

## Licencia:

(Creative Commons)

Esta obra está bajo una licencia Reconocimiento-No comercial-Sin obras derivadas 2.5 España de Creative Commons. Puede copiarlo, distribuirlo y transmitirlo públicamente siempre que cite al autor y la obra, no se haga un uso comercial y no se hagan copias derivadas. La licencia completa se puede consultar en:

<http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>

# UNIVERSIDAD OBERTA DE CATALUNYA

## Ingeniería Técnica Telecomunicaciones Telemática

Diseño de un sistema de evaluación de la integridad estructural

Alumno: Daniel Hoyo Zambrana

Dirigido por: Asier Ibeas Hernández

CURSO 2010-11 (Febrero/Septiembre)

## Agradecimientos.

A mi mujer Marta por su comprensión y ayuda prestada durante estos cuatro años, ha sido un largo camino, gracias por haber estado a mi lado.

Dedicado a mi hija Anna y a la pequeña Alba, mi futura hija.

Gracias Asier, por tu interés y soporte en este TFC.

*"Vivimos en el mundo cuando amamos. Sólo una vida vivida para los demás merece la pena ser vivida"*

**Albert Einstein**

## Resumen.

En este Trabajo de Final de Carrera (TFC), se desarrolla una posible solución para crear un dispositivo, que evalúa la integridad de una estructura.

Primero se exponen una serie de conceptos y cálculos para ir entrando en materia y comprender cómo se evalúa la integridad de una viga.

En segundo lugar se propone un diseño electrónico, que es capaz de medir la magnitud física de la deformación de la viga y convertir esta magnitud en un valor de tensión con el que poder trabajar.

Acto seguido, se diseña un circuito electrónico basado en un microcontrolador, el cual realiza las tareas de vigilancia de la viga, detectando si la estructura está en peligro o no lo está.

Se ha desarrollado un dispositivo polivalente que es capaz de realizar su misión en emplazamientos diferente, para poder realizar esta función, es necesario que el dispositivo se configure según las características del emplazamiento. Por esta razón se han diseñado 2 circuitos para poder establecer una comunicación con el dispositivo y configurarlo, una solución basada en el protocolo RS232 y otra basada en una comunicación por *Bluetooth*.

Una vez diseñada la parte electrónica, se diseña el *software* de control del microcontrolador en código C, el diseño del *software* se realiza bajo el entorno de programación MPLAB. Se establecen una serie de pruebas para verificar la correcta funcionalidad del software, estas pruebas estarán basadas en una tarjeta de desarrollo del fabricante Microchip.

Tras la verificación del *hardware* y *software* entramos en el diseño del circuito impreso, para este diseño se emplean componentes de montaje superficial "SMD".

Una vez están todos los aspectos técnicos resueltos se realiza una valoración económica sobre el coste de la realización de este proyecto.

# Índice

<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>5</b>
1.1 Objetivos .....	6
1.2 Tabla de tareas .....	7
1.3 Planificación .....	9
1.4 Evaluación de riesgos.....	10
1.5 Recursos necesarios. ....	11
1.6 Estructura de la memoria .....	11
<b>CAPÍTULO 2. ANÁLISIS DE ESTRUCTURAS .....</b>	<b>12</b>
<b>CAPÍTULO 3 .DISEÑO DE LOS BLOQUES ANALÓGICOS .....</b>	<b>15</b>
<b>3.1 Las galgas extensométricas. ....</b>	<b>15</b>
3.1.1 Principio de funcionamiento.....	15
<b>3.2 Mediciones en puente de Wheatstone. ....</b>	<b>17</b>
3.2.1 Configuración cuarto de puente. ....	18
3.2.2 Configuración medio de puente.....	19
3.2.3 Configuración puente completo.....	19
<b>3.3 Selección de la configuración.....</b>	<b>20</b>
3.3.1 Valores de salida del puente. ....	21
3.3.2 Tolerancias y diseño del circuito de medida.....	21
<b>3.4 Etapa amplificadora. ....</b>	<b>22</b>
3.4.1 El circuito buffer. ....	23
3.4.2 Bloque diferencial y no inversor. ....	23
3.4.3 El segundo amplificador diferencial.....	24
3.4.4 El bloque no inversor. ....	25
3.4.5 Esquema del módulo de amplificación.....	26
<b>3.5 Módulo de alarmas.....</b>	<b>26</b>
<b>3.6 La fuente de alimentación. ....</b>	<b>28</b>
<b>3.7 Simulación en Tina-ti. ....</b>	<b>29</b>
3.7.1 Simulación del puente y etapa amplificadora.....	29
3.7.2 Simulación de los estabilizadores de tensión del puente.....	33
<b>CAPÍTULO 4 .DISEÑO DEL BLOQUE DISCRIMINADOR.....</b>	<b>34</b>
<b>4.1 El microcontrolador. ....</b>	<b>34</b>
<b>4.2 Comunicaciones.....</b>	<b>34</b>
4.2.1 Comunicación cableada RS232 .....	34
4.2.2 Comunicación por Bluetooth.....	35
<b>4.3 El conversor analógico digital.....</b>	<b>35</b>

<b>CAPÍTULO 5 .PROGRAMACIÓN DE MICROCONTROLADOR.....</b>	<b>36</b>
<b>5.1 Entorno de desarrollo MPLAB. ....</b>	<b>36</b>
<b>5.2 Diagrama de flujo de la aplicación. ....</b>	<b>36</b>
5.2.1 Configuración del microcontrolador .....	37
5.2.2 Parametrización del dispositivo .....	38
5.2.3 Bucles de lectura .....	39
<b>5.3 Programación en código C.....</b>	<b>40</b>
<b>5.4 Validación del software. ....</b>	<b>40</b>
<b>CAPÍTULO 6 .DISEÑO DEL CIRCUITO IMPRESO. ....</b>	<b>44</b>
<b>6.1 Listado de componentes. ....</b>	<b>45</b>
<b>CAPÍTULO 7 .VALORACIÓN ECONÓMICA.....</b>	<b>47</b>
<b>CAPÍTULO 8 .CONCLUSIONES. ....</b>	<b>48</b>
<b>BIBLIOGRAFÍA .....</b>	<b>49</b>
<b>ANEXO A.....</b>	<b>1</b>
<b>ANEXO B.....</b>	<b>1</b>

## Índice de figuras.

Figura 1.1. Diagrama de bloques de la aplicación.....	5
Figura 1.1. Diagrama de Gantt Tarea 1.....	9
Figura 1.2. Diagrama de Gantt Tarea 2.....	9
Figura 1.3. Diagrama de Gantt Tarea 3.....	10
Figura 1.4. Diagrama de Gantt Tarea 4 y Tarea 5.....	10
Figura 2.1. Viga en suspensión.....	12
Figura 2.2. Viga sometida a una fuerza.....	12
Figura 2.3. Flecha de una viga.....	13
Figura 3.1. Galga en tensión.....	15
Figura 3.2. Galga en compresión.....	16
Figura 3.3. Galga extensométrica.....	16
Figura 3.4. Puente de Wheatstone.....	17
Figura 3.5. Configuración cuarto de puente.....	18
Figura 3.6. Configuración medio puente.....	19
Figura 3.7. Configuración puente completo.....	20
Figura 3.8. Disposición de las galgas sobre el plano X.....	20
Figura 3.9. Disposición de las galgas sobre el plano Y.....	20
Figura 3.10. Configuración regulador TL431.....	21
Figura 3.11. Esquema módulo de medición.....	22
Figura 3.12. Bloques etapa amplificadora.....	23
Figura 3.13. Estructura INA 146.....	23
Figura 3.14. Configuración no inversor.....	25
Figura 3.15. Esquema del módulo de amplificación.....	26
Figura 3.16. Zonas de trabajo.....	27
Figura 3.17. Módulo de alarmas.....	27
Figura 3.18. Esquema final de la fuente de alimentación.....	28
Figura 3.19. Circuito de simulación en Tina-ti.....	29
Figura 3.20. Resultados de la primera fase, medición de la offset.....	30
Figura 3.21. Resultados de la segunda fase (caso $\Delta L=1\text{mm}$ ).....	30
Figura 3.22. Resultados de la segunda fase (caso $\Delta L=1\text{mm}$ ).....	31
Figura 3.23. Resultados de la tercera fase (caso $\Delta L=50\text{mm}$ ).....	31
Figura 3.24. Resultados de la tercera fase (caso $\Delta L=50\text{mm}$ ).....	32
Figura 3.25. Diagrama de bode de la etapa amplificadora.....	33
Figura 3.26. Circuito de simulación en Tina-ti.....	33
Figura 3.27. Resultados de la simulación.....	33
Figura 4.1. Módulo comunicaciones Bluetooth.....	35
Figura 5.1. Entorno de desarrollo MPLAB.....	36
Figura 5.2. Diagrama de flujo de la configuración del microcontrolador.....	37
Figura 5.3. Diagrama de flujo para la parametrización del dispositivo.....	38
Figura 5.4. Diagrama de flujo para el bucle de funcionamiento.....	39
Figura 5.5. Librerías C18 y configuración del oscilador.....	40
Figura 5.6. Configuración de la Uart y el convertor analógico - digital.....	40
Figura 5.7. Montaje para la verificación del software.....	40
Figura 5.8. Parametrización del dispositivo mediante Hyperterminal.....	41
Figura 5.9. Variables de la aplicación.....	41
Figura 5.10. Variables de la aplicación.....	42
Figura 5.11. Variables de la aplicación.....	43
Figura 5.12. Mensajes de alarma.....	43

Figura 6.1. Layout del circuito final.....	44
Figura 6.2a. Layout de pistas superficie Top.....	44
Figura 6.2b. Layout montaje.....	44
Figura 6.3a. Layout de pistas superficie Bottom.....	45
Figura 6.3b. Layout montaje.....	45

## Índice de tablas.

Tabla 1.1. Descripción de la Tarea 1.....	7
Tabla 1.2. Descripción de la Tarea 2.....	7
Tabla 1.3. Descripción de la Tarea 3.....	8
Tabla 1.4. Descripción de la Tarea 4.....	8
Tabla 1.5. Descripción de la Tarea 5.....	8
Tabla 1.6. Riesgos y medidas.....	10
Tabla 2.1. Limitaciones de la flecha.....	13
Tabla 3.1. Rango de tensiones para las zonas de trabajo.....	26
Tabla 3.2. Necesidades de alimentación de la aplicación.....	28
Tabla 6.1. Tabla de componentes comunes.....	45
Tabla 6.2. Tabla de componentes comunicación por cable.....	47
Tabla 6.3. Tabla de componentes comunicación por BT.....	47

## Capítulo 1. Introducción

El paisaje urbanístico actual de cualquier gran ciudad nos muestra una imagen formada por edificios, puentes y todo tipo de estructuras que podamos imaginar. Gracias a la ingeniería de materiales cada día se avanza más en la obtención de materiales y aleaciones más ligeras y resistentes. Estos materiales son los elementos fundamentales de cualquier construcción que se realice hoy en día, además son la base fundamental de cualquier estructura.

Si tomamos un puente como ejemplo, éste estará construido con una gran cantidad de vigas de acero. Con el paso de los años, los agentes externos como la climatología, el agua, la salinidad del ambiente y el uso diario del puente pueden llegar a mermar las características de estas vigas.

En las construcciones modernas, cada vez está más extendida la instalación de dispositivos que nos indican si las estructuras principales están sufriendo algún tipo de deformación. Con la utilización de estos dispositivos se puede realizar un mantenimiento preventivo de la estructura en el momento que se ha detectado alguna desviación y evitando así el deterioro en cadena de la estructura.

En este Trabajo de Final de Carrera (TFC), se desarrollará un sistema capaz de detectar la deformación de una estructura que tendremos bajo observación, cuando el sistema detecte un cierto grado de deformación, éste emitirá una señal de alarma para ponernos en aviso, inicialmente esta señal de alarma será acústica.

Este sistema, está compuesto por una serie de galgas extensométricas [1] para poder medir la deformación de la viga, un circuito de amplificación de la señal de las galgas, un microcontrolador con conversor analógico digital integrado y un módulo de potencia para activar la alarma y los visualizadores ópticos.

Dado que el grado de deformación permitido puede variar según el material de la estructura, dimensiones de ésta, etc., el sistema contará con un interfaz de comunicaciones para poder configurar el dispositivo según las necesidades.

Las galgas extensométricas, instaladas en configuración puente de *Wheatstone* [2], varían su resistencia eléctrica en función de la fuerza a la que estén sometidas, esto nos dará una pequeña variación de tensión a la salida del circuito de medición. Esta señal pasará por un circuito amplificador compuesto por amplificadores operacionales [3] que amplificarán la señal para poder trabajar mejor con las variaciones de tensión, la conversión analógico digital y el procesamiento de la señal se realizará mediante un microcontrolador de la casa Microchip y en función de los parámetros que le indiquemos al microcontrolador, éste nos activará un zumbador si fuera necesario.

En la figura 1.1 se puede ver un esquema de los bloques que constituirán esta aplicación.

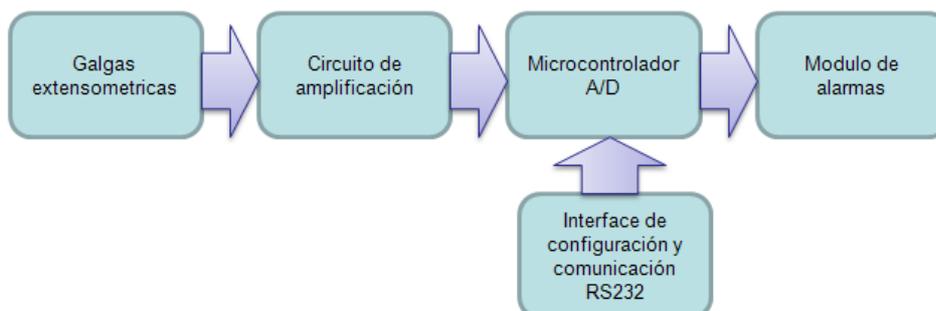


Figura 1.1. Diagrama de bloques de la aplicación

## 1.1 Objetivos

El objetivo principal de este TFC, es el diseño de un dispositivo para evaluar la integridad de una estructura, para ello es necesario que se cumplan los siguientes objetivos:

- Diseñar un circuito de medición en configuración puente de *Wheatstone*, determinar ecuaciones para conocer la salida del puente en función de la configuración seleccionada, cuarto de puente, medio puente o puente completo.
- Diseño de un circuito de amplificación de la señal del salida del puente de *Wheatstone*
- Diseñar un módulo de alarmas
- Diseñar un circuito discriminador mediante un microcontrolador
- Diseñar un circuito de comunicación entre el ordenador y el dispositivo
- Diseñar el *software* de control para la gestión del dispositivo
- Diseñar el circuito impreso
- Adquirir capacidades para la redacción de documentación técnica

## 1.2 Tabla de tareas

La tabla de tareas se ha estructurado en 5 tareas que representan las fechas de entrega de las 3 PACs, la entrega final del TFC y el debate virtual. Estas tareas se subdividen en hitos, intentando seguir una correlación lógica.

Para la reserva de recursos se ha calculado un tiempo medio por día de 2.5 horas. La descripción de tareas junto con la duración y fechas de inicio y final de estas se pueden ver en las tablas 1.1 a 1.5

**Tabla 1.1. Descripción de la Tarea 1**

Tarea	Hito	Descripción del hito	Composición del hito	Recursos	Fecha de inicio	Fecha límite	Hitos predecesores	
Tarea 1	Hito 1.1	Estudio de la embergadura	Lectura del Plan de estudios de la asignatura	1 día	02/03/2011	07/03/2011		
			lectura de la documentación adicional (Modulos)					
			Lectura del enunciado del Proyecto					
	Hito 1.2	Análisis de la situación	Analizar el enunciado del Proyecto	2 días	08/03/2011	11/03/2011		
			Borrador del Plan de trabajo - PAC1					
			Análisis de recursos necesarios					
			Fijación de objetivos					
	Hito 1.3	Asistencia Trobada Presencial	Aclarar con el tutor dudas Técnicas y Administrativas	1 día	12/03/2011	12/03/2011		1.1, 1.2

**Tabla 1.2. Descripción de la Tarea 2**

Tarea	Hito	Descripción del hito	Composición del hito	Recursos	Fecha de inicio	Fecha límite	Hitos predecesores
Tarea 2	Hito 2.1	Estudio Galgas extensiométricas	Busqueda de información galgas	8 días	07/03/2011	15/03/2011	1.2, 1.3
			Análisis de la información recopilada		15/03/2011	17/03/2011	
			Análisis de los modos de medición		15/03/2011	17/03/2011	
			Realizar correcciones de la PAC 1		20/03/2011	23/03/2011	
	Hito 2.2	Diseño puente de Wheatstone	Diseño de la configuración de los elementos de medición "galgas"	3 días	18/03/2011	19/03/2011	2.1
			Implementación en TINA -ti				
			Análisis de resultados				
			Modificaciones del diseño en función de los resultados				
			Validación y documentación (memoria técnica) del diseño		20/03/2011	22/03/2011	
	Hito 2.3	Diseño Modulo amplificación	Diseño de la etapa aplicadora "Amplificadores Operacionales"	3 días	22/03/2011	23/03/2011	
			Implementación en TINA -ti				
			Análisis de resultados				
			Modificaciones del diseño en función de los resultados				
			Validación y documentación (memoria técnica) del diseño		23/03/2011	25/03/2011	
	Hito 2.4	Simulación	Implementación en TINA -ti de la Fase 1 y la Fase 2 del diseño	3 días	27/03/2011	28/03/2011	2.2, 2.3
Análisis de resultados							
Modificaciones del diseño en función de los resultados							
Validación y documentación (memoria técnica) del diseño							
				28/03/2011	30/03/2011		
Hito 2.5	Diseño modulo de Alarmas	Diseño de la etapa de salida (Leds y zumbador) " Etapa de potencia con transistor"	1 días	01/04/2011	01/04/2011		
		Validación y documentación (memoria técnica) del diseño					
Hito 2.6	Diseño Fuente de alimentación	Diseño de electrónico de la fuente de alimentación	1 días	03/04/2011	03/04/2011		
		Validación y documentación (memoria técnica) del diseño					

**Tabla 1.3. Descripción de la Tarea 3**

Tarea	Hito	Descripción del hito	Composición del hito	Recursos	Fecha de inicio	Fecha límite	Hitos predecesores
Tarea 3	Hito 3.1	Diseño esquemático	Diseño de la unidad de control "Discriminador", Selección del Microcontrolador a utilizar según las necesidades de computación Diseño en Eagle del circuito Completo Realizar correcciones de la PAC 2	3 días	15/04/2011	18/04/2011	2.3, 2.4, 2.5, 2.6
	Hito 3.2	Programación PIC	Diseño del diagrama de flujo Implementación en MPLAB el diagrama de flujo Evaluar necesidad de modificar el hardware en función del software Validación y documentación (memoria técnica) del Software.	7 días	19/05/2011	03/05/2011	
	Hito 3.3	Diseño del PCB	Validación del Cto. En Eagle Hito 3.1 Diseño en Eagle del PCB. Validación y documentación (memoria técnica) del PCB	4 días	03/05/2011	07/05/2011	3.1

**Tabla 1.4. Descripción de la Tarea 4**

Tarea	Hito	Descripción del hito	Composición del hito	Recursos	Fecha de inicio	Fecha límite	Hitos predecesores
Tarea 4	Hito 4.1	Documentación Extra	Incluir Anexos en la memoria (datasheets, hojas de datos...) Incluir biblioteca de referencias Realizar correcciones de la PAC 3	2 días	25/05/2011	27/05/2011	
	Hito 4.2	Preparación del video de presentación	Decidir los puntos más relevantes del proyecto a incluir en el video Grabación Montaje del video	4 días	27/05/2011	02/06/2011	
	Hito 4.3	Entrega Proyecto	Repasar la memoria del proyecto Realizar correcciones necesarias Entrega del proyecto	2 días	02/06/2011 02/06/2011	04/06/2011 05/06/2011 06/06/2011	

**Tabla 1.5. Descripción de la Tarea 5**

Tarea	Hito	Descripción del hito	Composición del hito	Recursos	Fecha de inicio	Fecha límite	Hitos predecesores
Tarea 5	Hito 5.1	Debate virtual	Debatir el proyecto	2 días	28/06/2011	29/06/2011	

### 1.3 Planificación

Según la tabla de tareas descritas en el punto anterior se ha planificado su ejecución según se muestra en los siguientes diagramas de Gantt.

En esta planificación se han tenido en cuenta la carga de trabajo dedicada a otras asignaturas del semestre, compromisos personales y 2 viajes programados de carácter laboral. La planificación de tareas de este TFC, se muestra en los diagramas de Gantt de las figuras 1.2 a 1.5.



Figura 1.2. Diagrama de Gantt Tarea 1

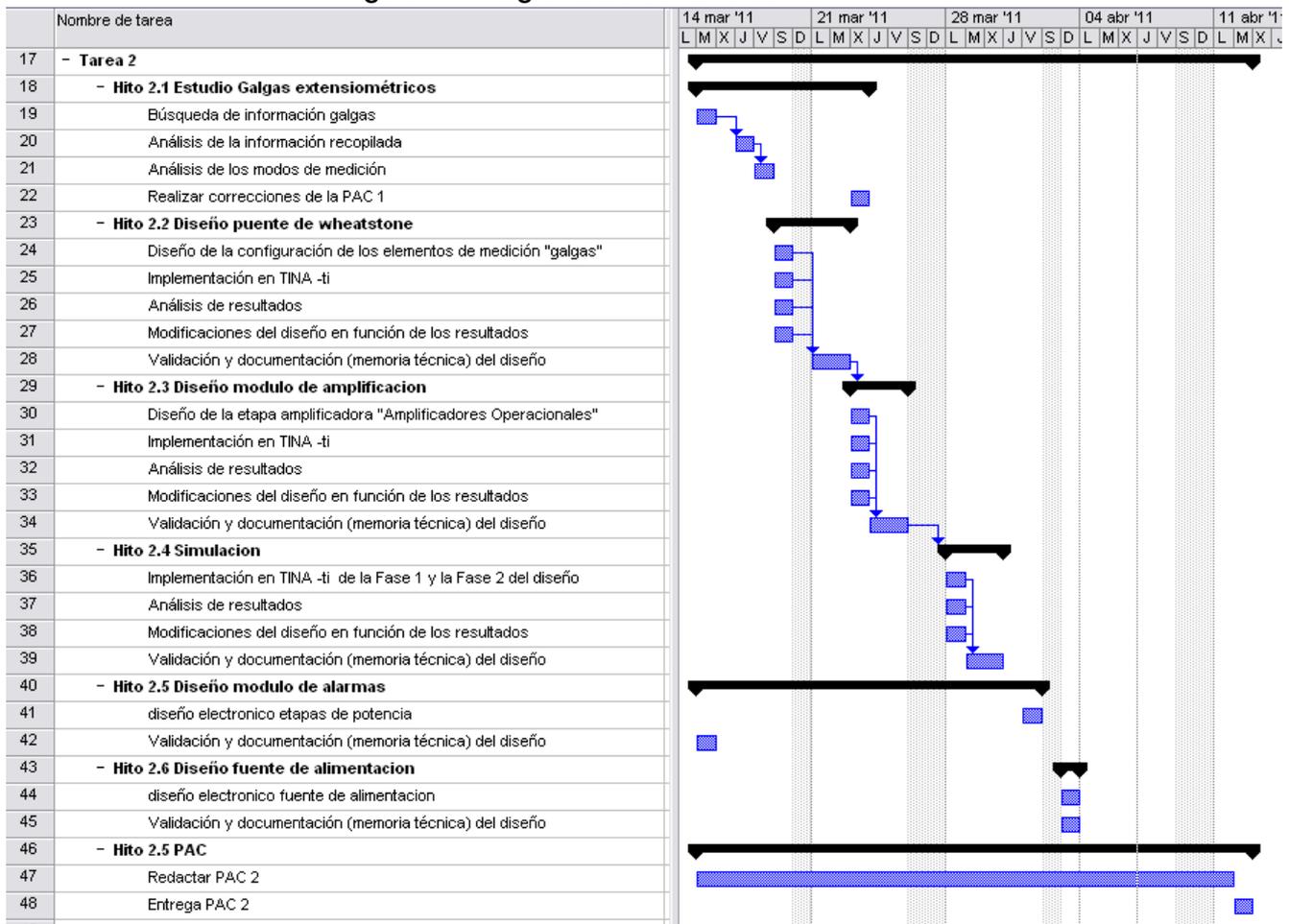


Figura 1.3. Diagrama de Gantt Tarea 2

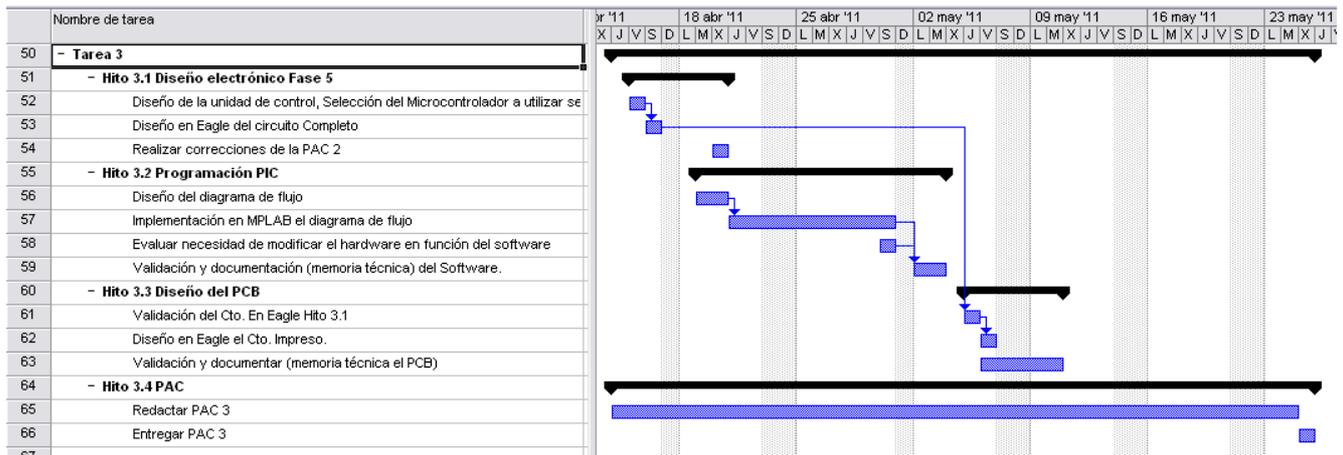


Figura 1.4. Diagrama de Gantt Tarea 3

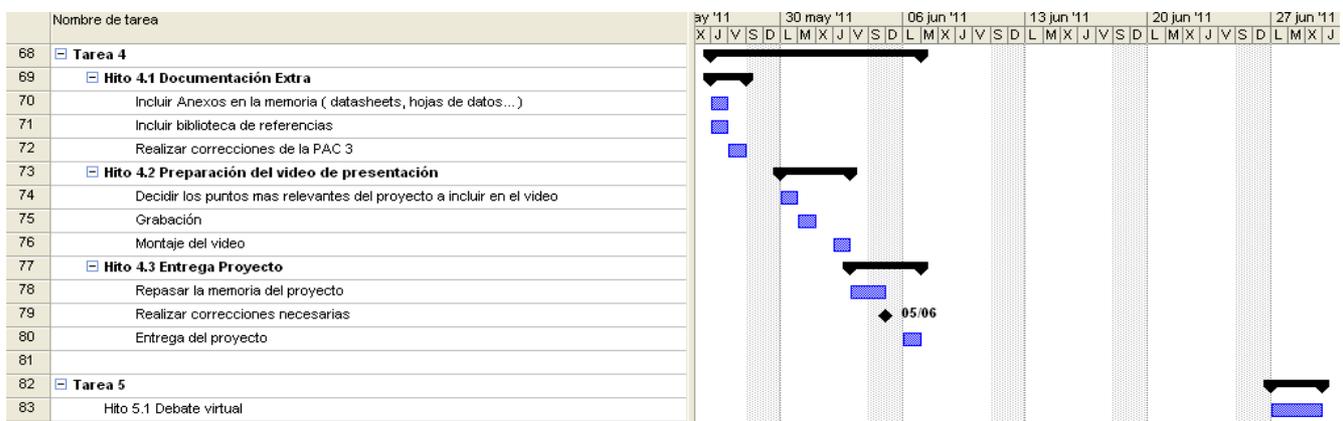


Figura 1.5. Diagrama de Gantt Tarea 4 y Tarea 5

## 1.4 Evaluación de riesgos

A continuación, en la tabla 1.6 se indican los posibles riesgos que podrían poner en peligro la planificación del PFC

Tabla 1.6. Riesgos y medidas

INCIDENCIA	MEDIDA DE CONTENCION
Necesidad de dedicar más tiempo a otras asignaturas	En la planificación de tareas, para las PAC 2 y PAC 3, se ha dejado un tiempo de guarda entre la terminación de los hitos y la entrega de 10 días, si fueran necesarios se utilizarían estas previsiones.
Viaje de trabajo no programado	
Nacimiento prematuro de mi 2ª hija, programación teórica 1ª semana de Julio 2011.	Aprovechar los días de permiso laboral y darle prioridad al TFC.
Perdida del Stick USB, con la información	Semanalmente, se realiza una copia de seguridad del Stick

## 1.5 Recursos necesarios

La relación de recursos tanto de hardware como de *software* para llevar a cabo este TFC es la siguiente:

Hardware:

- Ordenador personal (Sobremesa o portátil) con S.O. Windows XP
- Línea de datos ADSL
- Depurador ICD3 y placa de desarrollo Microchip

Software:

- Paquete de ofimática Microsoft Office, para la redacción de la memoria
- Microsoft Project, para la gestión del desarrollo del proyecto
- *Software* de simulación electrónica TINA-ti de Texas Instruments, para la simulación de los circuitos analógicos diseñados
- *Software* de programación MPLAB de Microchip y compilador C18, para la implementación del *software* de control
- *Software* diseño electrónico EAGLE de CadSoft, para el diseño del circuito impreso.
- Consola de Hyperterminal disponible en el S.O. Windows XP, este recurso es opcional, solo es necesario si se implementa físicamente el bloque discriminador, con esta consola se establecerá la comunicación con el dispositivo.

## 1.6 Estructura de la memoria

En el capítulo 2 se realiza una breve introducción al análisis de estructuras, dándose a conocer conceptos básicos para poder desarrollar el dispositivo.

En el capítulo 3, se introducen las galgas extensométricas y se entra de lleno en el diseño analógico del dispositivo (circuito de medida, amplificación y módulo de alarmas). Al final del capítulo se comentan los resultados obtenidos de la simulación.

En el capítulo 4, se desarrolla el circuito electrónico compuesto por el microcontrolador de la aplicación y las comunicaciones del dispositivo.

El capítulo 5, se centra en el *software* que debe implementar el microcontrolador para poder realizar su tarea.

En el capítulo 6, se diseña el circuito impreso del dispositivo diseñado.

En el capítulo 7, se realiza una estimación económica sobre la realización de este proyecto.

En el capítulo 8, se exponen las conclusiones extraídas durante el diseño de este dispositivo

A continuación, se podrá encontrar la bibliografía consultada para este TFC. El Anexo A que contiene el esquema completo del dispositivo y el Anexo B donde se puede encontrar el *software* de esta aplicación.

## Capítulo 2. Análisis de estructuras

El objeto de este capítulo es introducir al lector los conceptos básicos empleados para calcular la deformación de una viga. Existe toda una rama de la ingeniería civil dedicada al cálculo de deformaciones en estructuras. No es el objeto de este TFC el análisis exhaustivo de las deformaciones, en este TFC se utilizarán las expresiones y coeficientes genéricos empleados en arquitectura y nos centraremos en el diseño electrónico.

El primer concepto que vamos a introducir es el concepto de flecha de la viga [4]. Al aplicar una fuerza sobre una viga, la longitud de esta varía. Dependiendo del punto de aplicación de la fuerza, existirán zonas donde la longitud inicial de la viga ha aumentado, donde la longitud inicial de la viga ha disminuido y donde la longitud inicial de la viga se ha mantenido igual, denominaremos este último punto como zona neutral. En la figura 2.1 podemos observar una viga con 2 apoyos, esta viga solo está sometida a la fuerza que ejerce su propio peso, debido a la rigidez de la viga, esta no sufre ninguna deformación.



Figura 2.1. Viga en suspensión

Si a esta misma viga le aplicamos una fuerza tal y como se ve en la figura 2.2, esta viga sufrirá una deformación y describirá una trayectoria circular en el sentido de la fuerza, se puede observar que la longitud de la zona inferior de la viga, ha aumentado, la longitud de la zona superior de la viga ha disminuido y la longitud de la zona neutral se ha mantenido.

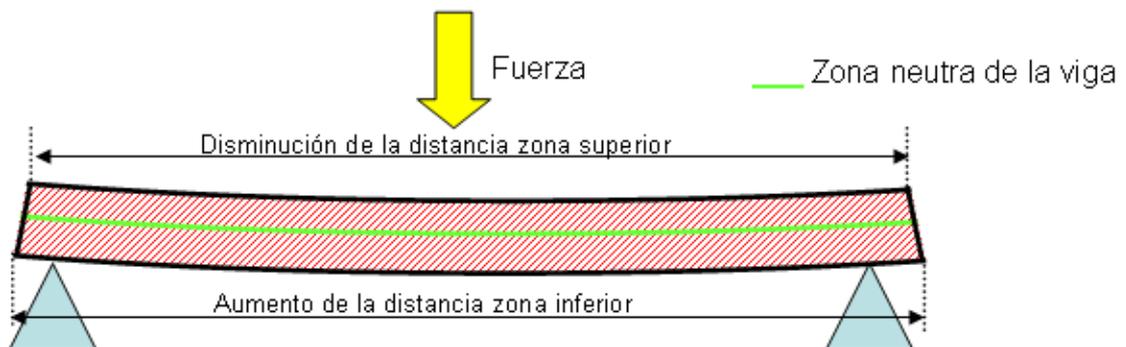
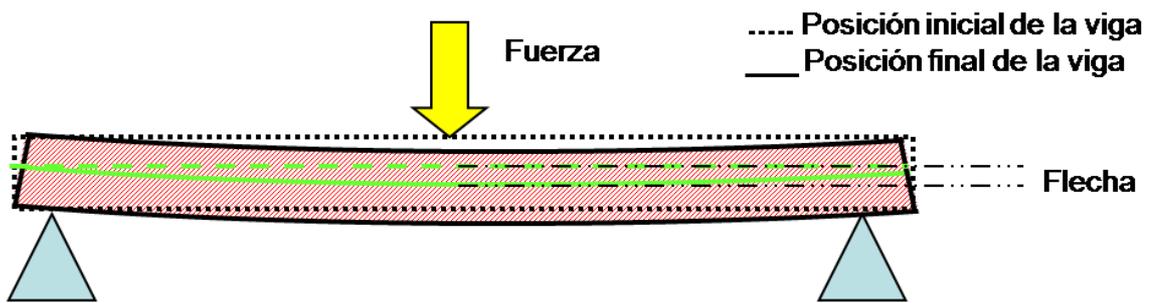


Figura 2.2. Viga sometida a una fuerza

Podemos definir la flecha de la viga como la distancia máxima que la zona neutra de la viga se puede desplazar sin que ésta represente un peligro para la estructura. Se puede ver una definición gráfica en la figura 2.3



**Figura 2.3. Flecha de una viga**

En función de cómo esta anclada a los puntos de apoyo esta viga, su longitud y el tipo de carga que deberá soportar, existen unas limitaciones en las flechas de las vigas. Tal y como he indicado anteriormente en este mismo capítulo, utilizaremos los coeficientes utilizados en arquitectura [4], [5]. Las limitaciones las podemos ver en la tabla 2.1.

**Tabla 2.1. Limitaciones de la flecha**

Modo de aplicación de la viga	limitación
Vigas o viguetas de cubierta	1/250
Vigas hasta 5m de luz y viguetas de forjado, que no soporten muros de Fábrica	1/300
Vigas de más de 5m de luz, que no soporten muros de fábrica	1/400
Vigas y viguetas de forjado, que soporten muros de fábrica	1/500
Ménsulas, con la flecha medida en el extremo libre	1/300

El siguiente concepto a introducir es la Luz de la viga, este término hace referencia a la longitud de la viga.

Existe una relación entre la flecha y la luz [5], ésta relación nos determinará cuál es la flecha máxima de una viga según su longitud y su modo de aplicación.

Para calcular la flecha máxima de la viga, primero debemos conocer la luz de la viga (longitud), después se debe escoger cual es el coeficiente de limitación máxima según la tabla 2.1.

Suponemos una viga de 5 metros para un forjado que soportará muros de fábrica. En la tabla 2.1 podemos ver que su limitación de flecha es de 1/500, por lo que su relación Flecha – Luz viene dado por:

$$\frac{Flecha_{(max)}}{Luz} \leq \frac{1}{500} \Rightarrow Flecha_{(max)} \leq \frac{Luz}{500} \Rightarrow Flecha_{(max)} \leq \frac{5000}{500} \quad (2.1)$$

$$Flecha_{(max)} \leq 10mm$$

De la expresión 2.1 sabemos que la Flecha máxima de nuestra estructura puede ser como máximo 10mm. Ahora suponemos que debido a una fuerza aplicada sobre la viga, esta sufre una deformación la cual hace variar su luz, a modo de ejemplo suponemos que la longitud final debido a la deformación es de 4950mm. Para conocer si la deformación que ha sufrido esta viga es crítica o no, debemos verificar que la relación Flecha-Luz no supera el coeficiente de limitación máxima.

$$Coef_{Limitacion} \frac{1}{500} = 0.002 \quad (2.2)$$

$$Relacion_{Flecha-Luz} = \frac{Flecha_{Maxima}}{Luz} = \frac{10}{4950} = 0.00202$$

Por lo tanto para este ejemplo dado que la Relación Flecha – Luz es mayor que el coeficiente de limitación máximo, la deformación que ha sufrido esta viga es peligrosa para la integridad de la estructura.

El umbral para conocer si una deformación es crítica o no lo es, variará en función de la luz de la viga y la flecha máxima de la viga. Dado que el dispositivo que pretendo diseñar, no está orientado a ninguna estructura específica, estos datos se deberán parametrizar en el dispositivo según las necesidades del emplazamiento de la estructura. Para realizar esta función, se ha pensado introducir una interface RS232 para poder configurar el dispositivo a través de un ordenador.

Como resumen de este capítulo, se ha deducido que en toda estructura, existen tres zonas de trabajo:

- Zona normal, es la deformación que puede sufrir una viga sin que sea necesario planificar una intervención de mantenimiento o afecte a la seguridad de la estructura.
- Zona de mantenimiento, es la zona donde la viga ha sufrido una deformación considerable pero la flecha de la viga no ha alcanzado el máximo permitido. Establecer el valor de la flecha para delimitar la zona de mantenimiento, es según el criterio del usuario. Este dispositivo permitirá seleccionar un valor mínimo de un 25% de la flecha hasta un valor máximo de un 75%.
- Zona crítica o de seguridad, es la zona en donde la flecha de la viga ha alcanzado su recorrido máximo, siendo una zona peligrosa para la integridad de la estructura.

Una vez analizada la sucesión de acontecimientos y fenómenos que ocurren durante una deformación, en el capítulo 3 comenzaremos a diseñar un circuito electrónico capaz de medir el grado de deformación.

## Capítulo 3 .Diseño de los bloques analógicos

El objeto de este capítulo es diseñar toda la parte de la electrónica analógica necesaria para llevar a cabo este TFC. Para ello primero se explica el funcionamiento de las galgas extensométricas, se dan a conocer las diferentes configuraciones de circuitos para llegar a conocer la deformación de la estructura y se diseña un circuito que ha de ser capaz de transformar la deformación sufrida por la estructura a valor de tensión, este circuito está compuesto por las galgas extensométricas y el módulo de amplificación.

### 3.1 Las galgas extensométricas

#### 3.1.1 Principio de funcionamiento

Una galga extensométrica es un componente piezoeléctrico el cual tiene la característica de variar su resistencia eléctrica cuando sufre una deformación [1].

Tal y como se ha visto en el capítulo 2, cuando a una viga se le aplica una fuerza suficiente para deformarla, la longitud de la viga varía en función de la fuerza que se aplica. Si colocamos una galga extensométrica en la superficie de la viga, las deformaciones que sufra la viga, también las sufrirá la galga y provocará una variación de su resistencia eléctrica, pudiendo aprovechar esta variación de resistencia eléctrica para conocer la magnitud de la deformación.

La unidad de medida de la deformación es el *strain* o deformación [1], la deformación se puede definir como el cociente entre la diferencia de longitudes y la longitud inicial de la viga.

$$\varepsilon = \frac{\Delta L}{L} \quad (3.1)$$

Donde  $\varepsilon$  es el *strain* o deformación relativa cuya unidad de medida es el  $\mu\varepsilon$  e indica el porcentaje de deformación dado que es una unidad adimensional,  $L$  es la longitud inicial de la viga y  $\Delta L$  es la diferencia de longitud de la viga una vez aplicada la fuerza.

La deformación puede ser positiva (tensión) o negativa (compresión), esto dependerá del punto donde se aplica la fuerza y la ubicación de las galgas. La deformación es una expresión adimensional, la cual suele representar el % de deformación. En la figura 3.1 la deformación que sufrirá la galga será positiva ya que la superficie de la galga al sufrir una deformación en el sentido de la fuerza se hará ligeramente mayor, lo que se traduce en un incremento de su resistencia eléctrica.

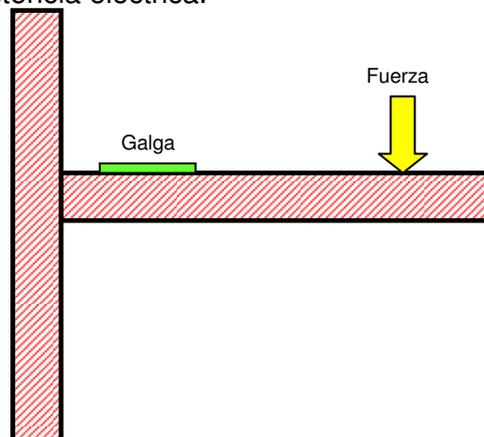
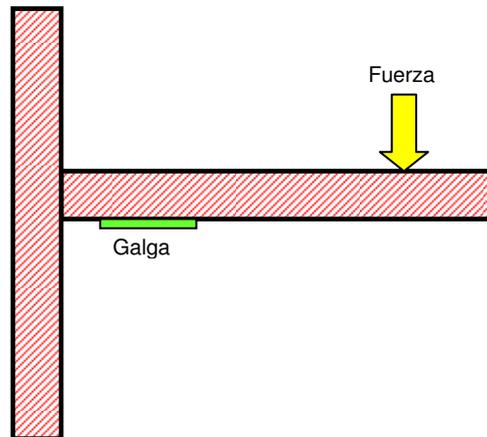


Figura 3.1 Galga en tensión

Si mantenemos la misma estructura y la misma fuerza, pero cambiamos la posición de la galga según se indica en la figura 3.2, la deformación que sufrirá la galga será negativa, ya que la superficie de la galga esta vez es ligeramente más pequeña, lo que se traduce en una disminución de su resistencia eléctrica.



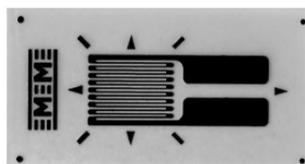
**Figura 3.2 Galga en compresión**

Una galga extensométrica, consiste en un cable muy fino adherido a un folio metálico, este cable tiene una resistencia fija la cual varía cuando la superficie del folio varía tal y como he indicado en el punto anterior, la resistencia aumenta cuando la galga está en tensión y la resistencia disminuye cuando la galga está en compresión.

Existen muchos modelos de galgas, los valores resistivos más estándares suelen ser de  $120\ \Omega$  y  $350\ \Omega$  [6]. El parámetro fundamental de la galga es el factor de galga o “Gage factor” GF. El factor de galga GF, se define como el ratio entre la variación de su longitud y la variación de su resistencia eléctrica [1].

$$GF = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} = \frac{\Delta R}{R \cdot \varepsilon} \quad (3.2)$$

Dependiendo de la geometría de la pieza de la que queremos conocer la deformación, existen infinidad de modelos de galgas extensométricas [6]. Para este TFC, se han seleccionado los modelos de galgas longitudinales, esto quiere decir que la variación de la resistencia eléctrica de la galga se producirá cuando exista una variación de longitud en el eje longitudinal de la galga, si la variación de longitud es en el eje transversal, ésta no variará su resistencia. En la figura 3.3 se puede ver una galga extensométrica longitudinal.



**Figura 3.3. Galga extensométrica**

Para la realización de este TFC, se ha seleccionado el modelo N11MA512023 de la casa RS [7]. La resistencia nominal de esta galga es de  $120\ \Omega$ , según su hoja de características su GF es 2.1.

Siguiendo con el ejemplo del capítulo 2, podemos calcular la variación de la resistencia eléctrica de la galga de la siguiente forma:

$$\varepsilon = \frac{\Delta L}{L} = \frac{5000 - 4950}{5000} = 0.01\varepsilon \quad (3.3)$$

$$GF = \frac{\frac{\Delta R}{R}}{\varepsilon} \Rightarrow \frac{\Delta R}{R} = GF(\varepsilon) = 2.1(0.01) = 0.021 \quad (3.4)$$

La ecuación 3.4, nos indica que la variación de resistencia de la galga será del 2.1% del valor nominal de la resistencia lo que representa un incremento o decremento (en función de donde se coloque la galga) de 2.52  $\Omega$ .

A continuación se realizan los mismos cálculos para ver la variación de la resistencia eléctrica de la galga frente a una deformación pequeña la cual varía la longitud de la viga en 1mm.

$$\varepsilon = \frac{\Delta L}{L} = \frac{5000 - 4999}{5000} = 0.0002\varepsilon \quad (3.5)$$

$$GF = \frac{\frac{\Delta R}{R}}{\varepsilon} \Rightarrow \frac{\Delta R}{R} = GF(\varepsilon) = 2.1(0.0002) = 0.00042 \quad (3.6)$$

La ecuación 3.6, indica que la resistencia variará en un 0.042% del valor nominal de la resistencia 0.0504  $\Omega$ .

Como se ha podido observar la variación de resistencia de las galgas con respecto a la deformación es muy pequeña. Para poder realizar unas mediciones de estos cambios tan pequeños de la resistencia eléctrica de la galga, es necesario un circuito eléctrico el cual sea sensible a estos pequeños cambios. Para realizar estas mediciones y conocer estas variaciones de resistencia, se dispondrán las galgas en configuración de puente de *Wheatstone* [2].

Como conclusiones de este capítulo, podemos decir que una galga extensométrica tiene dos modos de funcionamiento, compresión, figura 3.1 y estiramiento, figura 3.2. Cuando una galga trabaja en tensión el resultado es que su resistencia eléctrica aumenta y cuando una galga trabaja en compresión su resistencia eléctrica disminuye.

## 3.2 Mediciones en puente de Wheatstone

Tal y como hemos visto en el apartado 3.1, las variaciones en la resistencia eléctrica de la galga son muy pequeñas. Para poder medir estas pequeñas variaciones de resistencia, de la que no conocemos el valor, las galgas se suelen montar en configuración de puente de *Wheatstone*. La configuración puente de *Wheatstone*, consiste en 4 resistencias, a las que se le aplica una tensión de excitación o alimentación tal y como se indica en la figura 3.4. Si en este circuito conocemos el valor de 3 de las 4 resistencias, la salida del puente nos dará un valor de tensión en función del valor de la resistencia que desconocemos y queremos conocer el valor.

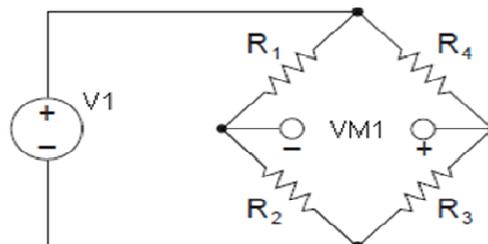


Figura 3.4. Puente de Wheatstone.

Si consideramos que  $R_1=R_2$  y  $R_3=R_4$ , y alimentamos el circuito con una tensión continua  $V_1$ , la tensión medida por el voltímetro  $VM1$  será cero. Cuando esta situación ocurre se dice que el puente está balanceado. Podemos observar que si una de las resistencias cambia su valor, el voltímetro  $VM1$ , leerá una tensión diferente de cero.

Si analizamos el puente mediante el principio del divisor de tensión [8], podemos conocer la tensión en los bornes del voltímetro  $VM1$  según se expresa en la ecuación 3.7.

$$VM1 = \left( \frac{R_4}{R_4 + R_3} - \frac{R_2}{R_2 + R_1} \right) V_1 \quad (3.7)$$

Para medir las variaciones de resistencia eléctrica en las galgas, sustituimos las resistencias del puente por las galgas, dependiendo del número de galgas que utilicemos, podemos disponer de 3 configuraciones del puente.

- Cuarto de puente, solo se utiliza una galga
- Medio puente, se introducen 2 galgas
- Puente completo, formado por 4 galgas.

Analíticamente, podemos decir que la resistencia eléctrica de una galga (a partir de ahora la denominaremos  $R_G$ ) tiene una parte fija y una parte variable, esta parte variable puede ser positiva o negativa según esté trabajando la galga, tensión o compresión.

$$R_G = R_G \pm \Delta R \quad (3.8)$$

### 3.2.1 Configuración cuarto de puente

Si en el circuito de la figura 3.4, sustituimos  $R_3$  por una galga, tal y como se indica en la figura 3.5, podemos reescribir la ecuación 3.7 como:

$$VM1 = \left( \frac{R_4}{R_4 + R_G + \Delta R} - \frac{R_2}{R_2 + R_1} \right) V_1 \quad (3.9)$$

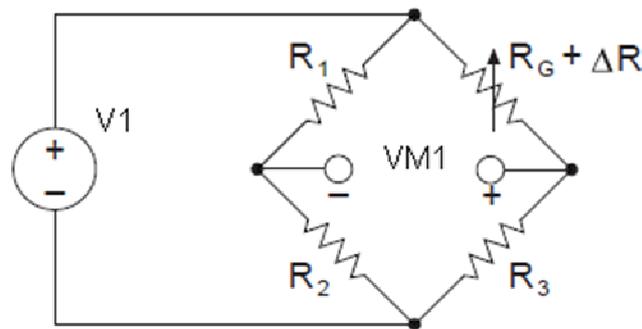


Figura 3.5. Configuración cuarto de puente.

Si tenemos en cuenta que para que el puente esté balanceado  $R_1 = R_2$  y  $R_4 = R_G$ , podemos desarrollar la expresión como a continuación:

$$VM1 = \left( \frac{R_4}{R_4 + R_4 + \Delta R} - \frac{R_2}{R_2 + R_1} \right) V_1 = \left( \frac{R_4}{R_4 + R_4 + GF\varepsilon} - \frac{1}{2} \right) V_1 = \left( \frac{-GF\varepsilon}{4 + 2GF\varepsilon} \right) V_1$$

$$\frac{VM1}{V1} = \frac{-GF\varepsilon}{4} \left( \frac{1}{1 + \frac{GF\varepsilon}{2}} \right) \quad (3.10)$$

En esta relación podemos observar que la salida del puente no es lineal debido al término

$$\left( \frac{1}{1 + \frac{GF\varepsilon}{2}} \right)$$

### 3.2.2 Configuración medio de puente

Si a la configuración de cuarto de puente le introducimos otra galga tal y como se muestra en la figura 3.6, tendremos una configuración en medio puente donde aplicando la misma teoría de análisis que en el caso del cuarto de puente, la ecuación de salida será:

$$VM1 = \left( \frac{R_G - \Delta R}{R_G - \Delta R + R_G + \Delta R} - \frac{R_2}{R_2 + R_1} \right) V1 = \left( \frac{R_G - \Delta R}{2R_G} - \frac{1}{2} \right) V1 = \left( \frac{1 - GF\varepsilon}{2} - \frac{1}{2} \right) V1 \quad (3.11)$$

$$\frac{VM1}{V1} = \frac{-GF\varepsilon}{2}$$

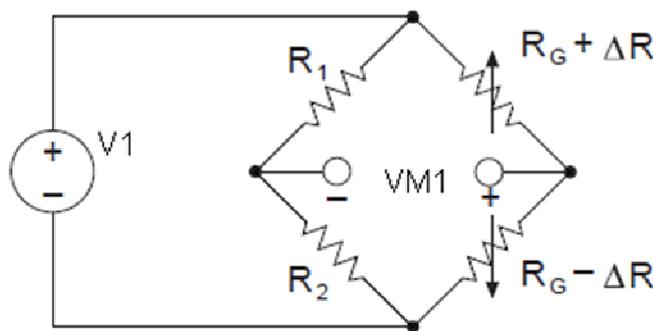


Figura 3.6. Configuración medio puente.

Se ha de notar que la relación de salida 3.11 para este tipo de configuración no introduce ningún componente de no linealidad.

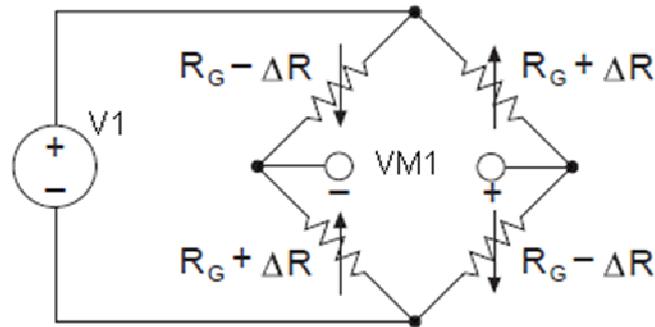
### 3.2.3 Configuración puente completo

La configuración en puente completo, consiste en reemplazar todas las resistencias del puente por galgas tal y como se indica en la figura 3.7. Actuando de la misma manera que en los dos casos anteriores la relación de salida del puente será:

$$VM1 = \left( \frac{R_G - \Delta R}{R_G - \Delta R + R_G + \Delta R} - \frac{R_G + \Delta R}{R_G - \Delta R + R_G + \Delta R} \right) V1 = \left( \frac{R_G - \Delta R}{2R_G} - \frac{R_G + \Delta R}{2R_G} \right) V1$$

$$= \left( \frac{1 - GF\varepsilon}{2} - \frac{1 + GF\varepsilon}{2} \right) V1 = GF\varepsilon V1 \quad (3.12)$$

$$\frac{VM1}{V1} = GF\varepsilon$$



**Figura 3.7. Configuración puente completo.**

Nótese que esta configuración también es lineal.

Si comparamos la ecuación 3.11 (cuarto de puente) con la 3.12 (puente completo), podemos ver que la tensión de salida de esta última configuración es el doble que la configuración de cuarto de puente, por tanto esta última configuración es la que mayor sensibilidad ofrece manteniendo la linealidad.

### 3.3 Selección de la configuración

En el apartado 3.2 hemos podido ver las diferentes configuraciones de puente que se pueden emplear, para este TFC utilizaremos la configuración de medio puente dado que se ha visto que esta configuración es lineal. Para ello se emplearán 2 puentes, cada uno de los puentes medirá la deformación en la viga en los planos X e Y.

Para medir la deformación en cada uno de los planos, se utilizarán 2 galgas por plano, 1 de ellas estará dispuesta en la dirección del eje longitudinal del plano y la otra galga estará dispuesta en la cara opuesta de la viga también en la dirección longitudinal de la viga según se muestra en las figuras 3.8 y 3.9



**Figura 3.8. Disposición de las galgas sobre el plano X**



**Figura 3.9. Disposición de las galgas sobre el plano Y.**

### 3.3.1 Valores de salida del puente

Una vez ya sabemos que configuración vamos a utilizar y antes de empezar con la etapa amplificadora, vamos a calcular que valores de tensión vamos a tener en la salida del puente.

Siguiendo con nuestro ejemplo del capítulo 2, vamos a calcular la tensión de salida del puente para los dos casos ya planteados ( $\Delta L = 1mm$  y  $\Delta L = 50mm$ ). Para ello utilizaremos la ecuación 3.11. La tensión de alimentación del puente la fijaremos en 6v, no es una tensión estándar como podrían ser 5v o 12v, la elección del valor de 6v se ha hecho pensando en el diseño de la etapa amplificadora, recordamos que las características del modelo de la galga seleccionada eran  $GF=2.1$  y  $R_G=120 \Omega$ .

Caso  $\Delta L = 1mm$

$$\frac{VM1}{V1} = \frac{-GF\varepsilon}{2} \Rightarrow VM1 = \frac{-GF\varepsilon}{2} V1 = \frac{-2.1(0.0002\varepsilon)}{2} 6v = -0.0012v \quad (3.13)$$

Caso  $\Delta L = 50mm$

$$\frac{VM1}{V1} = \frac{-GF\varepsilon}{2} \Rightarrow VM1 = \frac{-GF\varepsilon}{2} V1 = \frac{-2.1(0.01\varepsilon)}{2} 6v = -0.063v \quad (3.14)$$

Una vez calculados los valores de tensión en la salida del puente, podemos tomarlos como los máximos y mínimos, vemos que la diferencia entre el máximo y el mínimo es de tan solo 0.0618v, en el diseño de la etapa amplificadora tomaremos estos valores para su diseño.

### 3.3.2 Tolerancias y diseño del circuito de medida

En la configuración de medio puente de la figura 3.6, las resistencias R1 y R2, han de ser del mismo valor para que la rama que forman este par de resistencias este balanceada, esto quiere decir que la tensión de R1 tiene que ser igual a la tensión de R2.

Hoy en día el valor de una resistencia comercial puede tener tolerancias de  $\pm 1\%$ ,  $\pm 5\%$  y  $\pm 10\%$ . Si se seleccionaran resistencias con una tolerancia del 1%, la posible variación del valor de las resistencias, podrían desbalancear la rama y con ello el puente.

Para corregir este efecto se ha optado por introducir un regulador de tensión TL431 [9], el cual nos dará una tensión de referencia que podemos ajustar según nuestras necesidades. Como ya sabemos, la tensión de alimentación del puente es de 6v, y como  $VR1=VR2$ , sabemos que se debe ajustar el regulador de tensión para que asegure una tensión constante de 3v.

En el *datasheet* del TL431 [9], el fabricante nos indica la ecuación para poder fijar el valor de la salida según nuestra necesidad y configuración seleccionada, tal y como se puede ver en la figura 3.10.

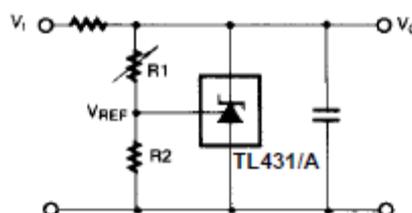


Figura 3.10. Configuración regulador TL431.

$$V_o = \left(1 + \frac{R1}{R2}\right) V_{ref} \quad (3.15)$$

En el mismo *datasheet* se puede ver que la tensión de referencia típica  $V_{ref}$  es de 2,495v, con este valor y fijando  $V_o=3v$  y  $R2=2k7$ , se obtiene un valor de  $R1=546,49\Omega$ .

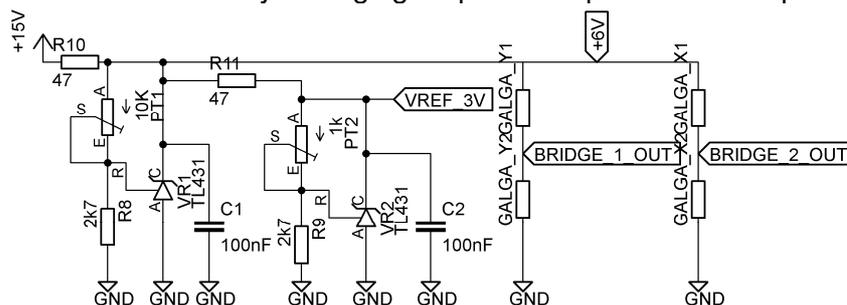
Dado que el valor de  $R1$ , no es un valor estándar, se ha optado por poner un potenciómetro de 1k, con este potenciómetro se podrá ajustar la tensión de referencia de 3v que necesitamos con mayor precisión.

La tensión de alimentación del puente, también es otro parámetro a tener en cuenta. Esta tensión debería ser siempre constante, sin ruido y sin picos. Dado que no se sabe la calidad de la fuente de alimentación que pueda alimentar esta aplicación, también se ha decidido montar otro regulador de tensión para alimentar el puente.

Aunque en ambos casos los dos reguladores regulan la tensión, esta vez este regulador nos servirá para mantener la tensión de alimentación estable a 6v aunque la tensión de alimentación de la aplicación pueda sufrir alguna variación.

Procediendo de la misma forma que para el caso anterior, fijamos  $V_o=6v$  y  $R2=2k7$  y obtenemos el valor de  $R1=3792,985\Omega$ , igualmente para esta resistencia, colocaremos un potenciómetro de ajuste de un valor de 10K.

La figura 3.11 muestra el esquema final de la etapa de medida, en esta figura podemos ver los 2 reguladores de tensión y las 4 galgas que se emplearán en la aplicación.



**Figura 3.11. Esquema módulo de medición.**

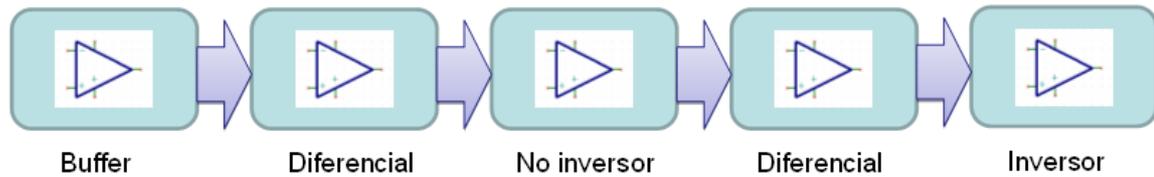
Con la terminación de este apartado 3.3, se da por alcanzado el primer objetivo específico para este TFC.

## 3.4 Etapa amplificadora

En las ecuaciones de salida del puente 3.13 y 3.14, hemos podido ver que los valores de tensión para una deformación mínima y para una deformación máxima, son muy bajos. Trabajar con estos niveles de tensión tan bajos, es una tarea casi imposible. Para facilitar la tarea, es necesario amplificar esta tensión y obtener un rango de tensiones con los que podamos trabajar con suma facilidad, por ello es necesario diseñar una etapa amplificadora.

Para el diseño de la etapa amplificadora se han utilizado amplificadores operacionales. Esta etapa consta de 4 bloques tal y como se puede ver en la figura 3.12. El primer bloque esta constituido por un amplificador operacional en configuración *buffer* [3], seguidamente la salida del *buffer* entra en otro amplificador operacional en configuración diferencial [3]. Esta salida entra en otro amplificador operacional en configuración no inversor con ganancia 40 [3], la salida la hacemos pasar por un segundo amplificador diferencial y la salida de éste entra finalmente en el último amplificador operacional en configuración inversor y ganancia unidad [3]. A la salida de este último amplificador

operacional, tendremos la señal amplificada y acondicionada para poder ser muestreada por el convertor analogico digital.



**Figura 3.12. Bloques etapa amplificadora.**

### 3.4.1 El circuito buffer

Para el bloque del *buffer* se ha seleccionado el amplificador operacional TL071 [10] de la casa Texas Instruments.

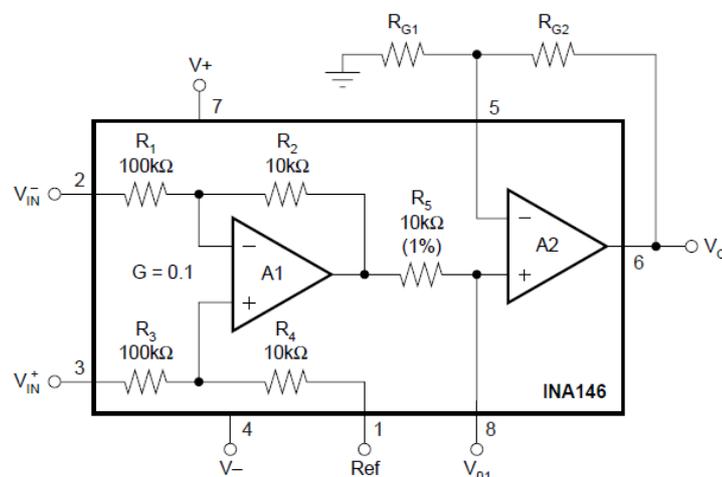
El circuito de medida de la sección 3.3 es un circuito muy sensible, a la salida de los puentes la impedancia es muy baja, por lo que para poder medir necesitamos un equipo de medición el cual afecte lo menos posible a la estabilidad del puente, es decir necesitamos que el equipo de medición tenga una impedancia de entrada muy grande. Una de las características de la configuración *buffer* es que tiene una alta impedancia de entrada [3], que es precisamente lo que se está buscando. La ecuación de salida de esta configuración es [3] :

$$V_{out} = V_{in} \quad (3.16)$$

Siguiendo con el ejemplo de la viga, y aplicando la ecuación 3.16 a los resultados obtenidos en las ecuaciones 3.13 y 3.14, vemos que a la salida del *buffer* seguiremos manteniendo los mismos resultados.

### 3.4.2 Bloque diferencial y no inversor

Para el diseño de estos 2 bloques se ha utilizado un amplificador diferencial de ganancia programable de la casa Texas instruments, el modelo de amplificador es el INA 146 [11]. Este amplificador esta compuesto por 2 amplificadores operacionales, uno de ellos configurado como amplificador diferencial y el otro configurado como amplificador no inversor. En la imagen 3.10 se muestra la composición interna del INA146.



**Figura 3.13. Estructura INA 146.**

Este amplificador es muy parecido a los llamados amplificadores de instrumentación, la gran diferencia entre ambos amplificadores es que los amplificadores de instrumentación

incorporan un *buffer* en la entrada, pero no incorporan ningún amplificador (inversor o no inversor) en la salida.

Tal y como podemos ver en la imagen anterior, el amplificador operacional A1, está montado en configuración de amplificador diferencial con una ganancia  $G=0.1$ . A la salida de A1 la tendremos una tensión de salida según la ecuación siguiente [11]:

$$V_{01} = (V_{IN}^+ - V_{IN}^-)G = (V_{IN}^+ - V_{IN}^-)0.1 \quad (3.17)$$

Hemos de notar que el resultado del término  $(V_{IN}^+ - V_{IN}^-)$  es igual al resultado de las ecuaciones 3.13 y 3.14. Por tanto en la salida  $V_{01}$  de INA 146 tenemos la salida del puente multiplicada por la ganancia 0.1.

La salida del A1 se conecta a la pata no inversora del A2, que está montado en configuración de amplificador no inversor. La ganancia de este amplificador se establece mediante las resistencias  $R_{G1}$  y  $R_{G2}$ . Esta ganancia viene dada por la ecuación [11]:

$$G = 1 + \frac{RG_2}{RG_1} \quad (3.18)$$

Por lo tanto la ganancia total del INA146 será:

$$G_{total} = 0.1 \left( 1 + \frac{RG_2}{RG_1} \right) \quad (3.19)$$

Para esta aplicación fijaremos la ganancia total en 40 y fijaremos un valor de  $RG_1$  de  $100\Omega$  por tanto:

$$R_{G2} = \left( \frac{G}{0.1} - 1 \right) R_{G1} = \left( \frac{40}{0.1} - 1 \right) 100 = 39.9K\Omega \quad (3.20)$$

Con una  $G=40$ , podemos volver a calcular los máximos y mínimos obtenidos en las ecuaciones 3.13 y 3.14, dado que el *buffer* no introduce ninguna modificación.

$$V_o = VM1(G) \quad (3.21)$$

Caso  $\Delta L = 1mm$

$$V_o = -0.0012(40) = -0.048v \quad (3.22)$$

Caso  $\Delta L = 50mm$

$$V_o = -0.063(40) = -2.52v \quad (3.23)$$

Es importante destacar que la ganancia seleccionada para este amplificador tiene que asegurar que la tensión de salida de todo el módulo de amplificación no supere los 6v, por este motivo se ha seleccionado una ganancia de 40.

Como se verá en el capítulo 4, la máxima tensión que podrá manejar el conversor analógico digital es de 6 v, si la tensión de salida del módulo de amplificación es superior a 6v, la lectura que realice el conversor analógico digital será errónea.

### 3.4.3 El segundo amplificador diferencial

Hasta el momento, solo se ha tenido en cuenta que a la viga del ejemplo, se le aplicaba una fuerza en un sentido, si esta fuerza se aplica con la misma magnitud pero en sentido contrario, todas las ecuaciones para calcular la tensión en la salida del puente cambiarían el signo de la tensión.

Para la ecuación 3.22, la tensión será de +0.048v y para la ecuación 3.23, la tensión será de +2.52v, por lo tanto dependiendo donde se aplique la fuerza a la viga, la tensión en la salida del puente podrá oscilar entre una tensión negativa y una tensión positiva, por lo

que se debe implementar un circuito que nos mantenga la salida en un rango de tensiones positivas o negativas y este efecto se consigue con este segundo amplificador diferencial. Para este bloque se ha utilizado el amplificador operacional TL071.

La ecuación de salida del amplificador diferencia ya se vio en la ecuación 3.17, en este caso la ganancia del amplificador será la unidad, y en  $V_{IN}^-$  aplicaremos una tensión de 3v. A la salida de este bloque tendremos las siguientes tensiones:

Caso  $\Delta L = 1mm$

$$V_0 = (V_{IN}^+ - V_{IN}^-)G = (-0.048 - 3)1 = -3.048v \quad (3.24)$$

$$V_0 = (V_{IN}^+ - V_{IN}^-)G = (0.048 - 3)1 = -2.952v \quad (3.25)$$

Caso  $\Delta L = 50mm$

$$V_0 = (V_{IN}^+ - V_{IN}^-)G = (-2.52 - 3)1 = -5.52v \quad (3.26)$$

$$V_0 = (V_{IN}^+ - V_{IN}^-)G = (2.52 - 3)1 = -0.48v \quad (3.27)$$

Para obtener una ganancia igual a la unidad,  $R1=R2=R3=R4$ , [3].

### 3.4.4 El bloque no inversor

Para este último bloque también se ha utilizado el amplificador operacional TL071, en esta ocasión se monta en configuración no inversor según se puede observar en la figura 3.14. El motivo de introducir este último bloque es la necesidad de disponer de una tensión positiva a la entrada del conversor analógico digital.

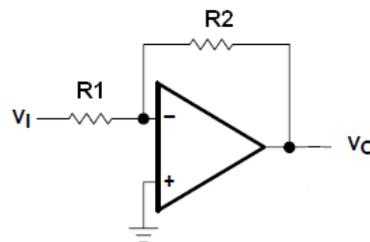


Figura 3.14. Configuración no inversor

La ecuación de salida de esta configuración viene dada por la siguiente ecuación [3]:

$$V_0 = -V_I G = -V_I \frac{R2}{R1} \quad (3.28)$$

Tal y como se ha dicho en la introducción del apartado 3.4, este bloque tiene ganancia unidad, por lo que de la ecuación 3.24 se deduce que  $R2=R1$  [3].

Aplicando la ecuación 3.24 a las ecuaciones 3.22 y 3.23, las tensiones de salida del módulo de amplificación serán:

Caso  $\Delta L = 1mm$

$$V_0 = -(-3.048v)1 = 3.048v \quad (3.29)$$

$$V_0 = -(-2.952v)1 = 2.952v \quad (3.30)$$

Caso  $\Delta L = 50mm$

$$V_0 = -(-5.52v)1 = 5.52v \quad (3.31)$$

$$V_0 = -(-0.48v)1 = 0.48v \quad (3.32)$$

Se puede observar que los valores de tensión calculados, cumplen con el requisito de no superar una tensión de 6v.

### 3.4.5 Esquema del módulo de amplificación

En la imagen 3.14 se puede ver el diseño final de la etapa amplificadora, son dos etapas iguales, una para las galgas del eje X y la otra para las galgas del eje Y de la viga.

En este esquema se puede ver que no se ha introducido ningún elemento de ajuste para ajustar la posible *offset* [3] a 0v. Para corregir este efecto, el tratamiento y eliminación de la *offset* se realizará mediante software.

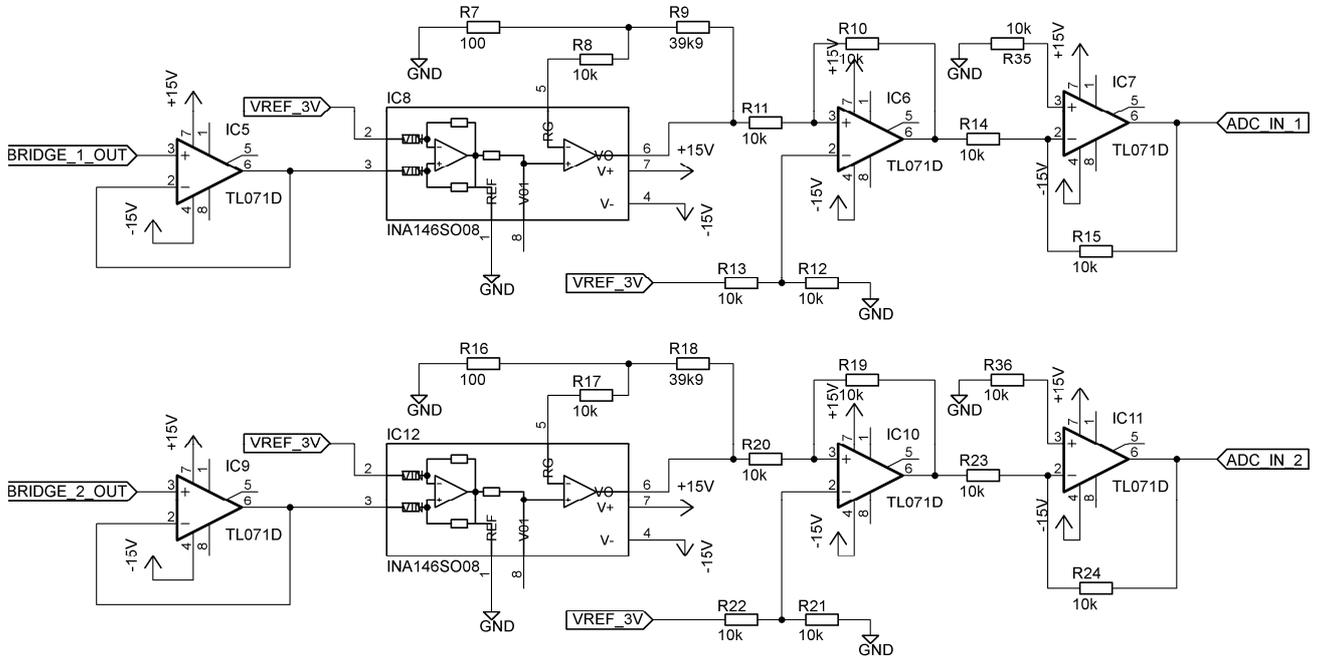


Figura 3.15. Esquema del módulo de amplificación.

En la tabla 3.1 se pueden ver los rangos de tensiones para cada zona de trabajo, esta tabla esta representada gráficamente en la figura 3.16.

Tabla 3.1. Rango de tensiones para las zonas de trabajo

Zona de trabajo	Superior	Inferior
Zona no segura	$\geq 5,52V$	$\leq 0,48V$
Zona de mantenimiento 25% hasta 100%	$> 3,36V$ y $< 5,52V$	$> 2,37V$ y $< 0,48V$
Zona segura 0% hasta 25%	$\geq 2,37V$ hasta $\leq 3,36V$	

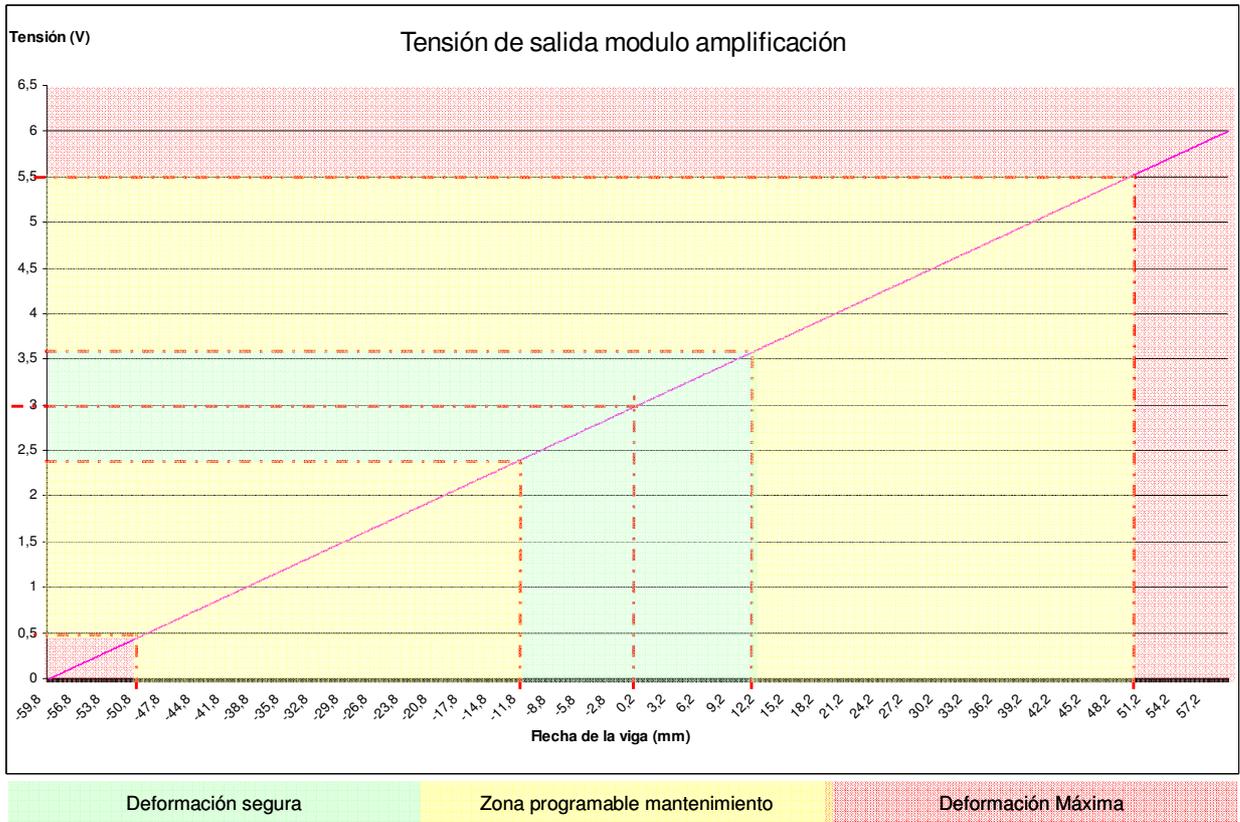
Con la terminación de este apartado 3.4, se da por alcanzado el segundo objetivo específico para este TFC.

## 3.5 Módulo de alarmas

El circuito de alarmas está compuesto por 2 leds y un zumbador. Uno de los leds nos indicará si el sistema está trabajando correctamente, o por el contrario el sistema tiene un error y no está trabajando. El segundo led nos avisará cuando la estructura alcance una deformación que nosotros deberemos haber programado en el sistema. Por último el zumbador nos avisará cuando flecha de la viga sea mayor que la flecha permitida.

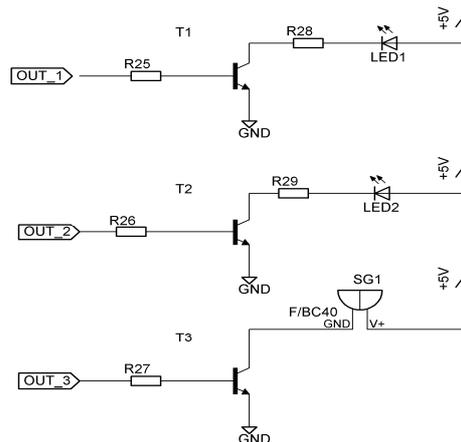
En la gráfica mostrada en la figura 3.16, se pueden observar las 3 zonas de funcionamiento, en verde la zona de trabajo, en amarillo la zona de mantenimiento y en rojo la zona crítica o seguridad, en la gráfica se puede observar también a que valor de tensión corresponde cada zona, es decir si a la salida del circuito de amplificación

tenemos una salida de 3,32v, podemos ver que la deformación que ha sufrido la viga se encuentra dentro de la zona de trabajo y no es necesaria la activación de ninguna alarma. Esta gráfica está basada en los datos del ejemplo utilizado para este TFC y los valores se pueden encontrar en la tabla 3.1.



**Figura 3.16. Zonas de trabajo.**

Los 3 avisadores se activarán a través del módulo discriminador (microcontrolador), para ello se introduce una etapa de potencia para cada avisador. Esta etapa de potencia, está formada por un transistor BC547 [14]. En la figura 3.17, podemos ver el diseño del módulo de alarmas.



**Figura 3.17. Módulo de alarmas.**

Primero calcularemos aplicando la ley de *Ohm* [8] el valor de las resistencias R28 y R29, en el *datasheet* del led (modelo TLHK5100) [12], podemos ver que la tensión directa del led es de 2v y la intensidad que consume es de 20mA, por tanto:

$$R28 = R29 = \frac{VR}{I_{led}} = \frac{5v - 2v}{20mA} = 1,3K\Omega \tag{3.33}$$

Calculo de R25 y R26, en el *datasheet* del transistor BC547, podemos ver que tiene un valor  $h_{FE}$  de 110 a 220, para este caso cogemos la mínima  $h_{FE}$  [3], [14]. Cogemos la mínima  $h_{FE}$  ya que en la gráfica de ganancia del transistor, la  $h_{FE}$  máxima comienza a caer exponencialmente con una  $I_C=30mA$ . Seleccionando la  $h_{FE}$  mínima nos aseguramos que el transistor continúe funcionando correctamente si la  $I_C$  sube más de 30mA.

$$R25 = R26 = \frac{VR}{I_{base}} = \frac{5v - V_{Base-Emisor}}{\frac{I_{Colector}}{h_{FE}}} = \frac{5v - 0.6}{\frac{20mA}{110}} = 24,2K\Omega \quad (3.34)$$

Para el cálculo de R27, consultamos el *datasheet* del zumbador KT-400142 [13] y vemos que para una tensión de 5v consume una intensidad de 5mA, para este caso tomaremos la misma  $h_{FE}$  del transistor,  $h_{FE} = 110$  [3].

$$R27 = \frac{VBuzzer}{I_{base}} = \frac{5v}{\frac{I_{Colector}}{h_{FE}}} = \frac{5v}{\frac{5mA}{110}} = 110K\Omega \quad (3.35)$$

Con la terminación de este apartado se da por alcanzado el tercer objetivo de este TFC.

### 3.6 La fuente de alimentación

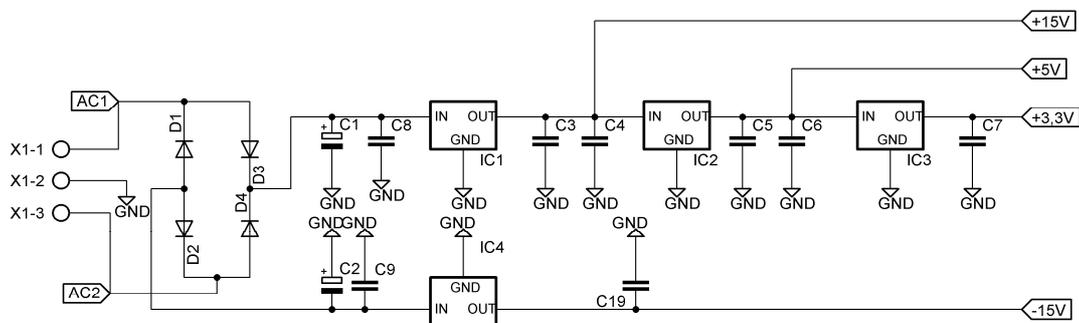
Los requerimientos de alimentación para esta aplicación los podemos ver en la tabla 3.1:

**Tabla 3.2. Necesidades de alimentación de la aplicación**

Componente	Tensión de alimentación	Tipo de tensión	Intensidad máxima	Cantidad de componentes	Total intensidad
TL071	±15v	Dual	2.5 mA	2	5 mA
INA146	±15v	Dual	15 mA	1	15 mA
Microcontrolador	+3.3 v	Simple	300 mA	1	300 mA
Led	+5 v	Simple	20 mA	2	40 mA
Zumbador	+5 v	Simple	5 mA	1	5 mA
Puente	+6 v	Simple	25 mA	2	50 mA
Módulo BT	+3.3v	Simple	10mA	1	10mA

Según la tabla 3.1, vemos que la fuente de alimentación debe tener 3 salidas de tensión diferentes y el consumo máximo de intensidad puede llegar a ser de 415 mA. Para las salidas de +15v y 5v, se utilizarán reguladores de tensión estándar de la serie LM78xx [15] para las tensiones positivas y el regulador de tensión LM7915 [16] para la tensión negativa de -15v. La tensión de 6v, ya quedó resuelta en el apartado 3.3.

En los *datasheet* de los reguladores podemos ver que según el tipo de encapsulado que seleccionemos, estos reguladores soportan una intensidad de hasta 1A con encapsulado TO220, por lo que cumple con nuestra necesidad. En la figura 3.18, se muestra el diseño completo de la fuente de alimentación.



**Figura 3.18. Esquema final de la fuente de alimentación.**

### 3.7 Simulación en Tina-ti

Para la simulación de la parte analógica de la aplicación, se ha pensado realizar la simulación del circuito de medida junto con la etapa amplificadora, solo para uno de los 2 puentes que constituyen la aplicación, los resultados obtenidos son aplicables al segundo puente, ya que sus diseños son iguales.

Para la etapa de potencia no se ha realizado ninguna simulación dado que es una configuración muy trivial.

Igualmente para la fuente de alimentación, tampoco se ha creído necesario realizar ninguna simulación dado que los reguladores están montados según la información de las hojas de características.

#### 3.7.1 Simulación del puente y etapa amplificadora

La figura 3.19 muestra el esquema empleado para la simulación. En este circuito se han puesto 5 voltímetros (VM1-VM5), la función de cada voltímetro es la siguiente:

- VM1 mide la tensión de alimentación del puente.
- VM2 mide la tensión de referencia que se aplicará a la etapa amplificadora
- VM3 mide la tensión a la salida de unos de los puentes "Y"
- VM4 mide la tensión a la salida del otro puente "X"
- VM5 mide la tensión a la salida del módulo de amplificación

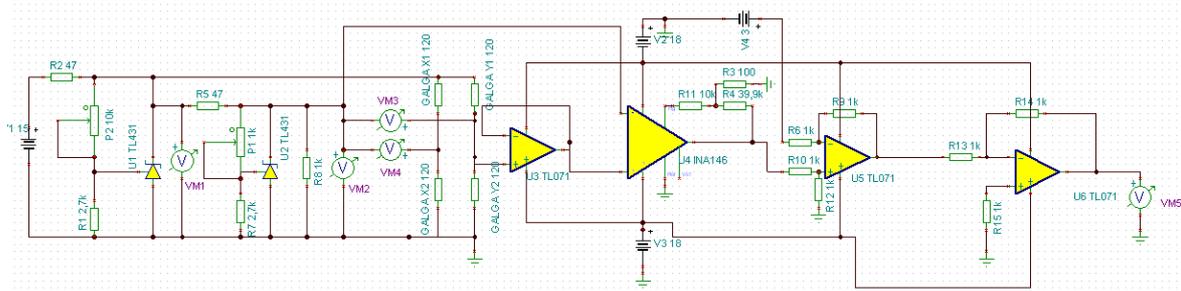


Figura 3.19. Circuito de simulación en Tina-ti.

Antes de realizar la simulación ha sido necesario ajustar los valores de alimentación del puente (6v) y tensión de referencia de la etapa amplificador (3v) mediante los potenciómetros P2 (10k $\Omega$ ) y P1 (1k $\Omega$ ) respectivamente.

Una vez ajustados los potenciómetros P2 y P1, la simulación de esta parte del circuito se ha realizado en 3 fases:

- Primera fase, medición de la *offset* mediante el voltímetro VM5
- Segunda fase, se modifican los valores de las galgas de un eje según la ecuación 3.4 y se hace una medición a la salida de la etapa amplificadora mediante el voltímetro VM5
- Tercera fase, se modifican los valores de las galgas de un eje según la ecuación 3.6 y se hace una medición a la salida de la etapa amplificadora mediante el voltímetro VM5

Con los resultados obtenidos por la simulación, se verificará que los valores a la salida de la etapa amplificadora se corresponden con los valores de tensión calculados según las ecuaciones 3.29 a 3.32.

Una vez verificada la funcionalidad del circuito, se realiza el análisis de frecuencias de la etapa amplificadora donde se verificará la funcionalidad de ésta.

Los resultados de la primera fase de la simulación se pueden ver en la figura 3.20. Aquí observamos que la salida de los puentes no es 0, existe una tensión muy pequeña del

orden de nV, esta pequeña tensión aplicada a la etapa amplificadora y sumada la posible *offset* de los amplificadores operacionales, nos da como resultado un *offset* total de 72mV.

Monitor	Value
VM1	6V
VM2	3V
VM3	391.08507nV
VM4	394.14892nV
VM5	3.07178V

**Figura 3.20. Resultados de la primera fase, medición de la *offset***

La figura 3.21 muestra los resultados de la segunda fase de la simulación. Solo se realiza la simulación de uno de los 2 puentes, el puente Y, por analogía los resultados obtenidos son validos para el puente X.

Para esta simulación se han modificado los valores resistivos de las galgas “Galga Y1” y “Galga Y2” según la ecuación 3.4, Galga Y1=120.0504 $\Omega$  y Galga Y2=119.9496 $\Omega$ .

Monitor	Value
VM1	6V
VM2	3V
VM3	-1.25961mV
VM4	394.1516nV
VM5	3.12211V

**Figura 3.21. Resultados de la segunda fase (caso  $\Delta L=1\text{mm}$ ).**

La tensión medida a la salida de la etapa amplificadora es de 3.1221v, si a este resultado le restamos la *offset*, se corresponde con el valor calculado en la ecuación 3.29.

Para la siguiente simulación se han invertido los valores resistivos de las galgas “Galga Y1” y “Galga Y2”, Galga Y2=120.0504 $\Omega$  y Galga Y1=119.9496 $\Omega$ . Se pueden ver los resultados en la figura 3.22.

Node	Value
VM1	6V
VM2	3V
VM3	1,26039mV
VM4	394,14754nV
VM5	3,02145V

Show

Nodal Voltages     Currents

Other Voltages     Outputs

Cancel    Help

**Figura 3.22. Resultados de la segunda fase (caso  $\Delta L=1\text{mm}$ ).**

La tensión medida a la salida de la etapa amplificadora es de 3.02145v, igual que en la simulación anterior, si a este resultado le restamos la *offset*, se corresponde con el valor calculado en la ecuación 3.30.

Para la tercera fase de la simulación, se realiza el mismo proceso que para la segunda fase pero con los resultados de la ecuación 3.6, Galga Y1=122.52 $\Omega$  y Galga Y2=117.48 $\Omega$ . Los resultados de la simulación se pueden ver en la imagen 3.21, vemos que la tensión medida a la salida de la etapa amplificadora es de 5.5887v tal y como indica la figura 3.23 y se calculó en la ecuación 3.31.

Node	Value
VM1	6V
VM2	3V
VM3	-62,99961mV
VM4	394,23933nV
VM5	5,58871V

Show

Nodal Voltages     Currents

Other Voltages     Outputs

Cancel    Help

**Figura 3.23. Resultados de la tercera fase (caso  $\Delta L=50\text{mm}$ ).**

Para la siguiente simulación se han invertido los valores resistivos de las galgas “Galga Y1” y “Galga Y2”, Galga Y2=122.52 $\Omega$  y Galga Y1=117.48 $\Omega$ . Se pueden ver los resultados en la figura 3.24 y se calculó en la ecuación 3.32.

Node	Value
VM1	6V
VM2	3V
VM3	63,00039mV
VM4	394,05877nV
VM5	554,87552mV

Show

Nodal Voltages     Currents

Other Voltages     Outputs

Cancel    Help

**Figura 3.24. Resultados de la tercera fase (caso  $\Delta L=50\text{mm}$ ).**

Para verificar que los resultados obtenidos en la simulación se corresponden con los resultados calculados en las ecuaciones 3.29 a 3.32, es necesario que a los resultados de la segunda y tercera fase se les reste la tensión *offset* medida en la primera fase.

$$\text{Result. 2}^{\text{a}} \text{ fase} = 3.1221\text{V} - 72\text{mV} = 3.05\text{V} \quad (3.36)$$

$$\text{Result. 2}^{\text{a}} \text{ fase} = 3.02145\text{V} - 72\text{mV} = 2.9494\text{V} \quad (3.37)$$

$$\text{Result. 2}^{\text{a}} \text{ fase} = 5.5887\text{V} - 72\text{mV} = 5.516\text{V} \quad (3.38)$$

$$\text{Result. 2}^{\text{a}} \text{ fase} = 554.87\text{mV} - 72\text{mV} = 482.87\text{mV} \quad (3.39)$$

Por tanto, los resultados de las ecuaciones 3.36 a 3.39, se corresponden con los cálculos teóricos de las ecuaciones 3.29 a 3.32 respectivamente. De estos resultados podemos afirmar que la funcionalidad del circuito es válida.

El análisis de frecuencias del circuito, (ver figura 3.25) nos indica que tendremos una ganancia constante de 40 dB para una frecuencia de entrada menor a 400 Hz. Para frecuencias superiores, esta ganancia comienza a decaer.

La salida de los puentes de medición es una salida de corriente continua. La componente frecuencial vendrá dada por la frecuencia con que se puedan suceder las posibles deformaciones en la estructura, dado que las deformaciones se producen de forma progresiva, un ancho de banda de 400 Hz es suficiente para asegurar el correcto funcionamiento de la aplicación, las deformaciones serán lentas en comparación de una frecuencia de 400Hz.

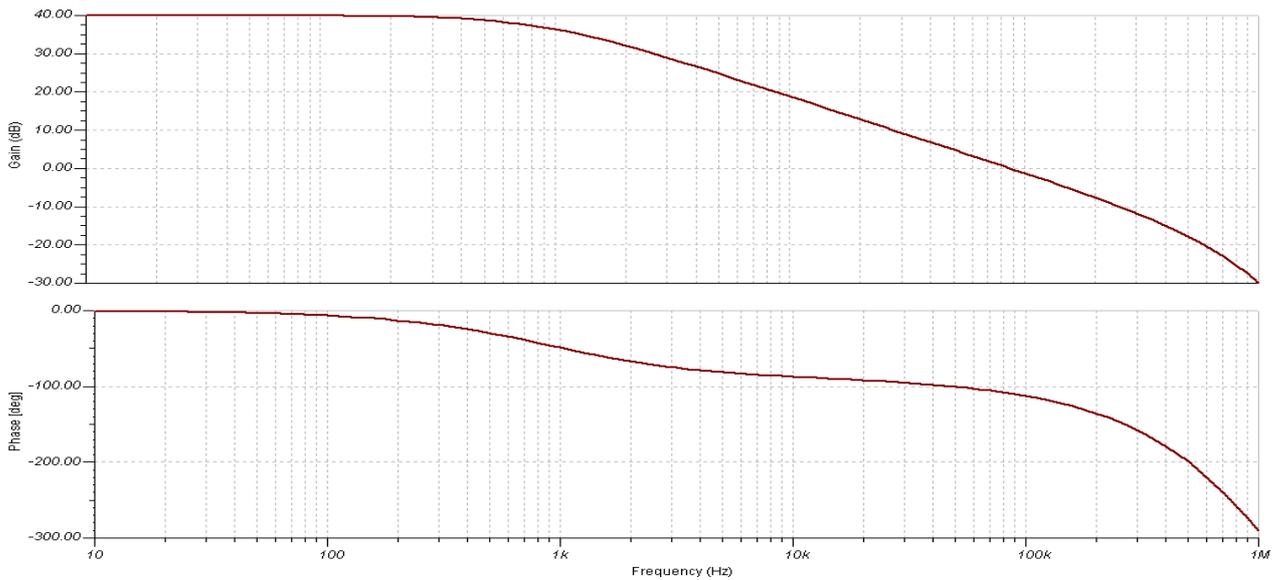


Figura 3.25. Diagrama de bode de la etapa amplificadora

### 3.7.2 Simulación de los estabilizadores de tensión del puente

Para la simulación del diseño de la fuente que alimenta el puente y genera la tensión de referencia de 3v, se utilizó el circuito de la figura 3.26. Para esta simulación, se fue variando el valor de alimentación V1, medida con el voltímetro VM1, se verificó que la tensión de alimentación del puente (6v), medida mediante el voltímetro VM2 y la tensión de referencia (3v), medida con el voltímetro VM3, permanecieran constantes. La imagen 3.27 muestra los resultados de la simulación para unas tensiones de entrada de 12v y 15v.

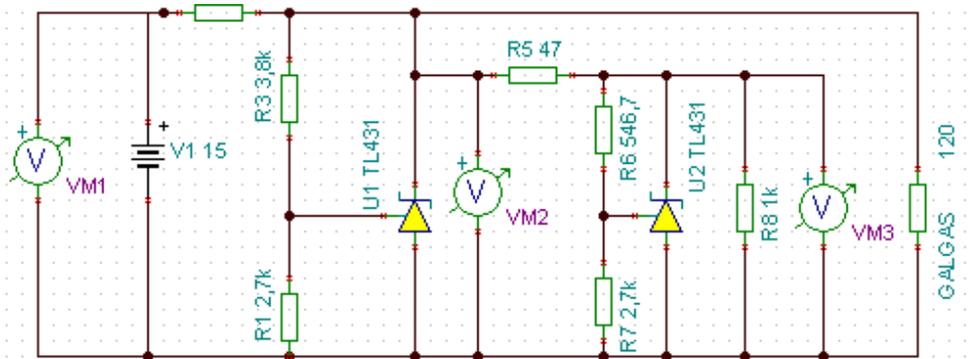


Figura 3.26. Circuito de simulación en Tina-ti.

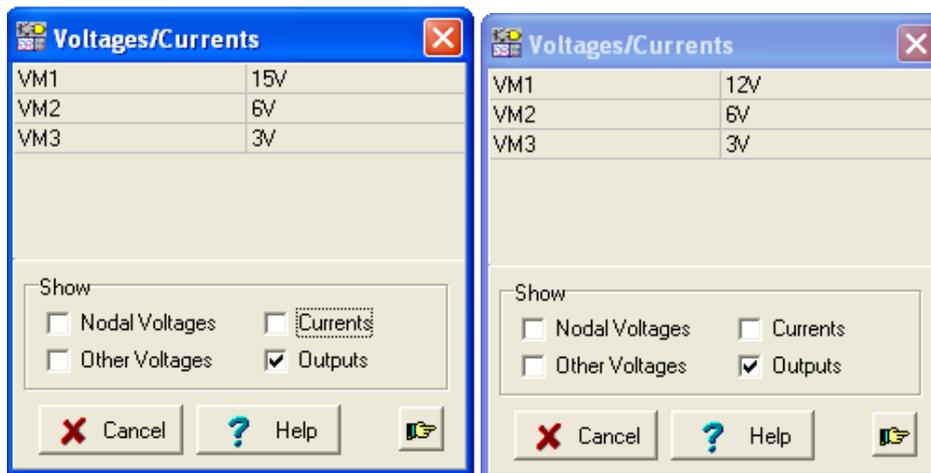


Figura 3.27. Resultados de la simulación

## Capítulo 4 .Diseño del bloque discriminador

En este capítulo, veremos el diseño del bloque digital, éste estará compuesto por un microcontrolador que realiza los cálculos necesarios para el funcionamiento del dispositivo y por un módulo de comunicaciones que se utiliza para la configuración inicial del dispositivo y adicionalmente si se sobrepasan los valores de mantenimiento o de seguridad lanzará mensajes de aviso.

### 4.1 El microcontrolador

Las necesidades de hardware que debe disponer el microcontrolador encargado de la etapa discriminadora son las siguientes:

- 2 canales de conversión analógico – digital
- 1 módulo de comunicaciones *Uart*
- 3 líneas de entrada - salida para uso general

En base a estas necesidades, se ha seleccionado el microcontrolador de la casa Microchip 18F4550 [17], aunque este microcontrolador supera con creces las necesidades de hardware, se ha seleccionado este modelo por la experiencia adquirida en años anteriores y disponer de herramientas de desarrollo para este modelo.

Para que este microcontrolador pueda funcionar, los únicos componentes externos necesarios son los referentes al oscilador. Para este punto se ha seleccionado un cristal oscilador de 4Mhz y por motivos funcionales se ha introducido un pulsador para poder resetear el microcontrolador.

El esquema completo del bloque discriminador se puede encontrar en el Anexo A.

### 4.2 Comunicaciones

Tal y como ya se comentó en la introducción del capítulo 1, este dispositivo dispone de un módulo de comunicaciones para poder configurar el dispositivo en función de las características del emplazamiento de la aplicación.

Para este apartado se han desarrollado 2 posibles vías de comunicación:

- Mediante una conexión serie RS232 por cable
- Mediante una conexión serie vía *Bluetooth*

El 18F4550, solo dispone de un módulo de comunicaciones *Uart* [18], por lo que solo se podrá establecer comunicación mediante una de las dos vías mencionadas (cable o *Bluetooth*). Durante el diseño del PCB (capítulo 6), se desarrollará una solución, la cual solo permitirá el montaje de una de las dos opciones.

#### 4.2.1 Comunicación cableada RS232

Tal y como se ha dicho en el punto 4.1, el microcontrolador solo dispone de un módulo *Uart*, las líneas del bus (RX y TX) toman valores de tensión TTL, lo cual quiere decir que un cero lógico está representado por 0v y un uno lógico está representado por 5v.

El protocolo de comunicación RS232 [19], disponible en cualquier ordenador personal, establece que un cero lógico está representado por valores de tensiones positivas que pueden llegar a alcanzar los 15v, mientras que para un uno lógico, los valores de tensión son negativos pudiendo llegar a los -15v.

Para poder llevar a cabo la comunicación entre el dispositivo y el puerto serie de un ordenador personal necesitamos que el dispositivo disponga de un *transceiver* que adapte las tensiones. Para realizar esta función se ha seleccionado *transceiver* DS275

[20], aunque existen soluciones más económicas como podría ser el *transceiver* MAX232, se ha seleccionado el DS275 dado que no requiere de ningún componente externo para su funcionamiento.

El esquema del *transceiver* DS275, se puede ver en el Anexo A.

## 4.2.2 Comunicación por Bluetooth

Para poder realizar una comunicación por Bluetooth con cualquier dispositivo es necesario que la aplicación disponga de un módulo para tal efecto. Para este caso, se ha seleccionado el módulo Bluetooth GP-GC021 [21] de la empresa Sure electronics, podemos ver el módulo en la figura 4.1. El GP-GC021 es un módulo Bluetooth con perfil SPP (*Serial Port Profile*) [22], dicho de otra manera la información que se envía y se recibe vía radio está disponible en los terminales Rx y Tx del módulo.



Figura 4.1. Módulo comunicaciones Bluetooth

El esquema de conexionado de este módulo puede verse en el Anexo A.

## 4.3 El convertor analógico digital

El microcontrolador 18F4550, dispone de un solo convertor analógico digital de 10 bits y 13 entradas o canales (AN0...AN12) que pueden ser seleccionadas para realizar la conversión. Para esta aplicación se seleccionarán los canales AN0 y AN1 para realizar la lectura de la tensión de salida del circuito de amplificación.

Las tensiones de referencia del convertor ( $V_{ref+}$  y  $V_{ref-}$ ) pueden ser internas, en cuyo caso se tomarán los valores de alimentación del microcontrolador (3,3v) o por otro lado podemos darle las tensiones de referencia desde una fuente externa al microcontrolador. Los canales AN2 y AN3 pueden ser configurados para establecer las tensiones de referencial máxima y mínima del convertor analógico digital.

Según se calculó en la ecuación 3.31 la tensión máxima de salida del circuito de amplificación, para el ejemplo de la viga, es de 5.52v, en este caso esta tensión es superior a la tensión de alimentación del microcontrolador, por lo que suministraremos una tensión de referencia externa  $V_{ref+}$  al convertor analógico digital. Esta tensión de referencia será de 6v y se aprovecha la misma línea de alimentación del puente de *Wheatstone*, la cual se estableció en 6v. En cuanto a la tensión de referencia  $V_{ref-}$  la fijaremos a 0v.

En este punto hemos establecido que la máxima conversión que podrá realizar el dispositivo será de 6v, teniendo en cuenta que el convertor analógico digital es de 10 bits, una tensión de 6v tendrá una representación binaria "111111111", que corresponde con el valor decimal 1023, por tanto cada salto equivaldrá a una tensión de:

$$V = \frac{V_{ref+} - V_{ref-}}{1024} = \frac{6v}{1024} = 0,005859V \quad (4.1)$$

Con la terminación de este capítulo se dan por alcanzados los objetivos cuarto y quinto de este TFC.

## Capítulo 5 .Programación de microcontrolador

En este capítulo, se describen las herramientas necesarias para el diseño del *software* que controlará el dispositivo, posteriormente se describe el diagrama de flujo que debe seguir el *software* de control, se hace mención al *software* desarrollado y se realiza la validación del software.

### 5.1 Entorno de desarrollo MPLAB

El desarrollo del *software* del microcontrolador se realiza mediante el entorno de desarrollo MPLAB [23], que es suministrado gratuitamente por el fabricante del microcontrolador. El entorno MPLAB ofrece la posibilidad de desarrollar el *software* en lenguaje de bajo nivel ASM. La intención es desarrollar el *software* en un lenguaje de alto nivel como C, por lo que además de instalar el IDE MPLAB es necesario instalar el compilador C18 [24], el cual también es suministrado por el propio fabricante como una versión demo, que es funcional los 60 primeros días. En la figura 5.1 se puede ver la interfaz de MPLAB para este proyecto.

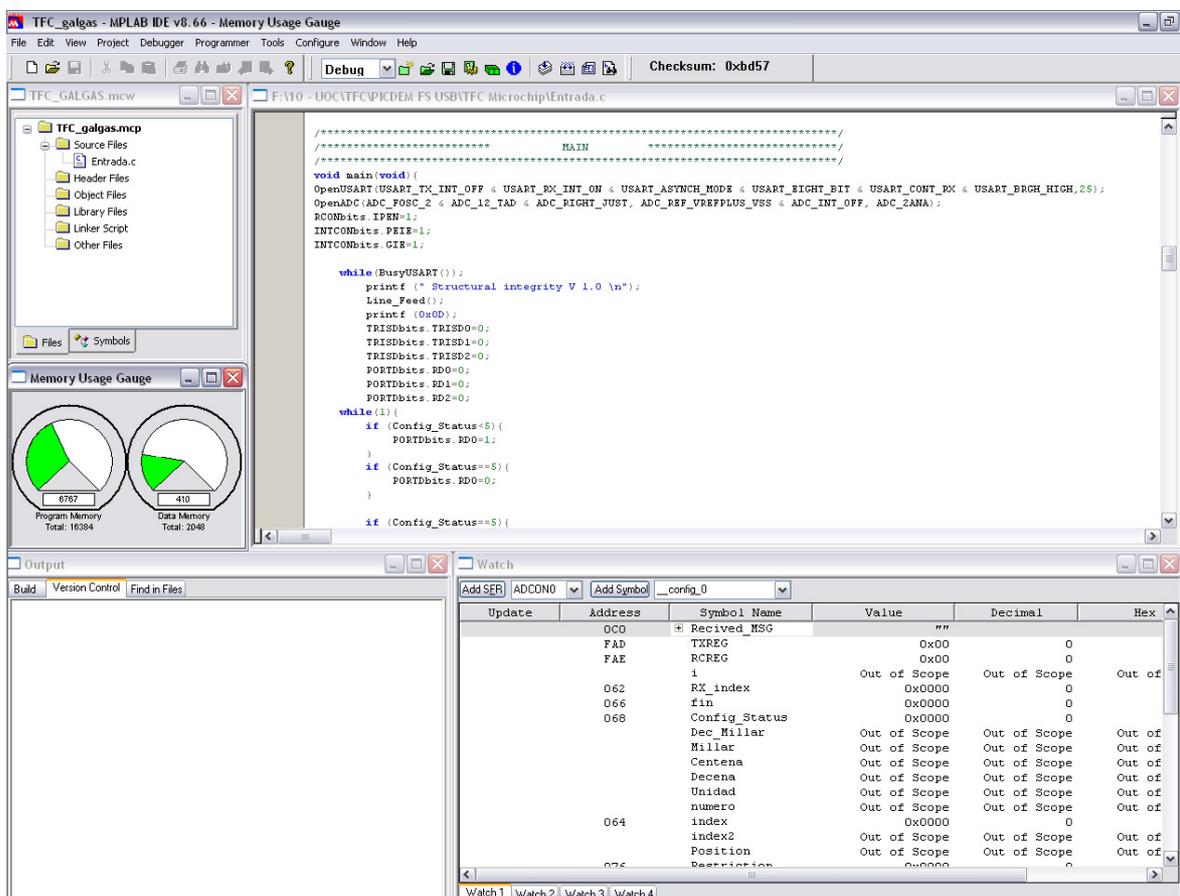


Figura 5.1. Entorno de desarrollo MPLAB

### 5.2 Diagrama de flujo de la aplicación

El desarrollo del *software* de esta aplicación está dividido en tres secciones:

- Configuración del microcontrolador
- Parametrización del dispositivo
- Bucles de lectura

### 5.2.1 Configuración del microcontrolador

Para la configuración del microcontrolador se establecen los siguientes parámetros: Oscilador de 4Mhz, las líneas 0,1 y 2 del Port D se tienen que configurar como salidas digitales. Se configura la *Uart* para trabajar a 9600 bps en modo asíncrono y se habilita el *flag* de interrupción en la recepción. Se habilita el conversor analógico digital y se establece la tensión de referencia  $V_{ref+}$  como una fuente externa y  $V_{ref-}$  referenciada a la propia masa del micro. Una vez se ha configurado el microcontrolador, éste se queda a la espera de ser parametrizado. La figura 5.2 muestra el diagrama de flujo que sigue el *software* de esta aplicación.

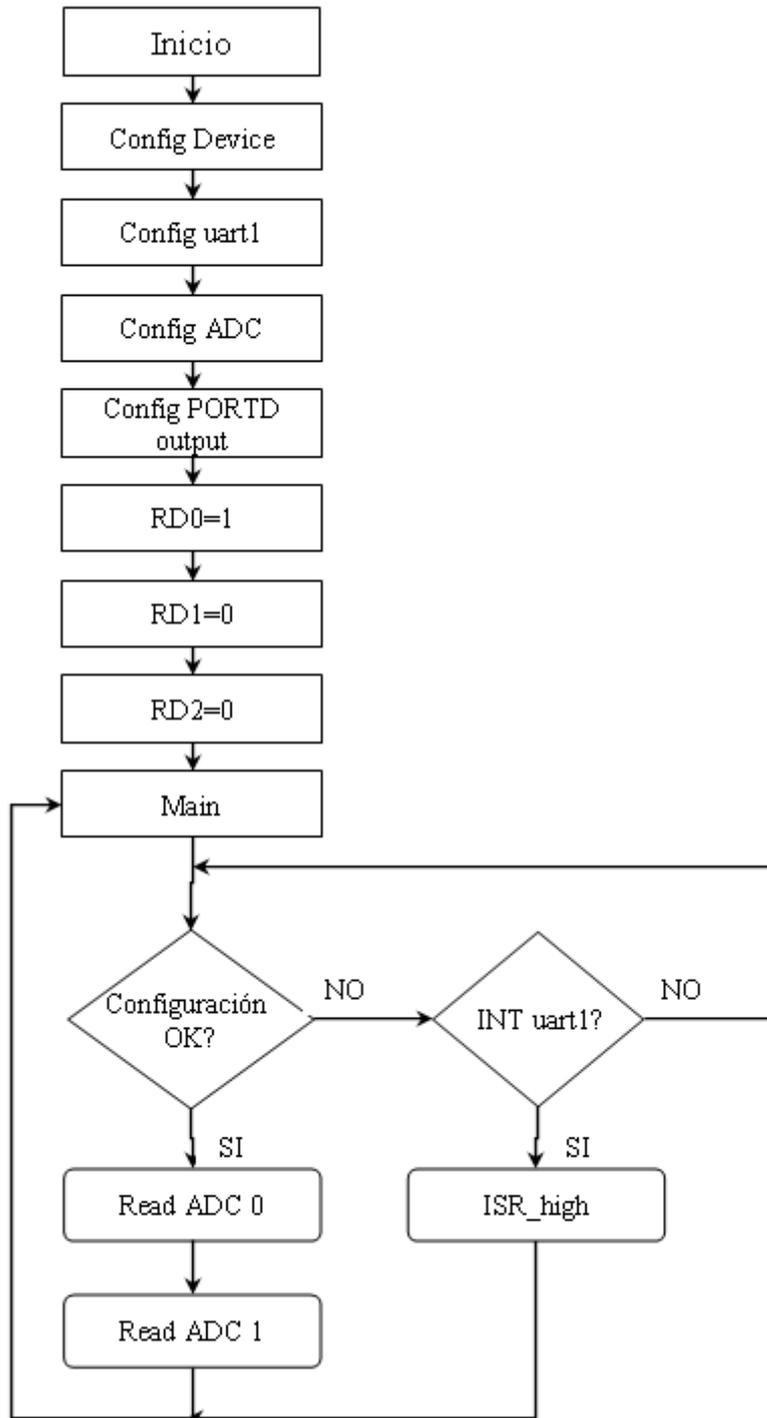


Figura 5.2. Diagrama de flujo de la configuración del microcontrolador

## 5.2.2 Parametrización del dispositivo

Para poder indicarle al dispositivo los parámetros necesarios para poder funcionar, es necesario iniciar una comunicación a través del puerto serie del ordenador. Una vez se inicia el proceso, el dispositivo nos irá preguntando los diferentes valores necesarios: Longitud, Limitación y el valor para el porcentaje de mantenimiento. Con todos estos datos, el dispositivo realiza los cálculos necesarios y realiza una lectura de cada canal, esta lectura la tomaremos como la *offset*. Si el proceso de Parametrización ha sido satisfactorio, el led conectado al terminal RD0 se apagará (Ver esquema en Anexo A), de lo contrario permanecerá encendido indicando que falta algún parámetro y el sistema no está funcionando. En la figura 5.3 se puede ver el diagrama de flujo para esta aplicación.

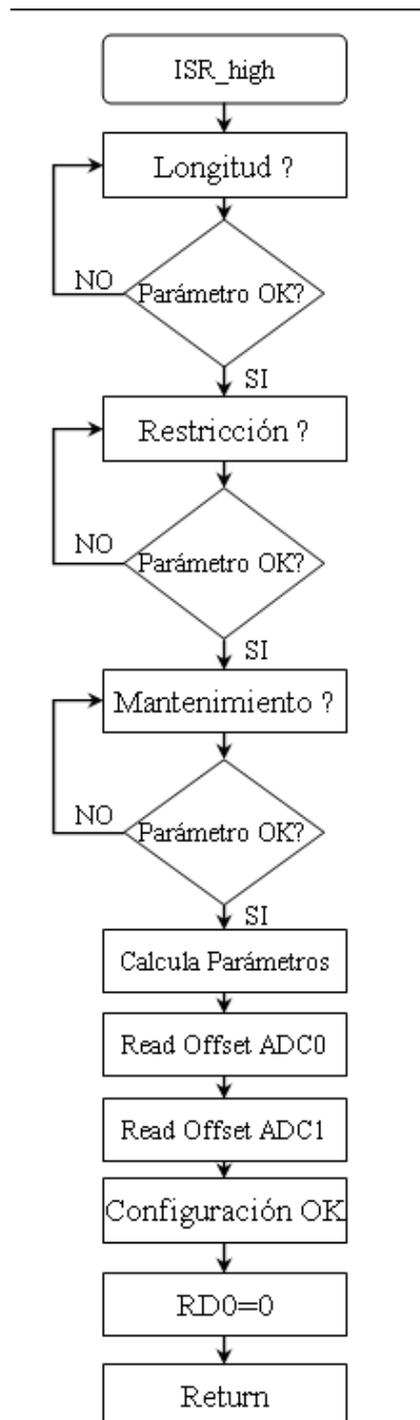


Figura 5.3. Diagrama de flujo para la parametrización del dispositivo

### 5.2.3 Bucles de lectura

Una vez el dispositivo está configurado y parametrizado, el *software* entra en bucle donde únicamente tiene que estar leyendo los dos canales y comparando el valor de estos canales con los valores calculados en el punto anterior. Si algún valor de las lecturas es superior a los parámetros calculados en el punto anterior, se activará la alarma pertinente, visual o acústica. Adicionalmente a través de la *Uart* del microcontrolador se lanza un mensaje de alarma indicando que tipo de alarma se ha activado, de mantenimiento o de seguridad. En la figura 5.4 se puede ver el diagrama de flujo del bucle.

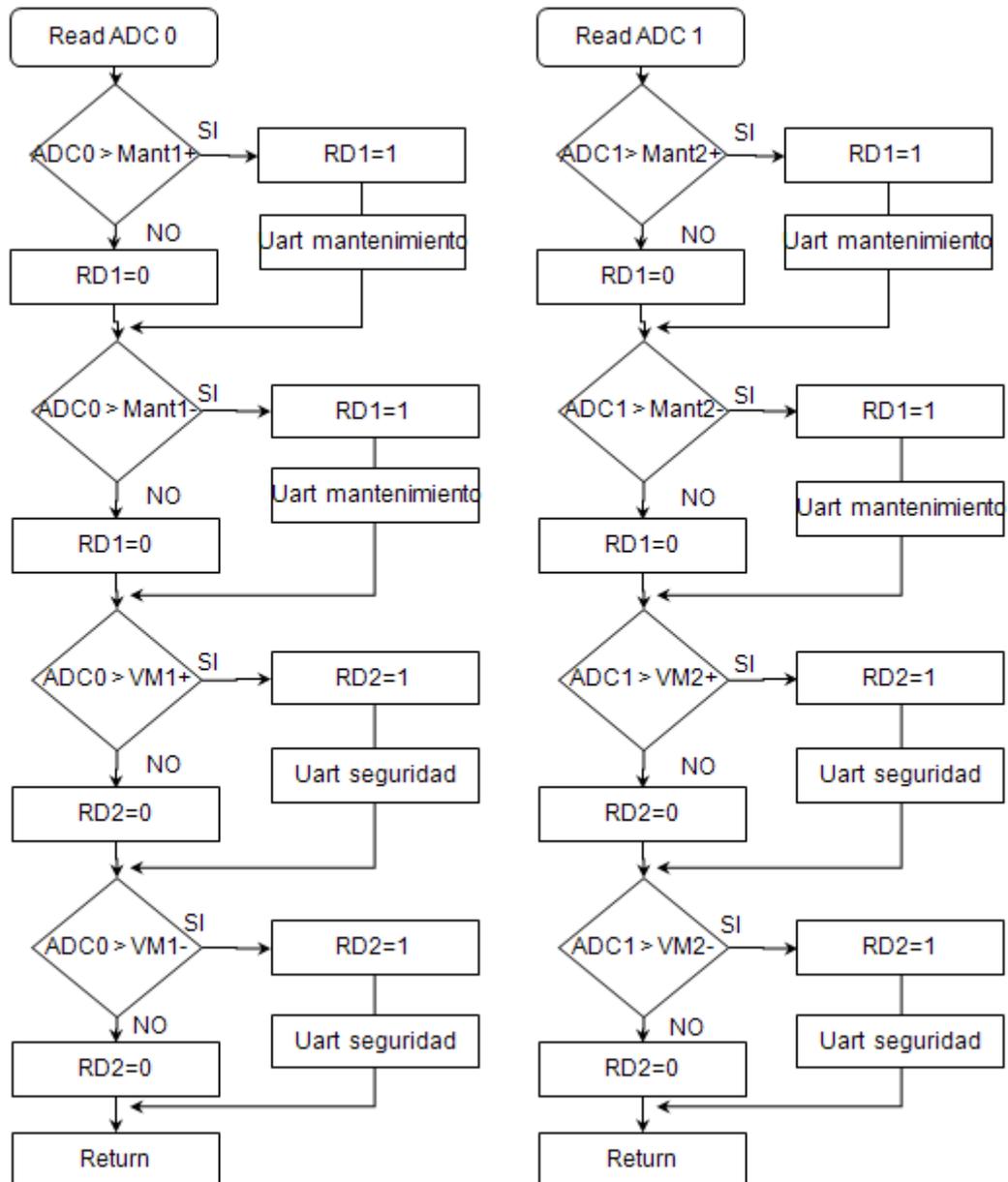


Figura 5.4. Diagrama de flujo para el bucle de funcionamiento

## 5.3 Programación en código C

Tal y como se ha comentado en el apartado 5.1, el desarrollo del *software* se ha realizado en el entorno de desarrollo MPLAB y el compilador C18. La utilización de este compilador permitió la utilización de librerías para controlar el microcontrolador 18F4550. En las imágenes 5.5 y 5.6 se puede ver un extracto del *software* donde se observan las directivas para incluir las librerías necesarias y un ejemplo de la utilización de las mismas para la configuración de la *Uart* y el conversor analógico digital.

```
#include <p18f4550.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <adc.h>
//#include <delays.h>
//#include <timers.h>
#include <usart.h>

#pragma config PLLDIV = 1, CPUDIV = OSC1_PLL2, USEDIV = 2, FOSC = XT_XT, FCMEN = OFF, IESO = OFF, PWRT = ON
#pragma config BOR = SOFT, BORV = 2, VREGEN = ON, WDT = OFF, WDTPS = 128, MCLRE = ON, LPT1OSC = OFF, PBAEN = OFF
#pragma config CCP2MX = ON, STVRN = ON, LVP = OFF, ICPR1 = OFF, XINST = OFF, DEBUG = OFF
#pragma config CPO = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF, CPB = OFF, CPD = OFF
#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF, WRTE = OFF, WRTC = OFF, WRTD = OFF
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF, EBTR4 = OFF, EBTR5 = OFF
```

Figura 5.5. Librerías C18 y configuración del oscilador

```
/*
*****
*****          MAIN          *****
*****
*/
void main(void){
OpenUSART(USART_TX_INT_OFF & USART_RX_INT_ON & USART_ASYNC_MODE & USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH,25);
OpenADC(ADC_FOSC_2 & ADC_12_TAD & ADC_RIGHT_JUST, ADC_REF_VREFPLUS_VSS & ADC_INT_OFF, ADC_2ANA);
RCONbits.IPEN=1;
INTCONbits.PEIE=1;
INTCONbits.GIE=1;
```

Figura 5.6. Configuración de la Uart y el conversor analógico digital

El código completo se puede encontrar en el Anexo B.

## 5.4 Validación del software

Para la validación del *software* se utilizó una tarjeta de desarrollo del fabricante Microchip, modelo PICDEM™ FS-USB, para simular el circuito de medición y amplificación se instalaron dos potenciómetros de ajuste y para las comunicaciones, se le instaló el módulo *bluetooth* GP-GC021. La imagen 5.7 muestra una fotografía del montaje realizado para la verificación del software. En la imagen 5.8 se puede ver la comunicación con el ordenador para la parametrización del dispositivo.

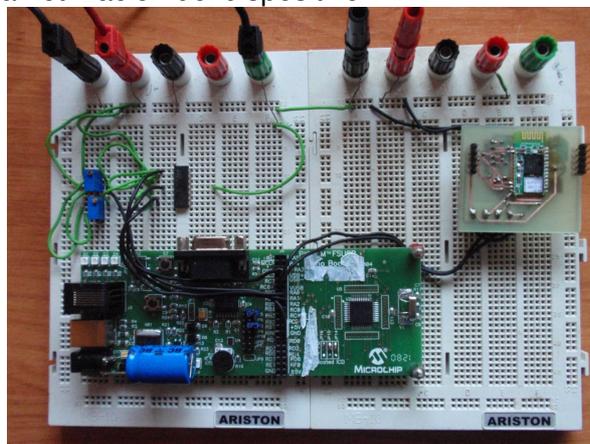
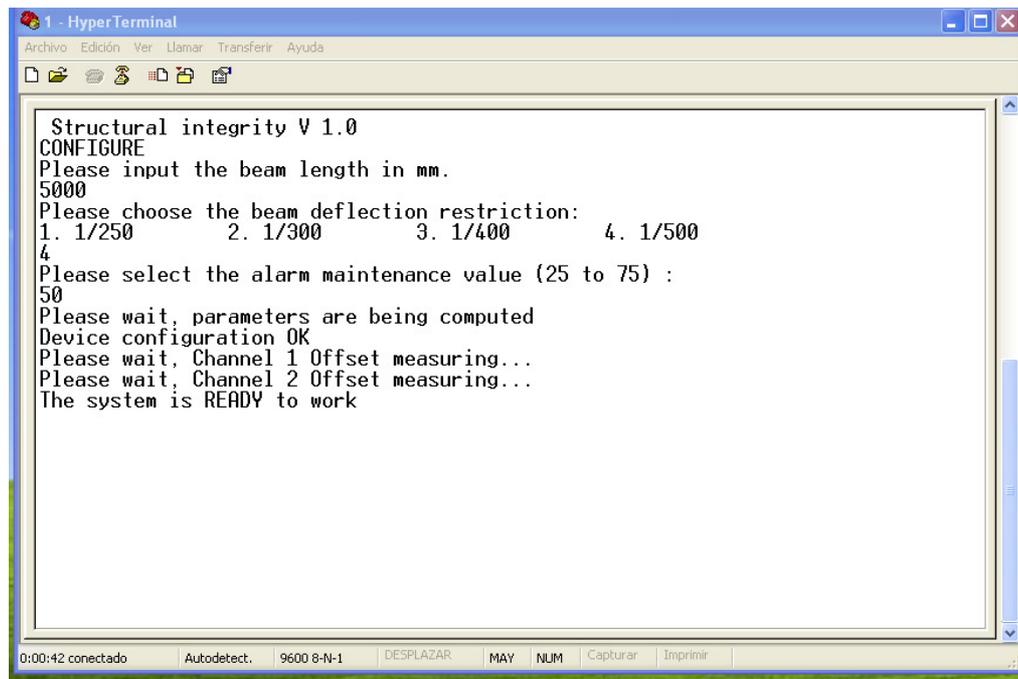


Figura 5.7. Montaje para la verificación del software



```

1 - HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda

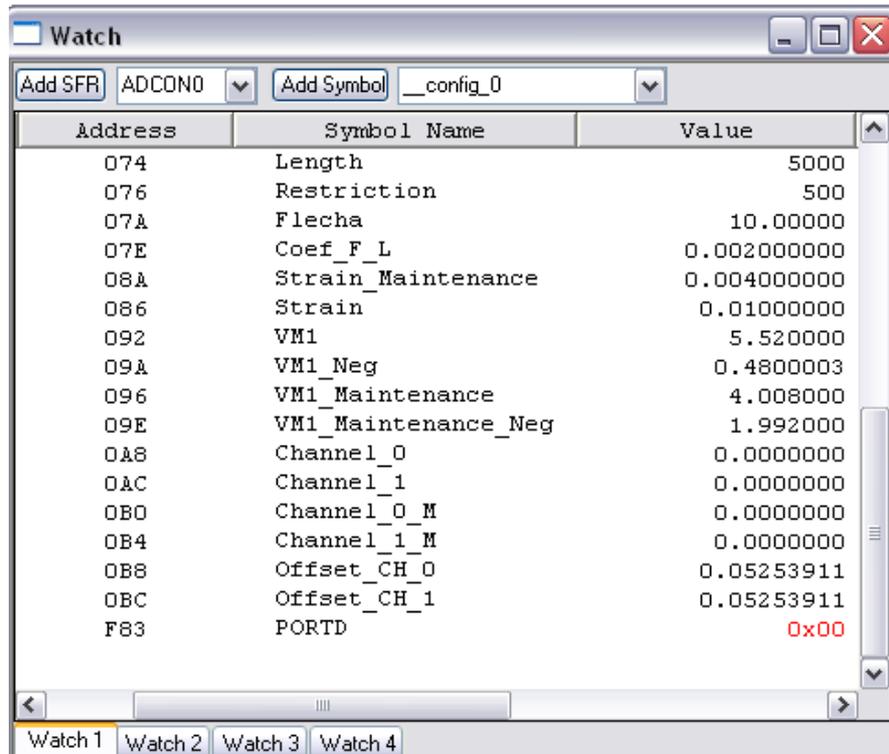
Structural integrity V 1.0
CONFIGURE
Please input the beam length in mm.
5000
Please choose the beam deflection restriction:
1. 1/250      2. 1/300      3. 1/400      4. 1/500
4
Please select the alarm maintenance value (25 to 75) :
50
Please wait, parameters are being computed
Device configuration OK
Please wait, Channel 1 Offset measuring...
Please wait, Channel 2 Offset measuring...
The system is READY to work

0:00:42 conectado  Autodetect.  9600 8-N-1  DESPLAZAR  MAY  NUM  Capturar  Imprimir

```

**Figura 5.8. Parametrización del dispositivo mediante Hyperterminal**

Para la verificación del *software* antes de comenzar la parametrización, se aplicó una tensión de 3v a los terminales AN0 y AN1 (canales 0 y 1 del convertor analógico digital) para simular la salida del módulo de amplificación. Como se puede ver en la imagen 5.8, el dispositivo hace una lectura de estos dos canales y resta 3 para calcular la *offset*, estos valores se pueden visualizar en la imagen 5.9.



Address	Symbol Name	Value
074	Length	5000
076	Restriction	500
07A	Flecha	10.00000
07E	Coef_F_L	0.002000000
08A	Strain_Maintenance	0.004000000
086	Strain	0.010000000
092	VM1	5.520000
09A	VM1_Neg	0.4800003
096	VM1_Maintenance	4.008000
09E	VM1_Maintenance_Neg	1.992000
0A8	Channel_0	0.0000000
0AC	Channel_1	0.0000000
0B0	Channel_0_M	0.0000000
0B4	Channel_1_M	0.0000000
0B8	Offset_CH_0	0.05253911
0BC	Offset_CH_1	0.05253911
F83	PORTD	0x00

**Figura 5.9. Variables de la aplicación**

En la figura 5.9 se puede ver que la *offset* que ha medido el dispositivo es de 52mV, para ambos canales, además se pueden observar los valores calculados para la activación de las alarmas.

La variable VM1 indica la tensión positiva donde comienza la zona de seguridad, y que se corresponde según se calculó en la ecuación 3.31. VM1\_Neg indica la tensión mínima donde comienza la zona de seguridad si la deformación es en sentido contrario, y que se corresponde con los resultados de la ecuación 3.32. En la figura 3.16 se puede comprobar que estos valores corresponden al área de la deformación máxima.

VM1\_Maintenance y VM1\_Maintenance\_Neg, indica los valores de tensión, a partir de los que tiene que saltar la alarma de mantenimiento. Como se muestra en la imagen 5.8 el valor de mantenimiento escogido ha sido del 50%.

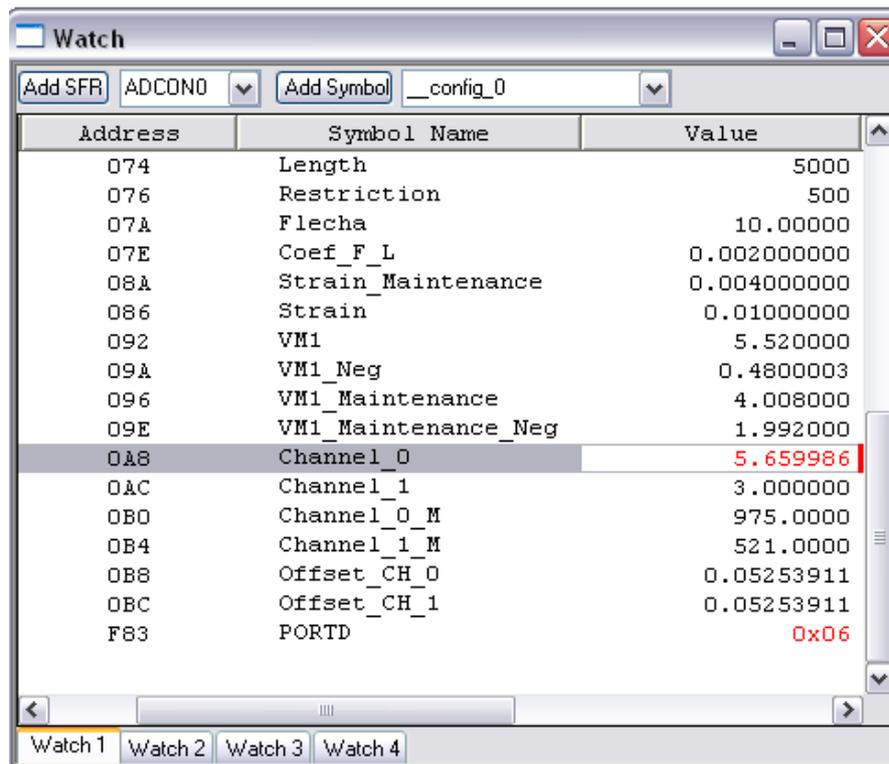
Las variables Chanel\_0 y Chanel\_1 muestran el valor de la tensión en la entrada de los conversores menos la *offset* medida cuando la aplicación entre en el bucle de trabajo.

Posteriormente al terminal AN0, se le aplicó una tensión de 4.2v, la cual al superar a la tensión de mantenimiento, hizo que se encendiera el led conectado al terminal RD1. El valor del Puerto D se refleja en el registro PORTD según se ve en la imagen 5.10 junto con el resto de variables.

Address	Symbol Name	Value
074	Length	5000
076	Restriction	500
07A	Flecha	10.00000
07E	Coef_F_L	0.002000000
08A	Strain_Maintenance	0.004000000
086	Strain	0.01000000
092	VM1	5.520000
09A	VM1_Neg	0.4800003
096	VM1_Maintenance	4.008000
09E	VM1_Maintenance_Neg	1.992000
0A8	Channel_0	4.195236
0AC	Channel_1	3.000000
0B0	Channel_0_M	725.0000
0B4	Channel_1_M	521.0000
0B8	Offset_CH_0	0.05253911
0BC	Offset_CH_1	0.05253911
F83	PORTD	0x02

**Figura 5.10. Variables de la aplicación**

La tensión suministrada al Terminal AN0 se fue incrementando hasta alcanzar una tensión de 5,6v, que hizo saltar la alarma de seguridad, conectada al Terminal RD2 del puerto D. En la imagen 5.11 se pueden ver las variables de esta verificación.

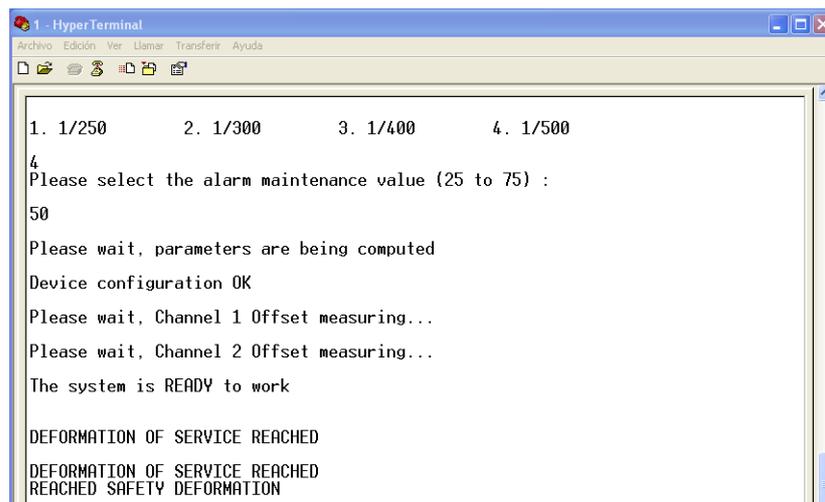


Address	Symbol Name	Value
074	Length	5000
076	Restriction	500
07A	Flecha	10.00000
07E	Coef_F_L	0.002000000
08A	Strain_Maintenance	0.004000000
086	Strain	0.01000000
092	VM1	5.520000
09A	VM1_Neg	0.4800003
096	VM1_Maintenance	4.008000
09E	VM1_Maintenance_Neg	1.992000
0A8	Channel_0	5.659986
0AC	Channel_1	3.000000
0B0	Channel_0_M	975.0000
0B4	Channel_1_M	521.0000
0B8	Offset_CH_0	0.05253911
0BC	Offset_CH_1	0.05253911
F83	PORTD	0x06

**Figura 5.11. Variables de la aplicación**

Actuando de la misma manera se verificó el correcto funcionamiento de las alarmas para las zonas de mantenimiento y seguridad opuestas a la comentada. Tras verificar el correcto funcionamiento del canal 0, se verificó el correcto funcionamiento del canal 1, para ello se procedió de la misma manera, obteniendo los mismos resultados.

Como ya se comentó en la introducción del capítulo 4, el dispositivo tiene que lanzar un mensaje a través de la *Uart*, indicando que tipo de deformación se ha alcanzado. En la imagen 5.12 se puede ver los mensajes de alarma lanzados por el dispositivo al saltar alguna alarma.



```

1. 1/250      2. 1/300      3. 1/400      4. 1/500
4
Please select the alarm maintenance value (25 to 75) :
50
Please wait, parameters are being computed
Device configuration OK
Please wait, Channel 1 Offset measuring...
Please wait, Channel 2 Offset measuring...
The system is READY to work

DEFORMATION OF SERVICE REACHED
DEFORMATION OF SERVICE REACHED
REACHED SAFETY DEFORMATION

```

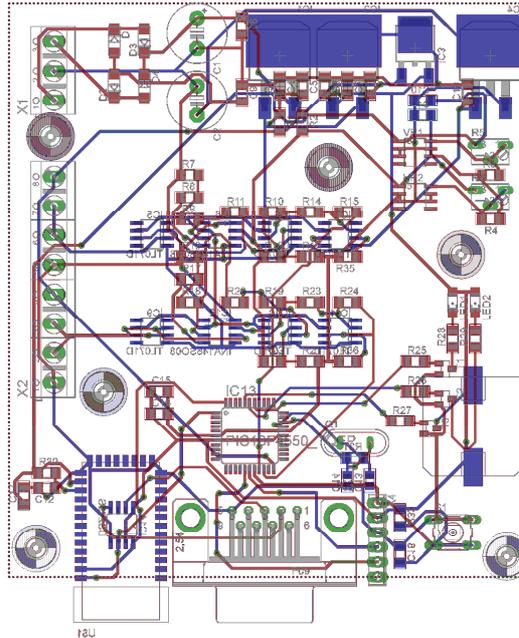
**Figura 5.12. Mensajes de alarma**

Mediante las pruebas realizadas, se ha podido ver que el *software* diseñado cumple con las especificaciones iniciales, por lo que se da por alcanzado el sexto objetivo de este TFC.

## Capítulo 6 .Diseño del circuito impreso

El diseño del circuito impreso se ha realizado mediante el *software* “Eagle” de la empresa CadSoft. El esquema final del dispositivo se puede encontrar en el anexo A. Para el diseño de este dispositivo se ha optado por la utilización de componentes de montaje superficial [25] “SMD”, el circuito impreso es de doble cara y los componentes están insertados en ambas caras del circuito.

Tal y como se comentó en el apartado 4.2, este dispositivo puede comunicarse mediante cable a través del transceiver DS275 o mediante *bluetooth*. El circuito impreso diseñado, ver figura 6.1, es un circuito polivalente en el que se pueden montar ambas configuraciones, pero tiene que tener en cuenta que no se deben montar ambas configuraciones a la vez.



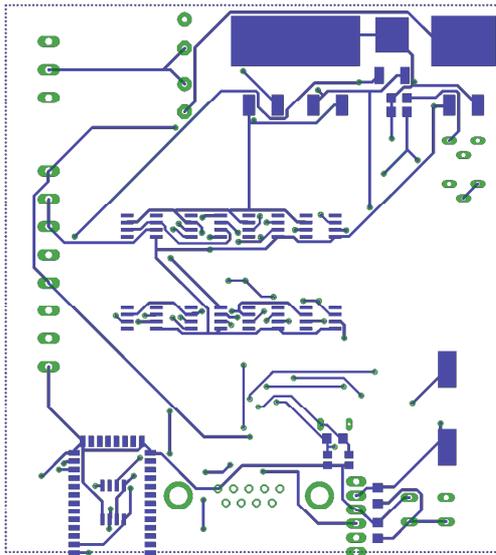


Figura 6.3a. Layout de pistas superficie Bottom

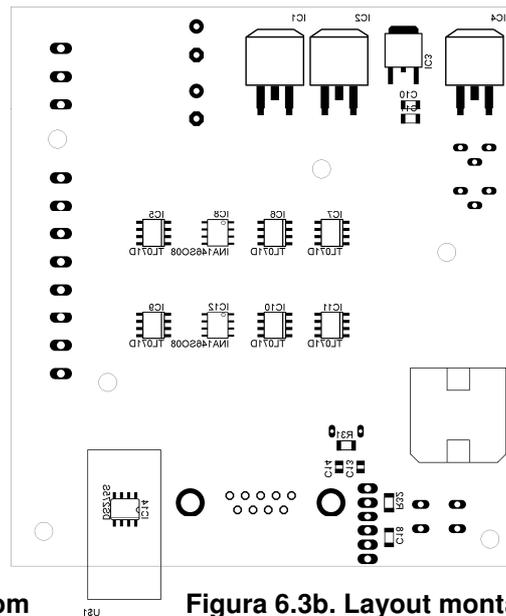


Figura 6.3b. Layout montaje

Para evitar posibles errores durante el montaje de los componentes relacionados con las comunicaciones, DS275 y GP-GC021, se ha solapado la ubicación de ambos componentes, siendo imposible montar ambas configuraciones en un mismo circuito.

Con el fichero del diseño del circuito impreso (archivo.BRD), cualquier fabricante de circuitos impresos puede fabricar esta placa.

Con el diseño del circuito impreso se da por alcanzado el séptimo objetivo establecido para este TFC.

### 6.1 Listado de componentes

Como es de esperar, debido a la polivalencia del circuito, el listado de materiales se divide en 3 tablas: Componentes comunes, Componentes comunicación por cable y Componentes comunicación BT.

Tabla 6.1. Tabla de componentes comunes

Componentes comunes		
Componente	Valor	Encapsulado
C1	470uF	E5-10,5
C2	470uF	E5-10,5
C3	100nF	C1206
C4	100nF	C1206
C5	100nF	C1206
C6	100nF	C1206
C7	100nF	C1206
C8	100nF	C1206
C9	100nF	C1206
C10	100nF	C1206
C11	100nF	C1206
C13	22pF	C0805
C14	22pF	C0805
C15	100nF	C1206
C16	100nF	C1206
C18	100nF	C1206
C19	100nF	C1206
D1	1n5817	DO214AC
D2	1n5817	DO214AC

D3	1n5817	DO214AC
D4	1n5817	DO214AC
IC1	LM7815	D2PACK
IC2	LM7805	D2PACK
IC3	LF33C	DPACK
IC4	LM7915	D2PACK
IC5	TL071D	SO08
IC6	TL071D	SO08
IC7	TL071D	SO08
IC8	INA146SO08	SO08
IC9	TL071D	SO08
IC10	TL071D	SO08
IC11	TL071D	SO08
IC12	INA146SO08	SO08
IC13	PIC18F4550_TQFP	TQFP44
LED1	LEDCHIPLED_1206	CHIPLED_1206
LED2	LEDCHIPLED_1206	CHIPLED_1206
Q1	4.00Mhz	QS
R1	47	R1206
R2	47	R1206
R3	2k7	R1206
R4	2k7	R1206
R5	10k	B64W
R6	1k	B64W
R7	100	R1206
R8	10k	R1206
R9	39k9	R1206
R10	10k	R1206
R11	10k	R1206
R12	10k	R1206
R13	10k	R1206
R14	10k	R1206
R15	10k	R1206
R16	100	R1206
R17	10k	R1206
R18	39k9	R1206
R19	10k	R1206
R20	10k	R1206
R21	10k	R1206
R22	10k	R1206
R23	10k	R1206
R24	10k	R1206
R25	24k	R1206
R26	24k	R1206
R27	110k	R1206
R28	1k3	R1206
R29	1k3	R1206
R31	1M	R1206
R32	10k	R1206
R35	10k	R1206
R36	10k	R1206
S1	10-XX	B3F-10XX
SG1	F/BC40	BUZZER_SMD
T1	BC847	SOT23-BEC
T2	BC847	SOT23-BEC
T3	BC847	SOT23-BEC
VR1	TL431CLP	SO08
VR2	TL431CLP	SO08
X1	AK500/3	AK500/3
X2	AK500/8	AK500/8
X4	MPT6	6POL254

**Tabla 6.2 Tabla de componentes comunicación por cable**

Componentes comunicación por cable		
Componente	Valor	Encapsulado
X3	F09HP	F09HP
IC14	DS275S	SO08

**Tabla 6.3 Tabla de componentes comunicación por BT**

Componentes comunicación BT		
Componente	Valor	Encapsulado
U\$1	SURE_BT_GP_GC021	SURE_BT_MODULE
C12	100nF	C1206
R30	10k	R1206
C17	100nF	C1206

## Capítulo 7 .Valoración económica

En este capítulo se realiza una valoración económica de todo el proyecto, desde el inicio hasta la entrega del prototipo. Para ello los costes están tabulados en 3 áreas:

- Costes de ingeniería, estos costes comprenden los trabajos derivados del diseño y gestión documental del proyecto. Se ha establecido un precio simbólico de 30€ hora.
- Coste de materiales, en estos costes se contempla el coste de los componentes para la realización del prototipo.
- Gastos de mano de obra, en estos costes, se incluyen los gastos para la fabricación y montaje del prototipo.

Costes de ingeniería 165 h. x 30€/h 4950 €

Coste de materiales

- Materiales comunes (4 galgas incluidas) 39,87€
- Opción comunicación por cable 4,40€
- Opción comunicación por BT 15,11€

Mano de obra

- Tooling PCB 90x100 mm (solo 1 vez) 35,70€
- PCB, FR4, doble cara (2 unidades) 60,20€
- Tooling dispensador estaño y componentes (solo 1 vez) 36,20€
- Montaje y soldadura de componentes 11,25€

La entrega final del proyecto constaría de los siguientes elementos:

- Memoria técnica descriptiva del proyecto
- 1 prototipo (comunicación por cable)
- 1 prototipo (comunicación por BT)

Total del proyecto: 5240,05€

## Capítulo 8 .Conclusiones

A lo largo de este TFC se ha intentado diseñar un dispositivo polivalente que pudiera ser emplazado en diferentes ubicaciones y a través de unos parámetros de configuración adaptar el dispositivo a las características específicas de la estructura a observar. Esta es una de las características que a mi juicio mas puedo valorar. Hoy en día la industria tiende a estandarizar cualquier aplicación o dispositivo teniendo una visión más global del área de aplicación, con esta política se consigue ahorrar en costes de desarrollo.

Aunque a priori esta política suena muy bien, en un mundo real, siempre existirán limitaciones en el uso y utilización de dispositivos a los que podríamos llamar “dispositivos polivalentes”. El dispositivo que se ha diseñado en este TFC, no está exento de limitaciones.

En el apartado 4.3 se vio el porqué la máxima tensión que debe suministrar el circuito de amplificación es de 6v, dado que la tensión de referencia del conversor analógico digital es de 6v. Si el amplificador supera esta tensión, la lectura no será correcta. Con esta idea en la cabeza, podemos pensar que un usuario podría llegar a configurar el dispositivo según sus necesidades y que los cálculos indicaran que la tensión para la zona de seguridad sea superior a 6v. Para corregir este problema, en el *software* de control ya se ha introducido unas líneas de código donde si la tensión calculada en función de los parámetros que nos indica el usuario supera los 6v, el dispositivo lanza una alarma avisándonos de que esta configuración no es posible. Para prevenir esta posible situación vale la pena calcular que tensión se tendrá a la salida del amplificador y así evitar posibles confusiones. La tensión a la salida del amplificador la podemos calcular aplicando la ecuación 8.1. Para el buen funcionamiento del dispositivo, el valor VM1 nunca deberá ser mayor de 6v:

$$VM1 = \left( \left( \left( 240GF \left( \frac{Luz - \frac{Coef_{Limitacion}(Luz)}{Coef_{Limitacion} + \frac{Coef_{Limitacion}}{100}}}{Luz} \right) \right) \right) - 3 \right) - 1 \quad (8.1)$$

Por otro lado, una funcionalidad extra a añadir al dispositivo sería la introducción de una memoria eeprom donde se fueran guardando un registro del avance en el tiempo de las deformaciones y poder recoger esta información mediante el puerto de comunicaciones para su posterior manipulación y estudio.

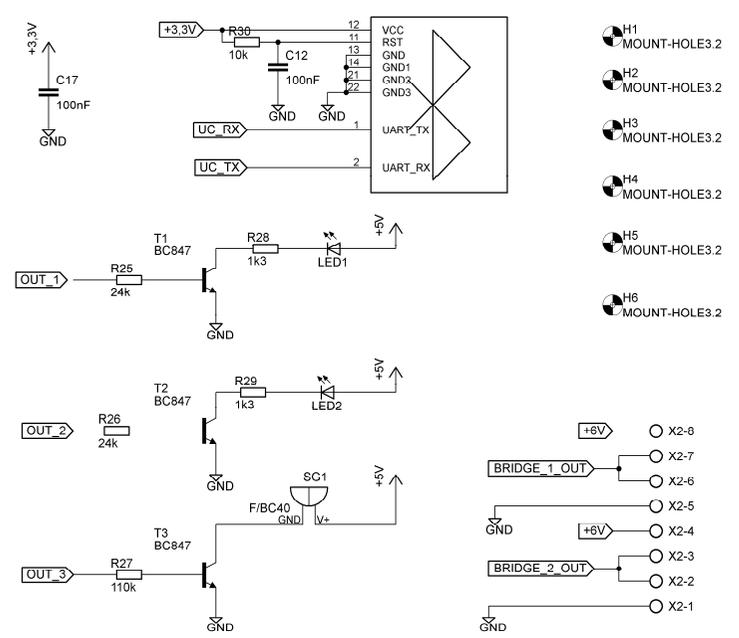
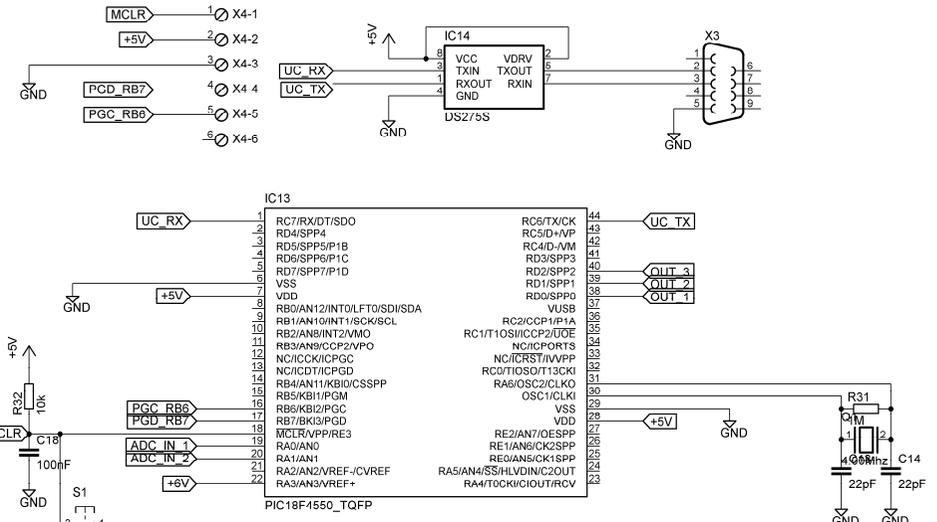
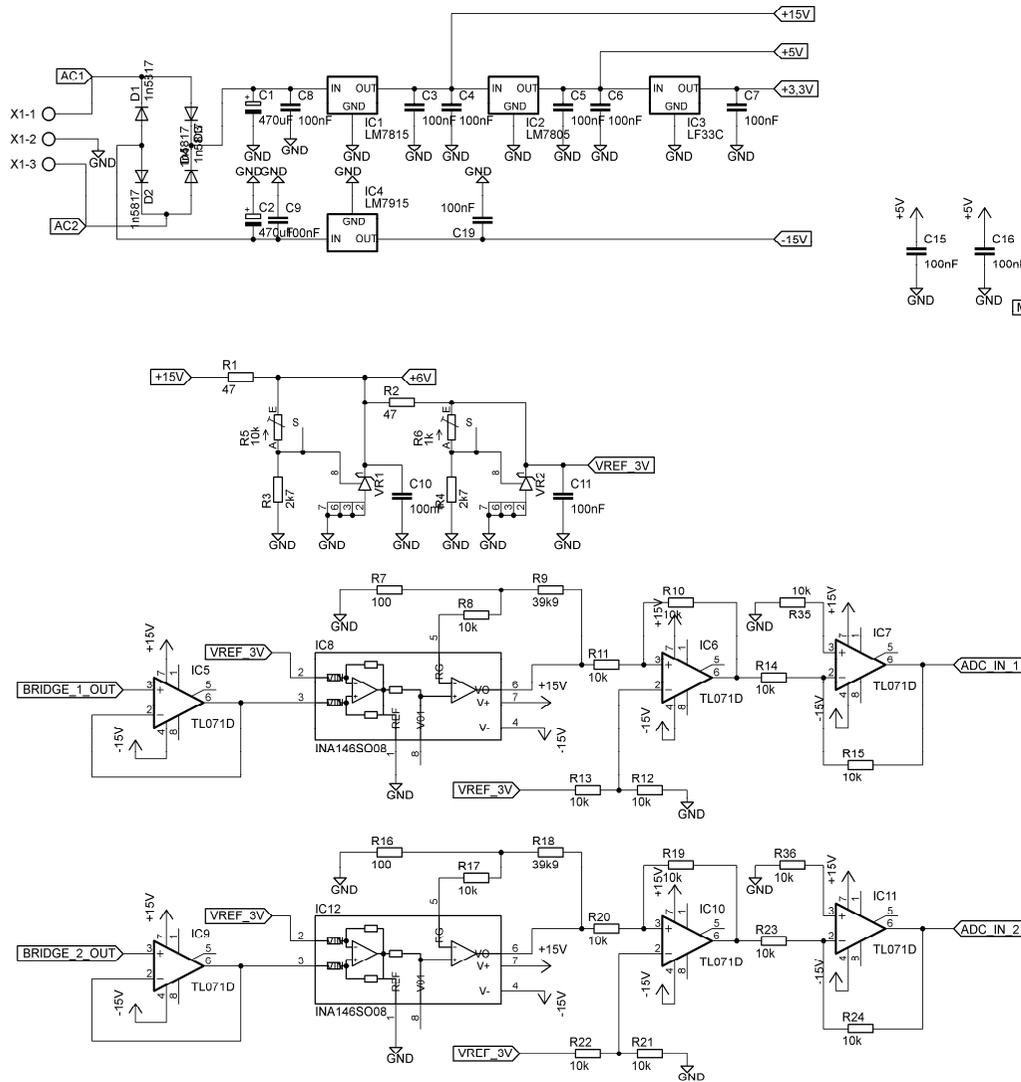
Durante la planificación de este TFC, se fijaron unos objetivos, que se han ido consiguiendo durante el desarrollo del TFC. Se ha utilizado una metodología de diseño y verificación planificando cuidadosamente los pasos a seguir para la consecución de los objetivos marcados. El resultado final es un producto del cual me siento muy orgulloso de haber desarrollado.

## Bibliografía

- [1] **Measuring Strain with Strain Gages**, <http://zone.ni.com/devzone/cda/tut/p/id/3642>  
artículo técnico publicado por National Instruments
- [2] Marcos Giménez Valentín, Jaime Riera Guasp, Ana Vidaurre Garayo, **Fundamentos electromagnéticos de la ingeniería**, Editorial UPV, ISBN 978-84-8363-063-1
- [3] Robert Boylestad, Louis Nashelsky, **Electrónica Teoría de Circuitos**, Editorial Prentice Hall, ISBN 968-880-347-2
- [4] A. Hugon, M. Serre, **Cálculos y ensayos, estudio de los proyectos**, ISBN 84-7146-193-5  
Version online  
<http://books.google.es/books?id=vXompjurA8gC&lpg=PP1&ots=wSwbFC3upg&dq=%22C%C3%A1lculos%20y%20ensayos%22%20%20A.%20Hugon%2CM.%20Serre&pg=PP1#v=onepage&q&f=false>
- [5] **Normas básicas de la edificación**,  
[http://normativaconstruccion.cype.info/1\\_02\\_a/pagina165.html](http://normativaconstruccion.cype.info/1_02_a/pagina165.html), Relación Flecha / luz
- [6] **Data Book PRECISION STRAIN GAGES**, <http://www.vishaypg.com/micro-measurements/stress-analysis-strain-gages/>, catálogo de galgas extensométricas de la empresa VISHAY MICRO-MEASUREMENTS
- [7] **Datasheet galga extensométrica N11MA512023**  
<http://es.rsonline.com/web/search/searchBrowseAction.html?method=searchProducts&searchTerm=632-180&x=9&y=9>
- [8] Juan Antonio Martínez Carrascal, **Circuitos eléctricos, conceptos fundamentales**, Módulo Fundamentos Tecnológicos II, UOC, PID\_00161696
- [9] **Datasheet TL431**, <http://www.fairchildsemi.com/ds/TL/TL431A.pdf>
- [10] **Datasheet TL071**, <http://focus.ti.com/lit/ds/symlink/tl071.pdf>
- [11] **Datasheet INA 146**, <http://focus.ti.com/lit/ds/symlink/ina146.pdf>
- [12] **Datasheet led TLHK5100**, <http://www.vishay.com/docs/83010/tlhe510.pdf>
- [13] **Datasheet zumbador KT-400142**, <http://www.mouser.com/catalog/specsheets/KT-400142.pdf>
- [14] **Datasheet Transistor BC547A**, <http://www.fairchildsemi.com/ds/BC/BC547.pdf>
- [15] **Datasheet LM78XX**, <http://www.fairchildsemi.com/ds/LM%2FLM7805.pdf>
- [16] **Datasheet LM79XX**, <http://www.fairchildsemi.com/ds/LM%2FLM7905.pdf>
- [17] **Datasheet 18F4550**,  
<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010300>
- [18] Información Uart,  
[http://en.wikipedia.org/wiki/Universal\\_asynchronous\\_receiver/transmitter](http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter)

- [19] Protocolo RS232, <http://en.wikipedia.org/wiki/RS-232>
- [20] **Datasheet** DS275, <http://datasheets.maxim-ic.com/en/ds/DS275.pdf>
- [21] **Datasheet** GP-GC021, <http://www.sureelectronics.net/goods.php?id=402>
- [22] Información **Bluetooth**, <https://www.bluetooth.org/apps/content>
- [23] Entorno de desarrollo MPLAB, <http://www.microchip.com>
- [24] Compilador C-18, <http://www.microchip.com>
- [25] Tecnología de montaje superficial,  
[http://es.wikipedia.org/wiki/Tecnolog%C3%ADa\\_de\\_montaje\\_superficial](http://es.wikipedia.org/wiki/Tecnolog%C3%ADa_de_montaje_superficial)

# Anexo A



## Anexo B

```

#include <p18f4550.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <adc.h>
//#include <delays.h>
//#include <timers.h>
#include <usart.h>

#pragma config PLLDIV = 1, CPUDIV = OSC1_PLL2, USBDIV = 2, FOSC =XT_XT, FCMEN
= OFF, IESO = OFF, PWRT = ON
#pragma config BOR = SOFT, BORV = 2, VREGEN = ON, WDT = OFF, WDTPS = 128,
MCLRE = ON, LPT1OSC = OFF, PBADEN = OFF
#pragma config CCP2MX = ON, STVREN = ON, LVP = OFF, ICPRT = OFF, XINST = OFF,
DEBUG = OFF
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF,CP3 = OFF,CPB = OFF,CPD = OFF
#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF, WRTB = OFF,
WRTC = OFF, WRTD = OFF
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF, EBTRB =
OFF

/*****
***/
/*****      PROTOTYPE DEFINITIONS
*****/
/*****
***/
void ISR_LOW_PRIORITY (void);
void ISR_HIGH_PRIORITY (void);
void Msg_Decoder (void);
void Get_Length (void);
void Get_Restriction (void);
void Get_Maintenance (void);
void Compute (void);
int Char_to_int (char *String, int Position);
void Line_Feed(void);

/*****
***/
/*****      INT VECTORS
*****/
/*****
***/
#pragma code InterrupcionHigh = 0x0008
void VectorHighInterrupcion(void){
_asm goto ISR_HIGH_PRIORITY _endasm
}
#pragma code

#pragma code InterrupcionLow = 0x00018
void VectorLowInterrupcion(void){
_asm goto ISR_LOW_PRIORITY _endasm
}
#pragma code

/*****
***/
/*****      VARIABLE DEFINITIONS
*****/
/*****
***/
volatile unsigned int RX_control =0;
    
```

```

volatile unsigned int RX_index=0,index=0,fin=0;
volatile unsigned int Config_Status=0;
volatile char Recived_MSG[11];
unsigned char MSG1[] ="CONFIGURE";
unsigned int Length=0;
unsigned int Restriction=0;
unsigned int Maintenance=0;
signed float Flecha=0;
signed double Coef_F_L=0;
unsigned int MaxAL=0;
unsigned int Al_Maintenance=0;
signed float Strain=0;
signed float Strain_Maintenance=0;
signed float Gauge_factor=2.1;
signed float VM1=0;
signed float VM1_Maintenance=0;
signed float VM1_Neg=0;
signed float VM1_Maintenance_Neg=0;
signed float V1=6;
unsigned int AO_Gain=45;
volatile float Channel_0=0;
volatile float Channel_1=0;
volatile float Channel_0_M=0;
volatile float Channel_1_M=1;
volatile float Offset_CH_0=0;
volatile float Offset_CH_1=0;

/*****
***/
/*****          ISR  ROUTINES
***/
/*****
***/

#pragma interrupt ISR_LOW_PRIORITY
void ISR_LOW_PRIORITY (void){
    if (PIR1bits.RCIF==1){};
}

#pragma interrupt ISR_HIGH_PRIORITY
void ISR_HIGH_PRIORITY (void){
    unsigned int a;
    if (PIR1bits.RCIF==1){
        PIR1bits.RCIF=0;
        INTCONbits.PEIE=0;
        INTCONbits.GIE=0;
        while (!DataRdyUSART());
        Recived_MSG[RX_index]=getcUSART();
        if (Recived_MSG[RX_index]==0x0A){
            Recived_MSG[RX_index-1]=0x00;
            RX_index=0;
            fin=1;
        }
        if (!fin && Config_Status==0){RX_index++;}
        if (!fin && Config_Status==1){RX_index++;}
        if (!fin && Config_Status==2){RX_index++;}
        if (!fin && Config_Status==3){RX_index++;}
        if (fin && Config_Status==0) {Msg_Decoder ();}
        if (fin && Config_Status==1) {Get_Length (); }
        if (fin && Config_Status==2) {Get_Restriction (); }
        if (fin && Config_Status==3) {Get_Maintenance (); }
        if (RX_index > 11){
            Line_Feed();
            printf (" Buffer overflow, System RESET" );
            Line_Feed();
            Delay100TCYx (0);
            Reset();    }
        }

    INTCONbits.PEIE=1;

```

```

        INTCONbits.GIE=1;
    }

/*****
***/
/*****                               MAIN
***/
/*****
***/
void main(void){
OpenUSART(USART_TX_INT_OFF & USART_RX_INT_ON & USART_ASYNC_MODE &
USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH,25);
OpenADC(ADC_FOSC_2 & ADC_12_TAD & ADC_RIGHT_JUST, ADC_REF_VREFPLUS_VSS &
ADC_INT_OFF, ADC_2ANA);
RCONbits.IPEN=1;
INTCONbits.PEIE=1;
INTCONbits.GIE=1;

while(BusyUSART());
    printf (" Structural integrity v 1.0 \n");
    Line_Feed();
    printf (0x0D);
    TRISDbits.TRISD0=0;
    TRISDbits.TRISD1=0;
    TRISDbits.TRISD2=0;
    PORTDbits.RD0=0;
    PORTDbits.RD1=0;
    PORTDbits.RD2=0;
while(1){
    if (Config_Status<5){
        PORTDbits.RD0=1;
    }
    if (Config_Status==5){
        PORTDbits.RD0=0;
    }

    if (Config_Status==5){
        SetChanADC (ADC_CH0);
        ConvertADC();
        while (BusyADC()==1){}
        Channel_0_M= ReadADC();
        Channel_0=Channel_0_M*0.00592;
        Channel_0=Channel_0-Offset_CH_0;
        if (Channel_0 > VM1_Maintenance | Channel_0 <
VM1_Maintenance_Neg){
            PORTDbits.RD1=1;
            Line_Feed();
            printf ("DEFORMATION OF SERVICE REACHED");
            Line_Feed();
        }else {PORTDbits.RD1=0;}
        if (Channel_0 > VM1 | Channel_0 < VM1_Neg){
            PORTDbits.RD2=1;
            Line_Feed();
            printf ("REACHED SAFETY DEFORMATION");
            Line_Feed();
        }else {PORTDbits.RD2=0;}
    }
    if (Config_Status==5){
        SetChanADC (ADC_CH1);
        ConvertADC();
        while (BusyADC()==1){}
        Channel_1_M= ReadADC();
        Channel_1=Channel_1_M*0.00592;
        Channel_1=Channel_1-Offset_CH_1;
        if (Channel_1 > VM1_Maintenance | Channel_1 <
VM1_Maintenance_Neg){
            PORTDbits.RD1=1;
            Line_Feed();
            printf ("DEFORMATION OF SERVICE REACHED");
            Line_Feed();
        }else {PORTDbits.RD1=0;}
    }
}

```

```

        if (Channel_1 > VM1 | Channel_1 < VM1_Neg){
            PORTDbits.RD2=1;
            Line_Feed();
            printf ("REACHED SAFETY DEFORMATION");
            Line_Feed();
        }else {PORTDbits.RD2=0;}
    }
}

void Msg_Decoder (void){
unsigned int i=0;
i = memcmp( MSG1, Recived_MSG, 9 );
    if (!i==0){
        printf ("sintax Error \n");
        Line_Feed();
        fin=0;}
    if (i==0){
        fin=0;
        Config_Status=1;
        printf ("Please input the beam length in mm. \n");
        Line_Feed();}
}

void Get_Length (void){
unsigned int valor=0;
valor=Char_to_int ( Recived_MSG, strlen (Recived_MSG));
    if (valor==0){
        printf ("Please input a decimal value, Maximum 99999 mm.\n");
        Line_Feed();
        fin=0;}
    if (valor>0){
        Length=valor;
        Config_Status=2;
        printf ("Please choose the beam deflection restriction: \n");
        Line_Feed();
        printf ("1. 1/250 2. 1/300 3. 1/400 4. 1/500 \n");
        Line_Feed();
        fin=0;}
}

void Get_Restriction (void){
unsigned int valor=0;
valor=Char_to_int ( Recived_MSG, strlen (Recived_MSG));
    if (valor==0){
        printf ("Please select a decimal value 1,2,3,4 \n");
        Line_Feed();
        fin=0;}
    if (valor<0 | valor>4) {
        printf ("Please select a value between 1 to 4 \n");
        Line_Feed();
        fin=0;}
    if (valor>0 & valor<5) {
        if (valor==1){ Restriction=250;}
        if (valor==2){ Restriction=300;}
        if (valor==3){ Restriction=400;}
        if (valor==4){ Restriction=500;}
        Config_Status=3;
        printf ("Please select the alarm maintenance value (25 to 75) :\n");
        Line_Feed();
        fin=0;
    }
}

void Get_Maintenance (void){
unsigned int valor=0;
valor=Char_to_int ( Recived_MSG, strlen (Recived_MSG));
    if (valor==0){
        printf ("Please input a decimal value between 25 to 75 \n");
        Line_Feed();
        fin=0;}
}

```

```

    if (valor<25 | valor>75){
        Line_Feed();
        printf ("Please input maintenance value between 25 to 75 : \n");
        fin=0;}
    if (valor>24 & valor<76) {
        Maintenance=valor;
        Config_Status=4;
        Line_Feed();
        printf ("Please wait, parameters are being computed \n");
        Compute ();
        fin=0;}
}
void Compute (void){
signed double aux=0;
aux=Length;
Flecha=aux/Restriction;
Flecha=Flecha;
Coef_F_L=1000/Restriction;
Coef_F_L=Coef_F_L/1000;
aux=Coef_F_L/100;
aux=Coef_F_L+aux;
MaxAL=Flecha/aux;
aux=Length-MaxAL;
aux=aux/100;
Al_Maintenance=aux*Maintenance;
Al_Maintenance=Length-Al_Maintenance;
Strain=Length-MaxAL;
Strain=Strain/Length;
Strain_Maintenance=Length-Al_Maintenance;
Strain_Maintenance=Strain_Maintenance/Length;
Gauge_factor=Gauge_factor*(-1);
VM1=Gauge_factor*Strain;
VM1=VM1/2;
VM1=VM1*V1;
VM1=VM1*45;
VM1=VM1-3;
VM1=VM1*(-1);
if (VM1>6){
    Line_Feed();
    printf ( "Device configuration no available, SYSTEM RESET \n");
    Reset();
    Line_Feed();
}
VM1_Maintenance=Gauge_factor*Strain_Maintenance;
VM1_Maintenance=VM1_Maintenance/2;
VM1_Maintenance=VM1_Maintenance*V1;
VM1_Maintenance=VM1_Maintenance*40;
VM1_Maintenance=VM1_Maintenance-3;
VM1_Maintenance=VM1_Maintenance*(-1);

Gauge_factor=Gauge_factor*(-1);
VM1_Neg=Gauge_factor*Strain;
VM1_Neg=VM1_Neg/2;
VM1_Neg=VM1_Neg*V1;
VM1_Neg=VM1_Neg*45;
VM1_Neg=VM1_Neg-3;
VM1_Neg=VM1_Neg*(-1);

VM1_Maintenance_Neg=Gauge_factor*Strain_Maintenance;
VM1_Maintenance_Neg=VM1_Maintenance_Neg/2;
VM1_Maintenance_Neg=VM1_Maintenance_Neg*V1;
VM1_Maintenance_Neg=VM1_Maintenance_Neg*40;
VM1_Maintenance_Neg=VM1_Maintenance_Neg-3;
VM1_Maintenance_Neg=VM1_Maintenance_Neg*(-1);

Line_Feed();
printf ( "Device configuration OK \n");
Line_Feed();
printf ( "Please wait, channel 1 offset measuring... \n" );
Line_Feed();

```

```

SetChanADC (ADC_CH0);
ConvertADC();
while (BusyADC()==1){}
Offset_CH_0 = ReadADC();
Offset_CH_0=Offset_CH_0*0.00592;
Offset_CH_0=Offset_CH_0-3;
printf ( "Please wait, Channel 2 offset measuring... \n" );
Line_Feed();
SetChanADC (ADC_CH1);
ConvertADC();
while (BusyADC()==1){}
Offset_CH_1 = ReadADC();
Offset_CH_1=Offset_CH_1*0.00592;
Offset_CH_1=Offset_CH_1-3;

printf ( "The system is READY to work \n" );
Line_Feed();
Config_Status=5;
}

int Char_to_int (char *String, int Position){
unsigned int  Dec_Millar=0;
unsigned int  Millar=0;
unsigned int  Centena=0;
unsigned int  Decena=0;
unsigned int  Unidad=0;
unsigned int  index=0;
unsigned int  index2=0;
unsigned char numero[5];

    for (index=(5-Position); index<5; index++){
        numero[index]=Recived_MSG[index2];
        index2++;
    }
    for (index=0; index<(5-Position); index++){
        numero[index]=48;
    }

    Dec_Millar=(numero[0]-48);
    if (Dec_Millar > 9 | Dec_Millar<0){ return 0;}
    Dec_Millar=Dec_Millar*10000;
    Millar=(numero[1]-48);
    if (Millar > 9 | Millar<0){ return 0;}
    Millar=    Millar*1000;
    Centena=(numero[2]-48);
    if (Centena > 9 | Centena<0){ return 0;}
    Centena=Centena*100;
    Decena=(numero[3]-48);
    if (Decena > 9 | Decena<0){ return 0;}
    Decena=Decena*10;
    Unidad=(numero[4]-48);
    if (Unidad > 9 | Unidad<0){ return 0;}
    return (Dec_Millar+Millar+Centena+Decena+Unidad);
}

void Line_Feed(void){
putcUSART(0x0D);
}

```