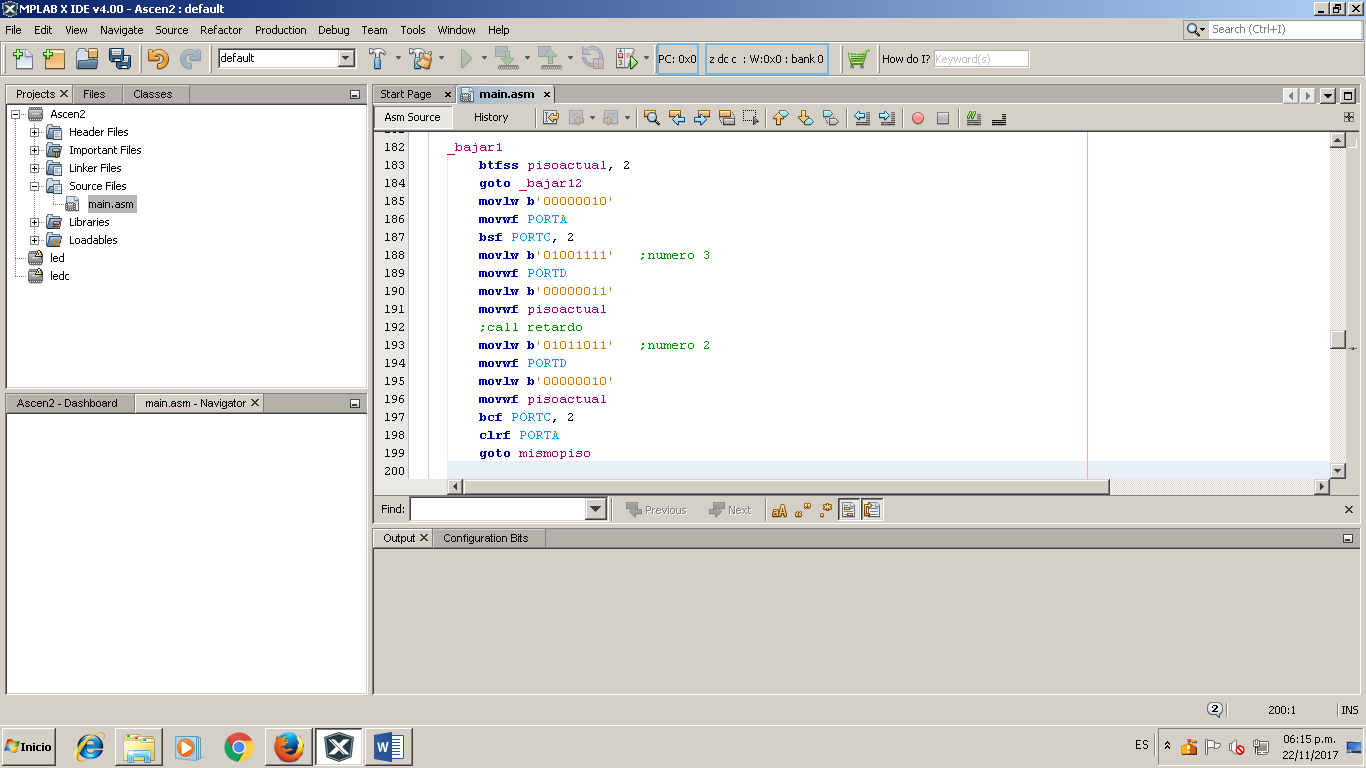


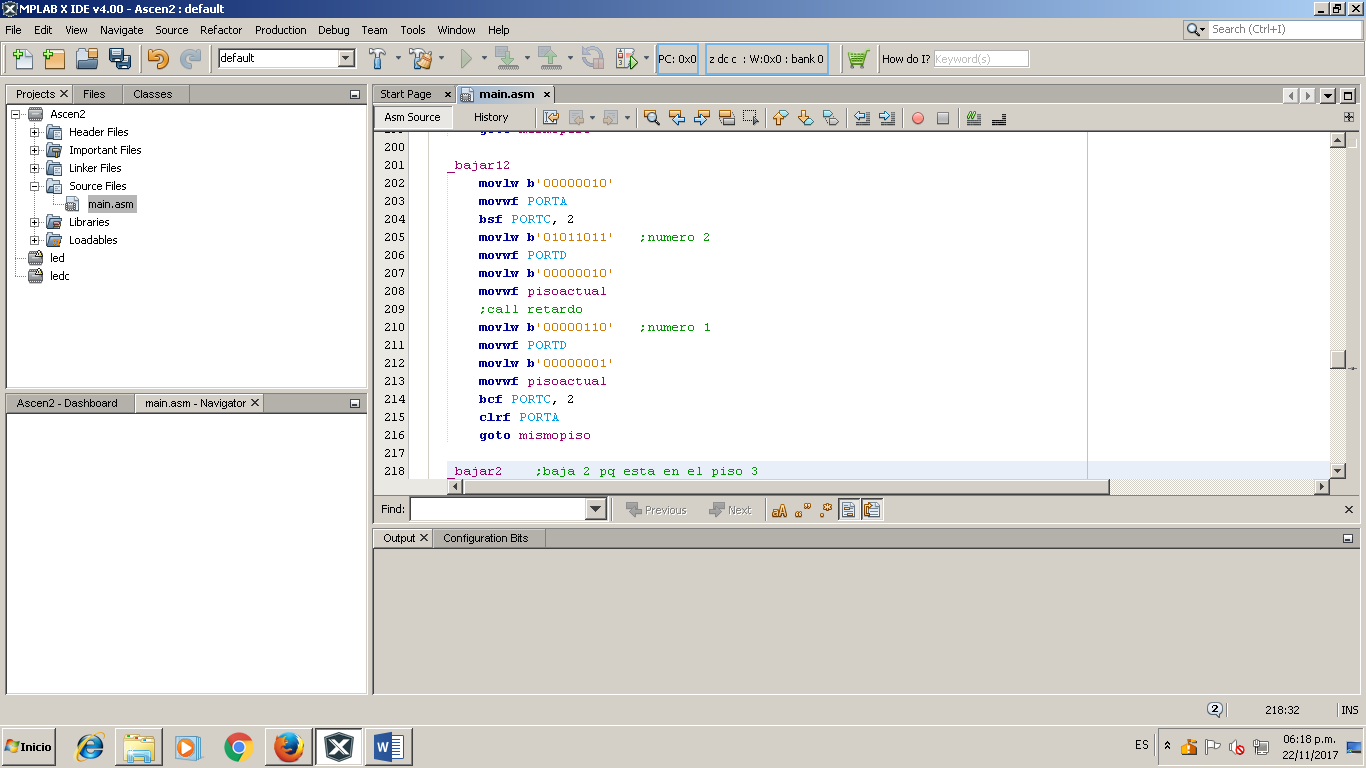
A continuación se muestra la subrutina que baja del segundo al primer piso



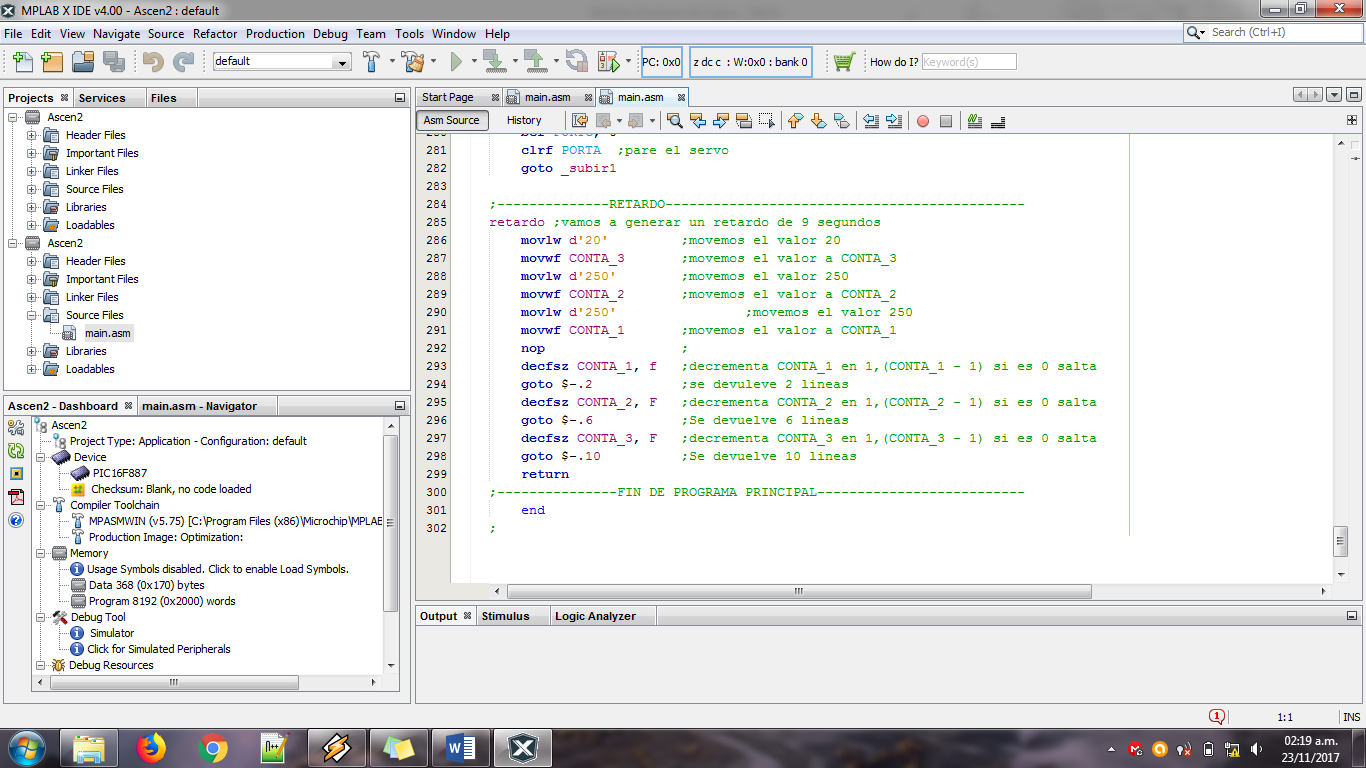
Finalmente se muestra la subrutina de bajar 2 pisos.

Que es la unión de las 2 subrutinas, ya que como están implementadas, lo único que se hace es reutilizar código, este es el proceso para bajar 2 pisos.



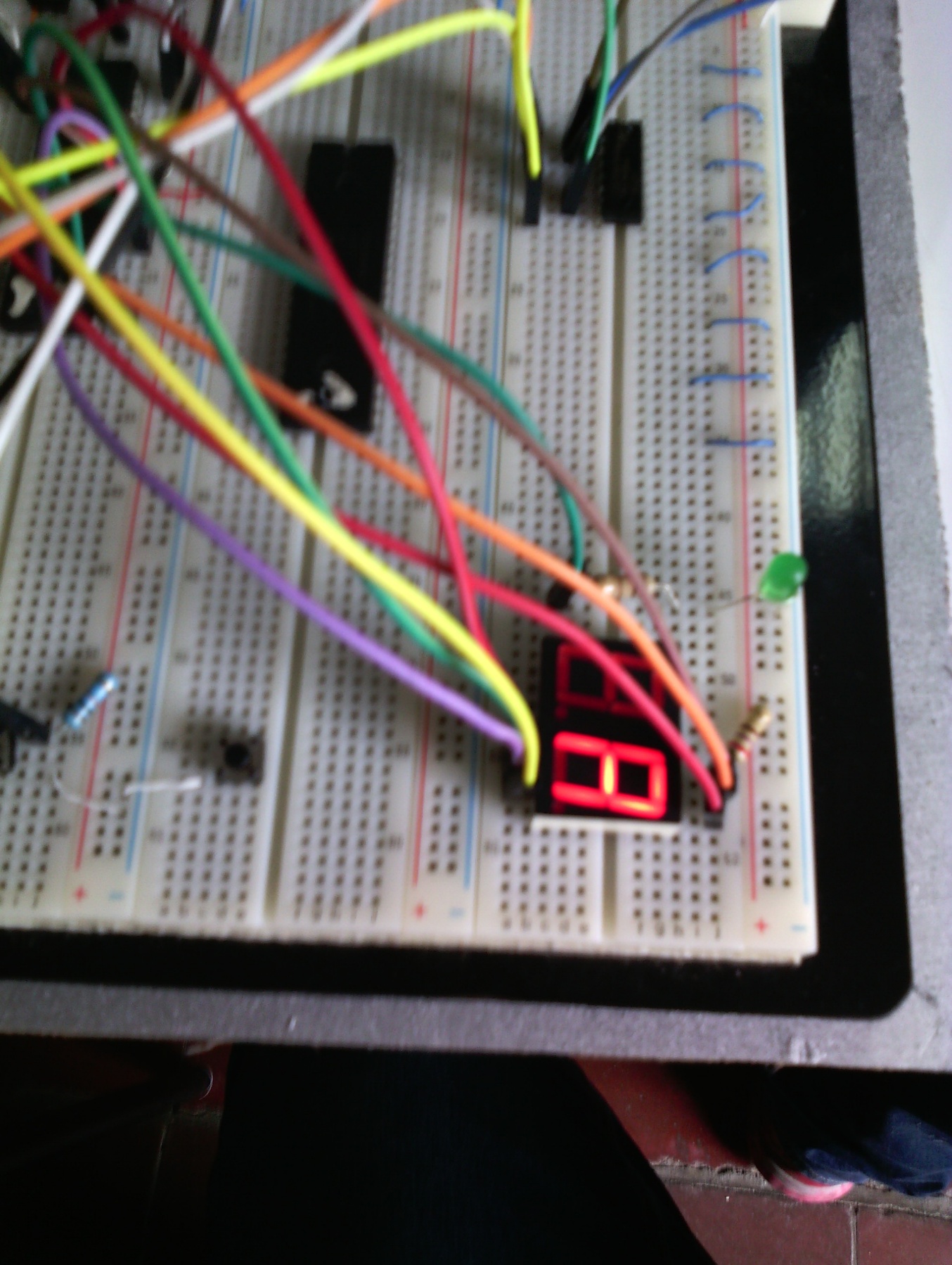


Para finalizar el código del ascensor se muestra la subrutina de retardo, la cual se va a utilizar para hacer que el servomotor gire subida y de bajada



1. Fotos de la maqueta

Inicialmente se muestra la letra en el ascensor:



Después de 3 segundos se muestra el número 1, el cual es el piso de “defecto”, el cual es el que inicia el ascensor:

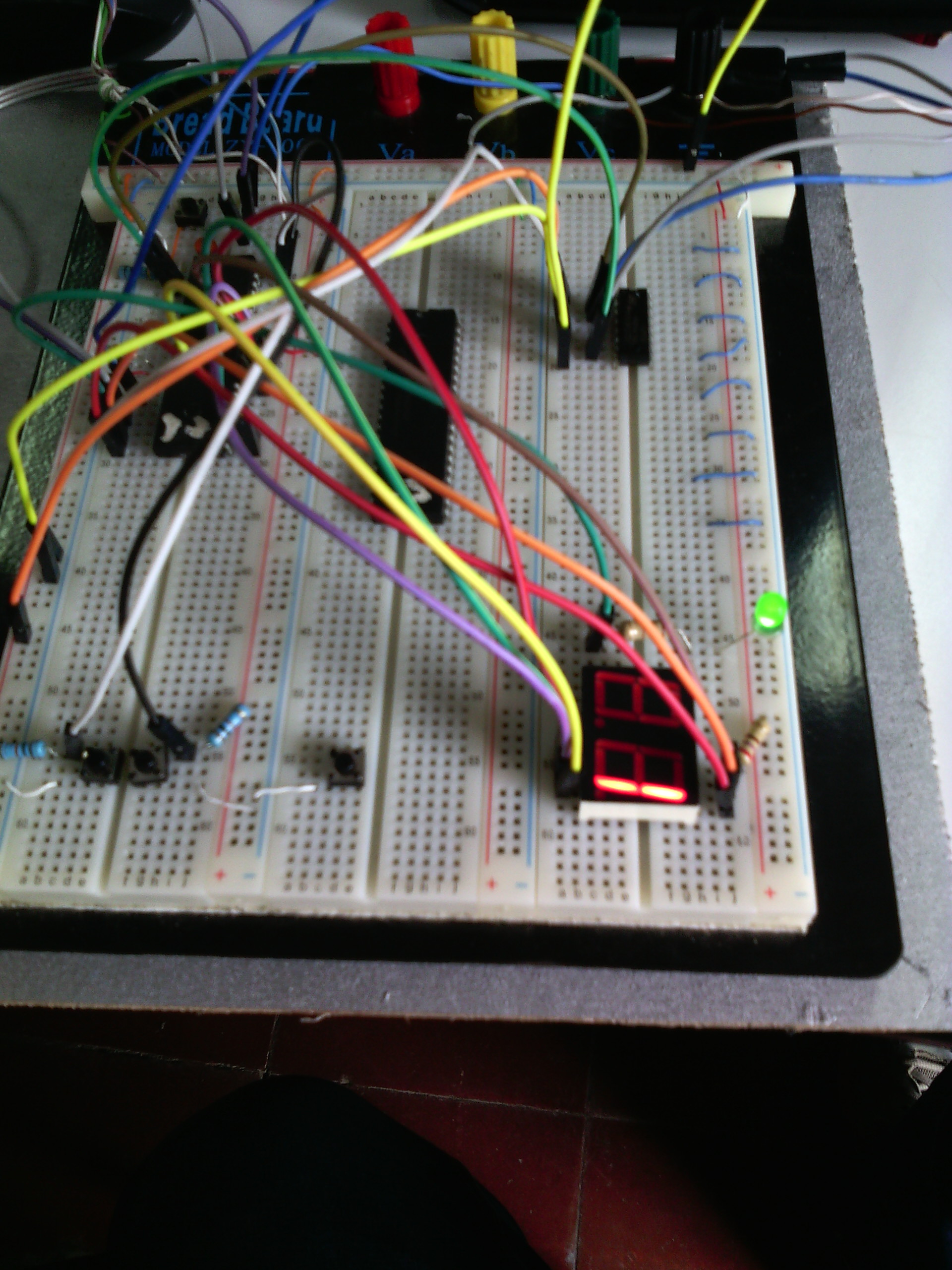
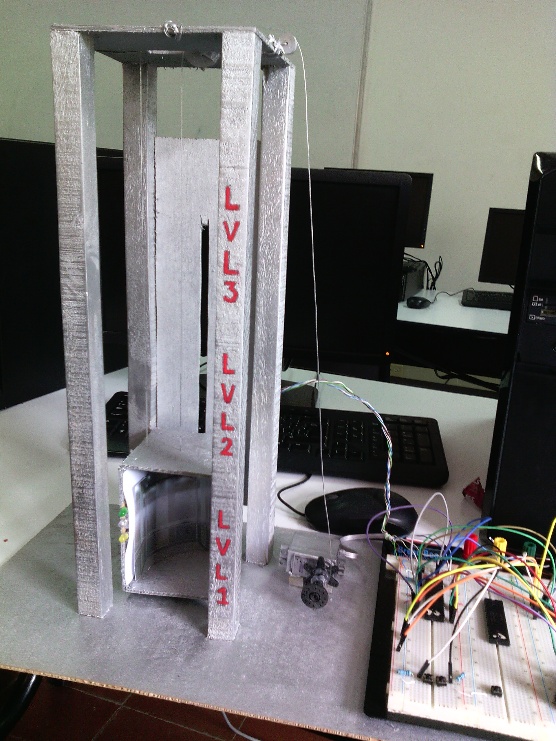


Foto de la maqueta, y la protoboard con el montaje



Led encendido cuando está subiendo:



Led enciende, indicando que se abren las puertas, al llegar al piso deseado por el usuario.



Led encendido, indicando que el ascensor está bajando:



1. Conclusiones:

* El lenguaje ensamblador es de bajo nivel, pero gracias a ello se puede aprovechar para recorrer a fondo el micro controlador a diferencia de otros lenguajes.
* Debido a la falta de un PICkit (quemador de PIC’s) para poder ver el programa implementado en la protoboard, es muy necesario una herramienta que nos simule de una manera muy semejante a la realidad, en este caso PROTEUS, ya que en este simulador se puede manejar leds, servomotores, displays (ánodo o cátodo común), el PIC16F887, conexión en pull-up o pull-Down, las resistencias y el valor de cada una de ellas, entre otras cosas
* El lenguaje ensamblador ayuda a conocer más a fondo, el micro controlador, ya que este lenguaje maneja espacios de memoria, bancos, y demás cosas que en el lenguaje C, no se tiene en cuenta
* Es de gran ayuda que el PIC16F887 tenga un reloj interno, ya que algunas veces se presentan problemas al momento de conectar un reloj externo, como a veces pasa con el PIC16F877A.
* Cuando se llega a la implementación hay que tener en cuenta todas las conexiones y un buen manejo en cuanto a la alimentación de todos los elementos, ya que estos pueden funcionar o no, según la alimentación que se les brinde.
* Hay que tener en cuenta los tiempos que se manejan en el retardo del pulso que se manda al servomotor, ya que en la implementación de este proyecto pueden resultar diferentes medidas entre cada piso.

1. Anexo

Servomotor: “Un servomotor es un tipo especial de motor que permite controlar la posición del eje en un momento dado. Está diseñado para moverse determinada cantidad de grados para ubicarse en cualquier posición dentro de su rango de operación y luego mantenerse fijo en una posición.”

Puente H: es un circuito electrónico que generalmente se usa para permitir a un [motor eléctrico DC](https://es.wikipedia.org/wiki/Motor_de_corriente_continua) girar en ambos sentidos, avance y retroceso. Otra aplicación del puente H es que puede usarse para frenarlo de manera brusca, al hacer un corto entre los bornes (partes metálicas) del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia.

STATUS: contiene: el estado aritmético de datos en el registro W, el estado RESET y los bits para seleccionar el banco para los datos de la memoria.

OPTION\_REG: Contiene varios bits de control para configurar el divisor de frecuencia o preescaler del TMR0/WDT, la interrupción externa INT, TMR0 y los pull-up para el PORTB. En este registro están los siguientes bits:

* RBPU
* INTEGD
* TOCS
* TOSE
* PSA
* PS2
* PS1
* PS0

1. Bibliografía:

http://ww1.microchip.com/downloads/en/DeviceDoc/41291D.pdf

http://panamahitek.com/que-es-y-como-funciona-un-servomotor/

https://es.wikipedia.org/wiki/Servomotor

http://www.electrontools.com/Home/WP/2016/03/09/como-funciona-el-puente-h-l293b/

http://www.circuitoselectronicos.org/2011/03/manejo-de-interrupciones-en-los.html

http://picfernalia.blogspot.com.co/2012/06/interrupciones-conceptos-basicos.html

http://electronica-teoriaypractica.com/retardos-en-los-pic/

https://learn.mikroe.com/ebooks/microcontroladorespicc/chapter/principales-registros-sfr/