



industriales  
etsii

Escuela Técnica  
Superior  
de Ingeniería  
Industrial

# UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

## DISEÑO Y CONSTRUCCIÓN DE UN EQUIPO DE MEDIDA DE RADIACIONES IONIZANTES

TRABAJO FIN DE GRADO

GRADO EN INGENIERIA INDUSTRIAL ELECTRÓNICA  
Y AUTOMÁTICA

**Autor:** Alberto García Salinas  
**Director:** Joaquín Francisco Roca González  
**Codirector:** José Hurtado Avilés

Cartagena, 31/03/2017



Universidad  
Politécnica  
de Cartagena

## ÍNDICE

<b>Capítulo 1 Introducción y objetivos .....</b>	<b>- 3 -</b>
<b>Capítulo 2 Revisión del estado del arte.....</b>	<b>- 4 -</b>
2.1 - Definiciones y fenómenos radioactivos.....	- 4 -
2.2 - Unidades de dosimetría.....	- 6 -
2.3 - Efectos biológicos de la radiación.....	- 7 -
2.4 - Tubo Geiger - Muller.....	- 8 -
2.4.1 - Circuitos de alimentación .....	- 11 -
2.4.2 - Circuitos de adquisición de pulsos.....	- 13 -
<b>Capítulo 3 Desarrollo hardware.....</b>	<b>- 15 -</b>
3.1 - Descripción general .....	- 15 -
3.2 - Tubo Geiger-Muller SI-3BG.....	- 17 -
3.3 - Circuito generador de alta tensión .....	- 19 -
3.3.1 - Inversor CC/CA.....	- 20 -
3.3.2 - Multiplicador de tensión .....	- 21 -
3.4 - Circuito de conformación de pulsos .....	- 22 -
3.5 - Características principales del PIC 18F14K50 .....	- 22 -
3.6 - Esquemático de la solución propuesta .....	- 26 -
<b>Capítulo 4 Desarrollo software.....</b>	<b>- 28 -</b>
4.1 - Entorno de desarrollo .....	- 28 -
4.1.1 - Compilador CCS C .....	- 29 -
4.1.2 - Placa de desarrollo 18F14K50.....	- 30 -
4.2 - Pruebas iniciales .....	- 31 -
4.2.1 - Encendido/apagado de LED.....	- 32 -
4.2.2 - Comunicación USB.....	- 34 -
4.2.3 - Generación de señal PWM .....	- 37 -
4.2.4 - Medición de CPS (Ciclos por segundo) y CPM (Ciclos por minuto) .....	- 38 -
4.2.5 - Comunicación USB de control .....	- 43 -
4.3 - Integración del software final.....	- 46 -
<b>Capítulo 5 Elaboración del PCB.....</b>	<b>- 53 -</b>
5.1 - Planteamiento del desarrollo .....	- 53 -
5.2 - Easily Applicable Graphical Layout Editor (EAGLE).....	- 53 -
5.3 - PCB, montaje y puesta en marcha.....	- 55 -
5.3.1 - Elaboración del layout del PCB.....	- 55 -
5.3.2 - Elaboración del PCB por insolación y ataque químico.....	- 55 -

---

<b>Capítulo 6 Conclusiones y trabajos futuros</b> .....	- 59 -
6.1 - Conclusiones.....	- 59 -
6.1 - Trabajos futuros.....	- 59 -
Anexo A. Bibliografía y páginas web.....	- 61 -
Anexo B. Presupuesto.....	- 61 -
Anexo C. Planos .....	- 64 -
Anexo D. Datasheet .....	- 69 -
Anexo E. Código fuente.....	- 69 -
Anexo F. Configuración PWM.....	- 71 -
Anexo G. Tabla ASCII .....	- 72 -
Anexo H. Índice de figuras .....	- 72 -

# Capítulo 1

## Introducción y objetivos

El objetivo principal de este trabajo es el diseño de una tarjeta que actúe como medidor de radiaciones electromagnéticas ionizantes, realizándose estas mediciones mediante un tubo Geiger-Muller contador de radiación. El tubo medidor es apto para medir las radiaciones gamma y las radiaciones beta ( $\beta$ ).

La tarjeta ha de ser capaz de comunicarse con el PC para enviar los datos procedentes de las mediciones realizadas y, al mismo tiempo, poder, desde el PC, cambiar parámetros en la configuración de la tarjeta.

El proyecto se ha estructurado en torno a diferentes bloques con el fin de facilitar el correcto y pleno entendimiento del mismo; en ellos se describen las características funcionales del sistema, además de los trabajos hardware y software efectuados para la elaboración del mismo.

Dicha tarjeta podrá ser utilizada en entornos clínicos, así como en docencia con el fin de ofrecer una demostración del modo de funcionamiento del tubo de medición, y uno de los modos posibles de cuantificación de estas radiaciones.



**Fig 1.1 - Diferentes tubos Geiger-Muller**

# Capítulo 2

## Revisión del estado del arte

Como introducción, se ha realizado un análisis de las distintas unidades de medida de estas radiaciones, asimismo también se ha hecho un breve resumen de los riesgos que conlleva exponerse a ellas.

En este proyecto se ha empleado el tubo Geiger-Muller para realizar las distintas mediciones de radiación. A lo largo de este capítulo se ha explicado su principio de funcionamiento, así como los distintos circuitos electrónicos necesarios para su correcta utilización.

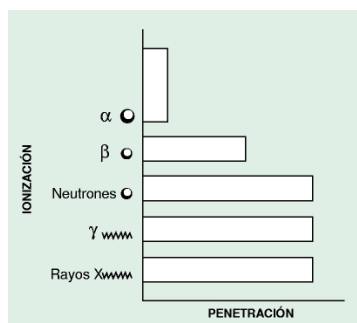
### 2.1 - Definiciones y fenómenos radioactivos

Se define una radiación como ionizante cuando al interaccionar con la materia produce la ionización de la misma, es decir, origina partículas con carga eléctrica (iones). El origen de estas radiaciones es siempre atómico, pudiéndose producir tanto en el núcleo del átomo como en los orbitales y pudiendo ser de naturaleza corpuscular (partículas subatómicas) o electromagnética (rayos X, rayos gamma ( $\gamma$ )).

Las radiaciones ionizantes de naturaleza electromagnética son similares en naturaleza física a cualquier otra radiación electromagnética, pero con una energía fotónica muy elevada (altas frecuencias, bajas longitudes de onda) capaz de ionizar los átomos.

Existen varios tipos de radiaciones emitidas por los átomos, siendo las más frecuentes: la desintegración Alfa " $\alpha$ ", la desintegración Beta " $\beta$ ", la emisión gamma " $\gamma$ " y la emisión de rayos X y neutrones. Las características de cada radiación varían de un tipo a otro, siendo importante considerar su capacidad de ionización y su capacidad de penetración, que en gran parte son

consecuencia de su naturaleza. En la siguiente figura se representan esquemáticamente estas radiaciones.



**Fig. 2.1 - Tipos de radiación**

### **Radiación $\alpha$**

Son núcleos de helio cargados positivamente; tienen una energía muy elevada y muy baja capacidad de penetración.

### **Radiación $\beta^-$**

Son electrones emitidos desde el núcleo del átomo como consecuencia de la transformación de un neutrón en un protón y un electrón.

### **Radiación $\beta^+$**

Es la emisión de un positrón, partícula de masa igual al electrón y carga positiva, como resultado de la transformación de un protón en un neutrón y un positrón. Las radiaciones  $\beta$  tienen un nivel de energía menor que las  $\alpha$  y una capacidad de penetración mayor.

### **Radiación de neutrones**

Es la emisión de partículas sin carga, de alta energía y gran capacidad de penetración. Los neutrones se generan en los reactores nucleares y en los aceleradores de partículas, no existiendo fuentes naturales de radiación de neutrones.

### **Radiación gamma $\gamma$**

Son radiaciones electromagnéticas procedentes del núcleo del átomo, tienen menor nivel de energía que las radiaciones  $\alpha$  y  $\beta$  y mayor capacidad de penetración, lo que dificulta su absorción por los apantallamientos.

### **Rayos X**

Son radiaciones de naturaleza electromagnética que se originan en los orbitales de los átomos como consecuencia de la acción de los electrones rápidos sobre la corteza del átomo. Son de menor energía que las radiaciones gamma, pero presentan una gran capacidad de penetración y son absorbidos por apantallamientos especiales de grosor elevado.

Type	Origin	Process	Charge	Mass [MeV]	Spectrum (energy)
$\alpha$ -particles	Nucleus	Nuclear decay or reaction	+ 2	3727.33	Discrete [MeV]
$\beta^-$ -rays	Nucleus	Nuclear decay	- 1	0.511	Continuous [keV - MeV]
$\beta^+$ -rays (positrons)	Nuclear	Nuclear decay	+ 1	0.511	Continuous [keV - MeV]
$\gamma$ -rays	Nucleus	Nuclear deexcitation	0	0	Discrete [keV - MeV]
x-rays	Electron cloud	Atomic deexcitation	0	0	Discrete [eV - keV]
Internal conversion electrons	Electron cloud	Nuclear deexcitation	- 1	0.511	Discrete [high keV]
Auger electrons	Electron cloud	Atomic deexcitation	- 1	0.511	Discrete [eV - keV]
Neutrons	Nucleus	Nuclear reaction	0	939.57	Continuous or discrete [keV - MeV]
Fission fragments	Nucleus	Fission	$\approx 20$	80 - 160	Continuous 30 - 150 MeV

Fig. 2.2 - Relación energética de las radiaciones

## 2.2 - Unidades de dosimetría

Para cuantificar la radiación es necesario utilizar unidades de medida que indiquen su valor energético, así como la dosis recibida, ya que ambos parámetros definen en gran medida las consecuencias de esta radiación.

### Nivel de energía

El nivel de energía de una radiación ionizante se mide en electronvoltios (eV). El electronvoltio corresponde a la energía que adquiere un electrón cuando se aplica, en el vacío, una diferencia de potencial de 1 voltio y equivale a  $1,6 \times 10^{-19}$  Julios.

### Dosis absorbida (D)

Es la cantidad de energía (D) cedida por la radiación a la materia irradiada por unidad de masa. La unidad de medida en el sistema internacional es el Gray (Gy).

### Dosis equivalente ( $H_T$ )

Es también una magnitud que considera la energía cedida por unidad de masa, pero considerando el daño biológico. Es el producto de la dosis absorbida (D) por un factor de ponderación de la radiación  $W_r$ . Su unidad de medida es el Sievert (Sv). Esta es una unidad muy grande para su utilización en protección radiológica y por ello se suele utilizar sus submúltiplos, el millisievert (mSv,  $10^{-3}$  Sv) y el microsievert ( $\mu$ Sv,  $10^{-6}$  Sv).

### Dosis efectiva (E)

El efecto biológico de una radiación depende del tejido sobre el cual está incidiendo esta radiación. Para considerar este efecto se utiliza un coeficiente denominado factor de *ponderación del tejido*.

Tissue or organ	Tissue weighting factor, $w_T$
Gonads	0.20
Bone marrow	0.12
Colon	0.12
Lung	0.12
Stomach	0.12
Bladder	0.05
Breast	0.05
Liver	0.05
Oesophagus	0.05
Thyroid	0.05
Skin	0.01
Bone surface	0.01
Remainder	0.05

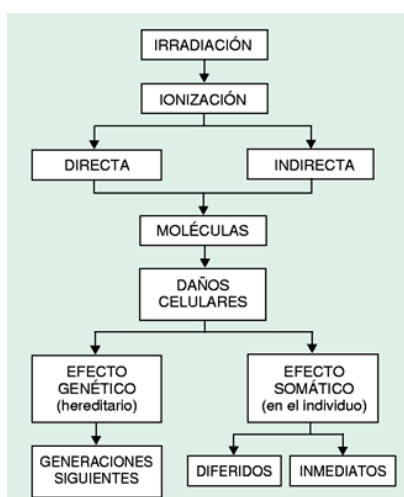
**Fig. 2.3 - Factores de ponderación**

## 2.3 - Efectos biológicos de la radiación

La radiación de alta energía puede producir graves daños biológicos debido a su facultad para originar cambios en la estructura molecular de la materia, logrando afectar a los órganos internos.

Esta ionización puede dañar directamente enlaces químicos de moléculas importantes (ADN) o indirectamente al crear radicales químicos que atacan a las moléculas mediante procesos químicos. Los efectos de esta radiación pueden ser:

- Muerte de la célula.
- Un funcionamiento incorrecto que puede derivar en cáncer.
- Una alteración que puede ser transmitida a generaciones futuras.



**Fig. 2.4 - Esquema de efectos biológicos**

Las consecuencias de la radiación dependen de la dosis equivalente absorbida. Estas consecuencias pueden manifestarse de manera inmediata o aparecer transcurrido un gran periodo de tiempo, y es esta cualidad la que hace que para dosis reducidas sea complicado relacionar la dosis recibida y el efecto producido.



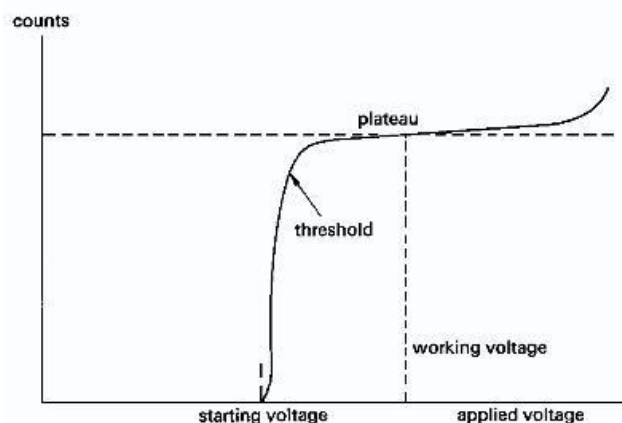
Stage of development	Effect	Threshold dose (Sv)	Radiation exposure	Excess fatal cancers (per 10 <sup>5</sup> persons exposed)
Embryo	Small head circumference	0.04	Single, brief exposure to 0.1 Sv	790
Fetus	Diminished body growth Increased infant mortality	0.2	Continuous lifetime exposure to 1 mSv/yr	560
Child	Hypothyroidism	5	Continuous exposure to 0.01 Sv/yr from age 18 until age 65	3000
Adult	Opacity of the eye lens	2.5		
Adult	Death	2-3		
Adult	Aging	3		
Adult	Erythema (reddening of the skin)	3-10		
Male adult	Temporary sterility	0.5-1		
	Permanent sterility	>5		
Female adult	Permanent sterility	3-4		

**Fig. 2.5 - Dosis umbral para diferentes efectos y probabilidad de aparición de cáncer según la dosis**

## 2.4 - Tubo Geiger - Muller

El tubo Geiger-Muller es el elemento sensible a la radiación del contador Geiger, utilizado para la medición de radiación ionizante, es decir, rayos gamma, rayos X o partículas alfa ( $\alpha$ ) o beta ( $\beta$ ), sin embargo se debe señalar que el tubo usado en el presente trabajo solamente es capaz de detectar rayos gamma y partículas  $\beta$ .

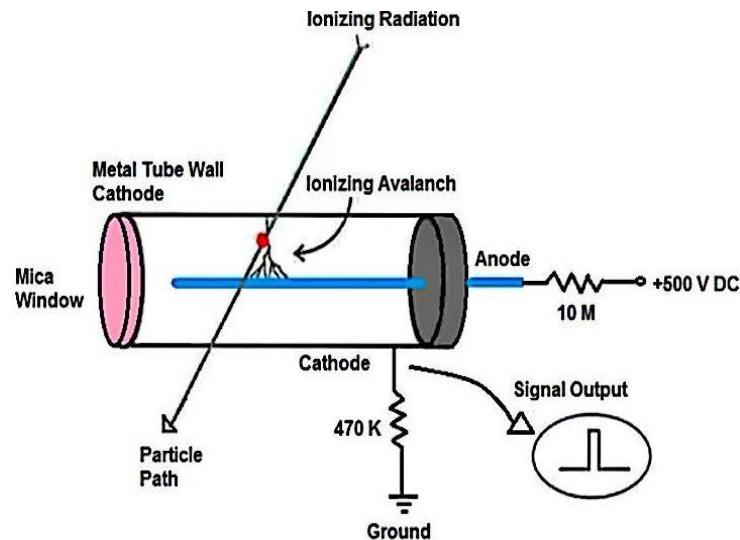
El tubo consiste en una cámara llena de un gas inerte a baja presión. La cámara contiene dos electrodos, entre los que hay una gran diferencia de potencial, típicamente entre 300 y 500 voltios. Las paredes del tubo son de metal o tienen su superficie interior revestida con un conductor para formar el cátodo, mientras que el ánodo es un alambre en el centro de la cámara.



**Fig.2.6 - Comportamiento del tubo frente a la tensión de entrada**

Cuando la radiación ionizante externa alcanza al tubo, algunas moléculas del gas son ionizadas, ya sea directamente por la radiación incidente o indirectamente por medio de electrones secundarios producidos en las paredes del tubo, lo cual crea iones cargados positivamente y electrones, conocidos como pares de iones, en el gas del interior de la cámara. El fuerte campo eléctrico creado por los electrodos del tubo acelera los iones positivos hacia el

cátodo y los electrones hacia el ánodo, cerca del ánodo, en la "región de avalancha", los electrones ganan suficiente energía como para ionizar moléculas de gas adicionales y crear una avalancha de electrones que se extiende a lo largo del ánodo y de manera efectiva a lo largo de la región de avalancha. Este es el efecto de "multiplicación de gas" que da al tubo su característica clave de poder producir un impulso de salida significativo a partir de una sola partícula ionizante.



**Fig. 2.7 - Esquema y funcionamiento de un tubo Geiger-Muller**

Si bien es un detector robusto y barato, el G-M es incapaz de medir altas tasas de radiación de manera eficiente, su tiempo de vida útil disminuye considerablemente en áreas de alta radiación y no puede medir la energía de radiación incidente, por lo que no se puede generar información espectral al no producir ningún tipo de discriminación entre tipos de radiación, con lo cual, no es posible saber el tipo de radiación incidente.

Un parámetro importante en el manejo de estos dispositivos es el tiempo que debe transcurrir entre dos detecciones de partículas sucesivas, llamado *dead time*, puesto que si se producen dos choques con un espaciado de tiempo menor que éste, el tubo es incapaz de detectarlo. En el caso del tubo LND 712 este tiempo máximo es de 90 microsegundos, pero este tiempo de latencia es propio de cada tipo de tubo.

El otro parámetro esencial es la relación entre la cantidad de impulsos generados por el tubo y la intensidad de la dosis de radiación que se está midiendo. La siguiente gráfica muestra la curva de respuesta típica para un tubo GM genérico frente a una radiación, aunque cada tubo posee su propia curva de respuesta típica. Habitualmente se suele emplear Cesio (Cs) o Cobalto (Co) para la calibración estos tubos.

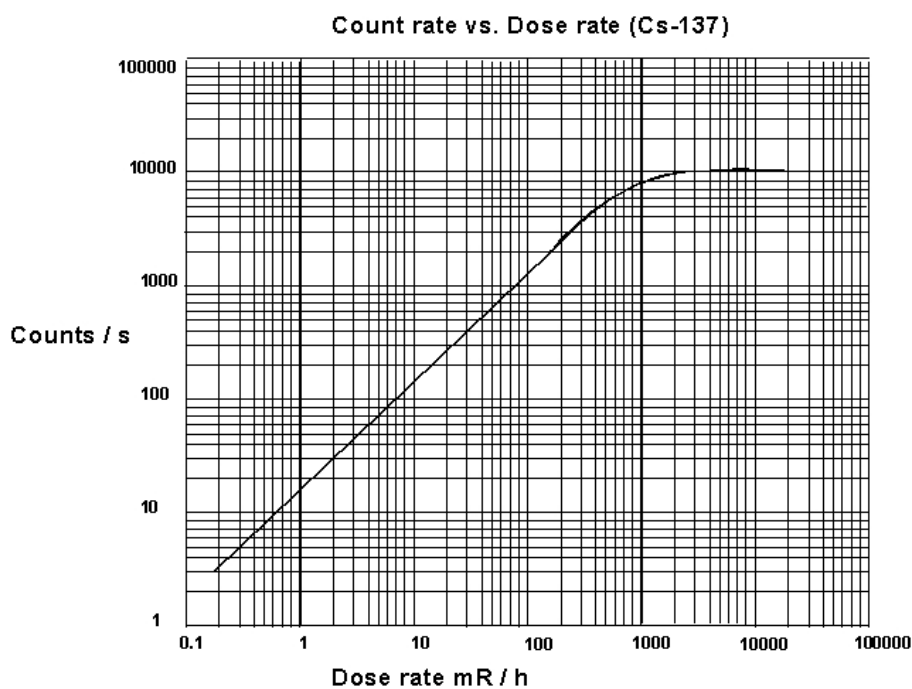


Fig.2.8 - Número de pulsos vs dosis medida

En la siguiente imagen se pueden observar diferentes comportamientos de estos tubos frente a distintas fuentes de radiación.

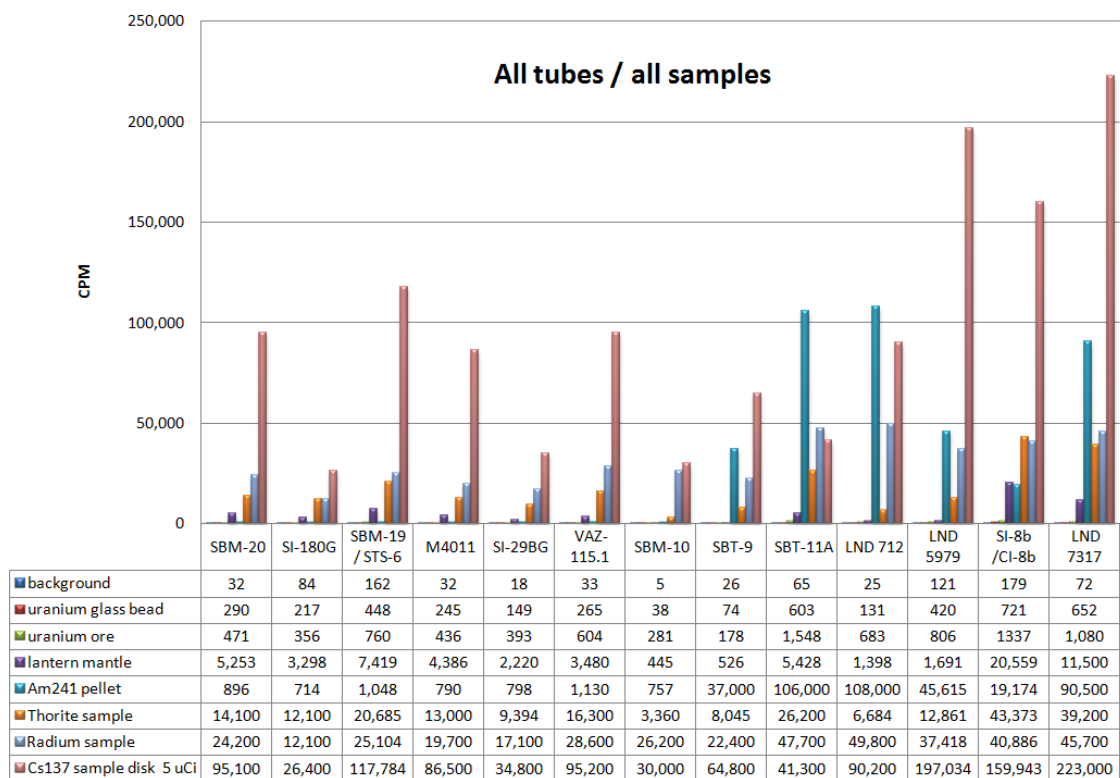


Fig.2.9 - Mediciones de ciclos por segundo frente a determinadas radiaciones

## 2.4.1 - Circuitos de alimentación

El alto voltaje con que se ha de alimentar al tubo para que sea posible la ionización del gas, hace necesaria la utilización de distintos circuitos que conviertan una señal TTL (5V) a una señal comprendida entre los 300 o 500 voltios. A continuación se pueden observar distintos ejemplos de estos circuitos.

### ✚ Primer circuito de alimentación del tubo

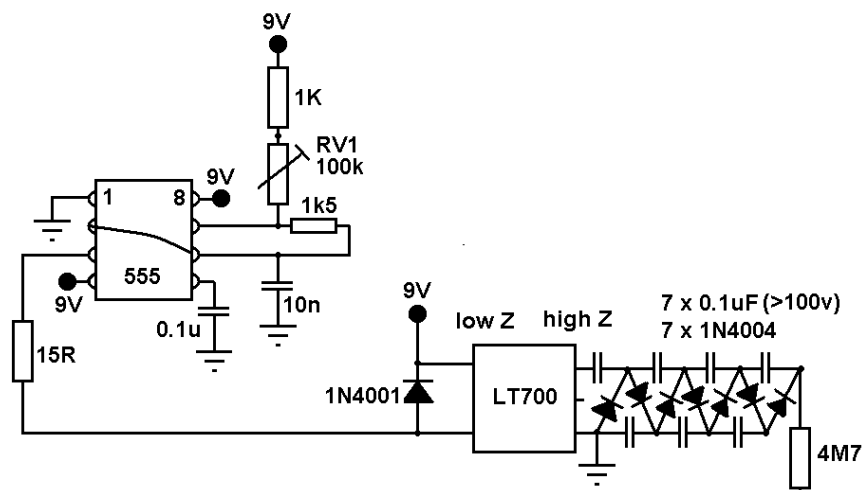


Fig. 2.10 - Primer circuito generador de tensión

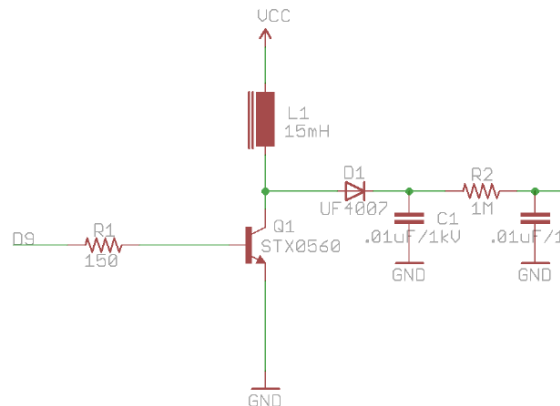
El funcionamiento de este primer circuito se basa en el uso del circuito integrado 555 en configuración astable para actuar como circuito temporizador y generador de pulsos.

Estos pulsos generados por el integrado van a parar a un transformador elevador que los convierte a pulsos de 50-100V. El diodo colocado en paralelo con el transformador LT700 impide el funcionamiento óptimo del circuito, por lo que éste debe ser eliminado.

Acto seguido, este nivel de tensión es aumentado mediante una etapa multiplicadora de la tensión basada en el uso de condensadores y diodos, obteniéndose una señal de alto voltaje. La resistencia de 4,7 Megaohmios es una resistencia limitadora de corriente para no dañar el tubo medidor.

Este circuito es capaz de convertir una señal de 9V en otra de 300V.

### Segundo circuito de alimentación del tubo(DIY Geiger counter GK mini)

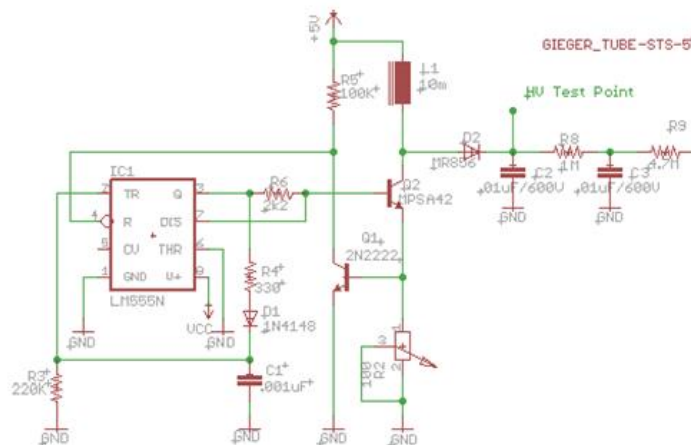


**Fig. 2.11 - Segundo circuito generador de tensión**

El funcionamiento de este segundo circuito está basado en un transistor (MOSFET, BJT o IGBT) atacado mediante una señal PWM generada por un microcontrolador PIC. Estos pulsos generan la apertura y el cierre del circuito que alimenta a la bobina, lo que genera una fuerte polarización en la bobina, originando una señal de alto valor de tensión, la cual, a continuación, es recortada y estabilizada.

Este circuito proporciona una señal 500V a partir de una onda de 5V.

### Tercer circuito de alimentación del tubo (Geiger counter GK v2)



**Fig. 2.12 - Tercer circuito generador de tensión**

Este último circuito es una combinación de los dos circuitos mencionados anteriormente, es decir, está formado por un integrado 555 que genera los pulsos que llegan al transistor Q2, el cual excita a la bobina L1, lo que genera un gran valor de tensión a la entrada del diodo D2. A continuación, la señal es recortada, estabilizada y se le hace pasar por un filtro pasa-bajo para eliminar los armónicos de alta frecuencia para mejorar la señal. La resistencia de 4,7 Megaohmios es una resistencia de protección limitadora de corriente.

Este circuito es capaz de transformar una señal de 5V de amplitud a una de 400 Voltios.

## 2.4.2 - Circuitos de adquisición de pulsos

Para la correcta adquisición del tren de impulsos generado por el tubo medidor se precisa la inclusión de un circuito de adquisición de pulsos que genere pulsos TTL (de 0 a 5 voltios). Seguidamente se pueden observar distintas configuraciones de estos circuitos.

### ✚ Primer circuito de adquisición (Sparkfun Geiger counter)

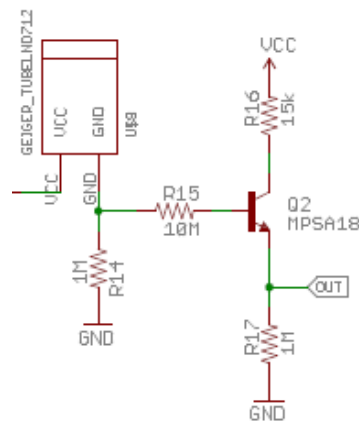


Fig. 2.13 - Primer circuito de adquisición

Este circuito, al igual que los siguientes, está basado en el ataque a un transistor, el cual cuando recibe un pulso por la base, la señal "OUT" pasa de nivel bajo a nivel alto. Las resistencias R15 y R14 son utilizadas para adecuar la señal de entrada procedente del tubo hasta la base del transistor.

### ✚ Segundo circuito de medición (DIY Geiger counter GK mini)

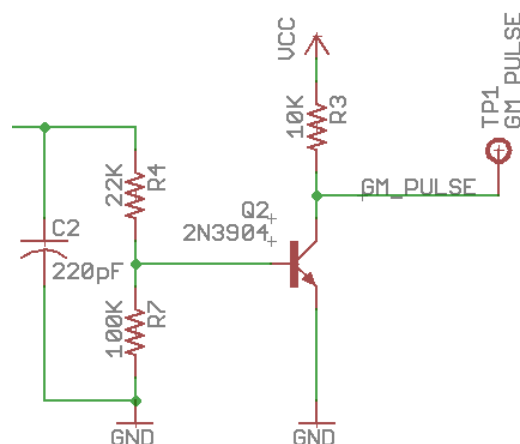
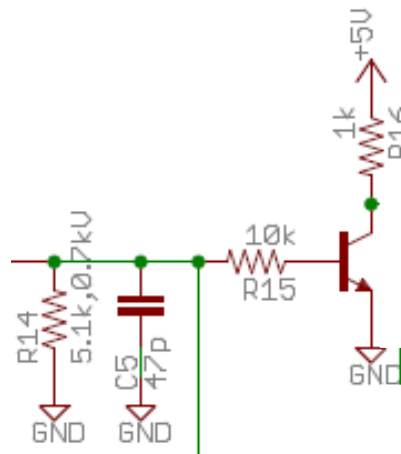


Fig. 2.12 - Segundo circuito de adquisición

En este circuito cuando se produce un pulso en la base del transistor, la señal medida pasa de 5V a 0, es decir, funciona con lógica inversa. Las resistencias R7 y R4 están encargadas de adecuar la señal de entrada procedente del tubo hacia la base del transistor, y el condensador C2 se utiliza para evitar posibles picos de voltaje. Esta configuración ha sido la escogida en este proyecto.

#### ✚ Tercer circuito de adquisición (Build your own Geiger-Muller counter)



**Fig. 2.12 - Tercer circuito de adquisición**

Este tercer y último circuito funciona con lógica directa, al igual que el primer circuito, es decir, cuando llega un impulso a la base del transistor, la señal medida pasa de nivel bajo a nivel alto. Tanto las resistencias R14 y R15, como el condensador C5 son empleados para adecuar la señal producida por el tubo.

# Capítulo 3

## Desarrollo hardware

### 3.1 - Descripción general

En el siguiente diagrama de bloques se muestra la organización de los componentes que conforman el proyecto, mostrando las relaciones entre cada uno de los diferentes dispositivos que se emplean en este documento.

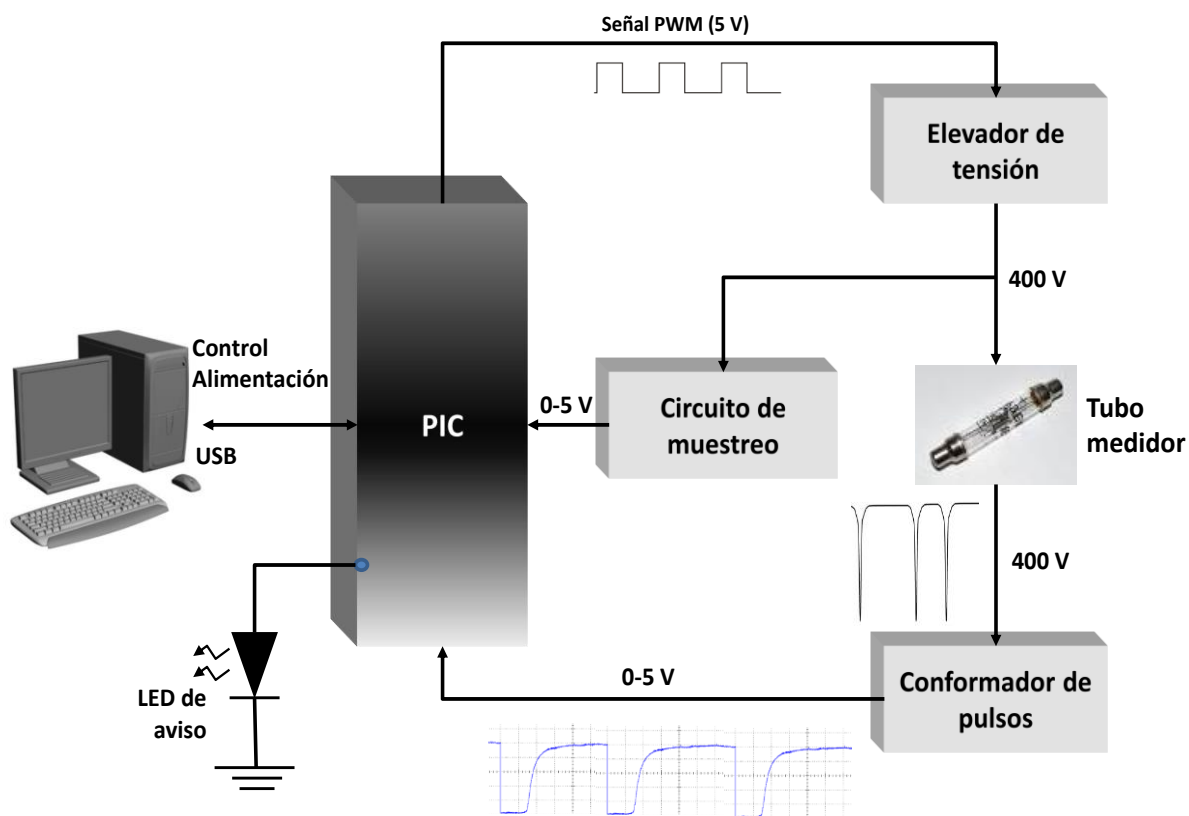


Fig. 3.1 - Diagrama Hardware



El sistema comienza con la inicialización del PIC y de las tablas de valores almacenadas en la memoria RAM del mismo. Estas tablas son modificadas por el tren de impulsos generado por el tubo Geiger-Muller, el cual llega al microcontrolador a través del circuito conformador de pulsos. Posteriormente estas tablas son enviadas al PC vía USB. Se dispone de un led señalizador que se activa cuando este tren de impulsos supera una frecuencia determinada.

El tubo necesita ser alimentado con una señal de gran diferencia de potencial, como ya se ha mencionado, la cual se consigue mediante un circuito elevador de tensión capaz de convertir una señal PWM de 5V a otra de 200 - 500 Voltios. La configuración de esta señal PWM determina el valor del voltaje generado, siendo controlada la señal por el usuario a través del teclado, tanto en frecuencia como en relación cíclica de trabajo.

Para la comunicación de datos entre el PIC y el PC, se hace uso de la comunicación USB, además, el puerto USB es utilizado como alimentación del PCB. La programación del PIC sólo se puede efectuar a través del conector ICSP.

Como comentario final señalar que se dispone de un circuito de muestreo analógico-digital para que la señal de alta tensión pueda ser gestionada de manera autónoma por el microcontrolador. Este módulo no ha sido desarrollado, dejándose su estudio para futuros trabajos.

Para realizar este proyecto se ha hecho necesaria una programación modular a causa de su dificultad. En este capítulo y en el siguiente, se exponen tanto el entorno de desarrollo hardware con el que se ha elaborado el proyecto, así como el desarrollo software realizado para tal fin.

### 3.2 - Tubo Geiger-Muller SI-3BG

En este proyecto se ha optado por utilizar el modelo SI-3BG de tubo Geiger- Muller. Las características principales de este dispositivo son:

- Valor máximo del rango de medición. 300 Roentgen/hora  $\equiv$  2,631 Sievert/hora
- Valor máximo de radiación que puede recibir.  
900 Roentgen/hora  $\equiv$  7,893 Sievert/hora durante 1 minuto
- Voltaje umbral. Entre 290 y 330 V
- Amplitud de voltaje del modo de trabajo. 80 V
- No más de 0.25 % / V en la pendiente de la recta de trabajo.
- Sensibilidad = como mínimo  $2 \cdot 10^{-5}$  pulsos/segundo
- Voltaje de trabajo en modo pulso. De 380 a 460 V
- Voltaje de trabajo en modo corriente. De 382 a 398 V
- Corriente de trabajo. De 15 a 20  $\mu$ A



**Fig. 3.2 - Tubos SI-3BG**

Para la consulta con más detalle de este elemento en la imagen 3.3 se muestra el datasheet del componente, con el hándicap de encontrarse en alfabeto cirílico, ya que el dispositivo es de origen ruso.


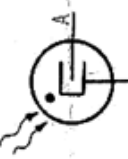
<p>ОКД 63 6451 1436</p>  <p>Счетчик СИЗБГ Э т к е т к а</p>	<p>Счетчик бета-гамма-излучения СИЗБГ предназначен для регистрации гамма-излучения мощностью экспозиционной дозы до <math>21,5 \cdot 10^{-6}</math> А/кг (300 Р/ч), как в импульсном, так и в токовом режиме.</p> <p>Счетчик поставляют в климатическом исполнении УХЛ категории 2.1 по ГОСТ 15150-69. <b>ИЮНЬ 1985</b></p> <p>Дата выпуска</p> <p>Схема соединения электродов с выводами</p>	 <p>А - анод К - катод</p>	<p>Основные электрические параметры</p> <p>Напряжение начала счета, В, не менее 290 330</p> <p>Протяженность плеча счетной характеристики, В, не менее 80</p> <p>Наклон плеча счетной характеристики, %/В, не более 0,25</p> <p>Наклон вольтамперной характеристики, %/В, не более 1,5</p> <p>Чувствительность по скорости счета, ИМП/с · (А/кг)⁻¹ ИМП/с · (Р/ч)⁻¹, не менее 1,3 · 10⁻⁵ (188) 2 · 10⁻⁵ (282)</p>
<p>Д о п у с т и м н ы е р е ж и м ы э к с п л у а т а ц и и</p>	<p>Наименование параметра</p> <p>Напряжение на аноде в импульсном режиме, В, не менее 380 460</p> <p>Напряжение на аноде в токовом режиме, В, не менее 382 398</p> <p>Наибольший рабочий ток, мкА, не менее 15 20</p> <p>Максимально-допустимая мощность экспозиционной дозы гамма-излучения, А/кг (Р/ч), (в течение 1 мин.), не более 6,4 · 10⁻⁵ (900)</p>	<p>У к а з а н и я п о э к с п л у а т а ц и и</p> <p>1. При эксплуатации счетчика значения величин, определяющих режим эксплуатации, не должны выходить за указанные предельно допустимые значения. Невыполнение этого требования может привести к потере работоспособности счетчика.</p>	<p>2. Для продолжительной и устойчивой работы счетчика необходимо:</p> <p>не допускать попадания на счетчик всякого рода загрязнений, могущих ухудшить его изоляционные свойства; оберегать счетчик от ударов и механических повреждений;</p> <p>при включении счетчика соблюдать полярность.</p> <p>Технические условия ОДО 339 387 ТУ.</p>

Fig. 3.3 - Datasheet tubo SI-3BG

### 3.3 - Circuito generador de alta tensión

La generación de esta alta tensión se ha realizado mediante dos módulos claramente diferenciados, un inversor CC/CA y un circuito multiplicador de tensión.

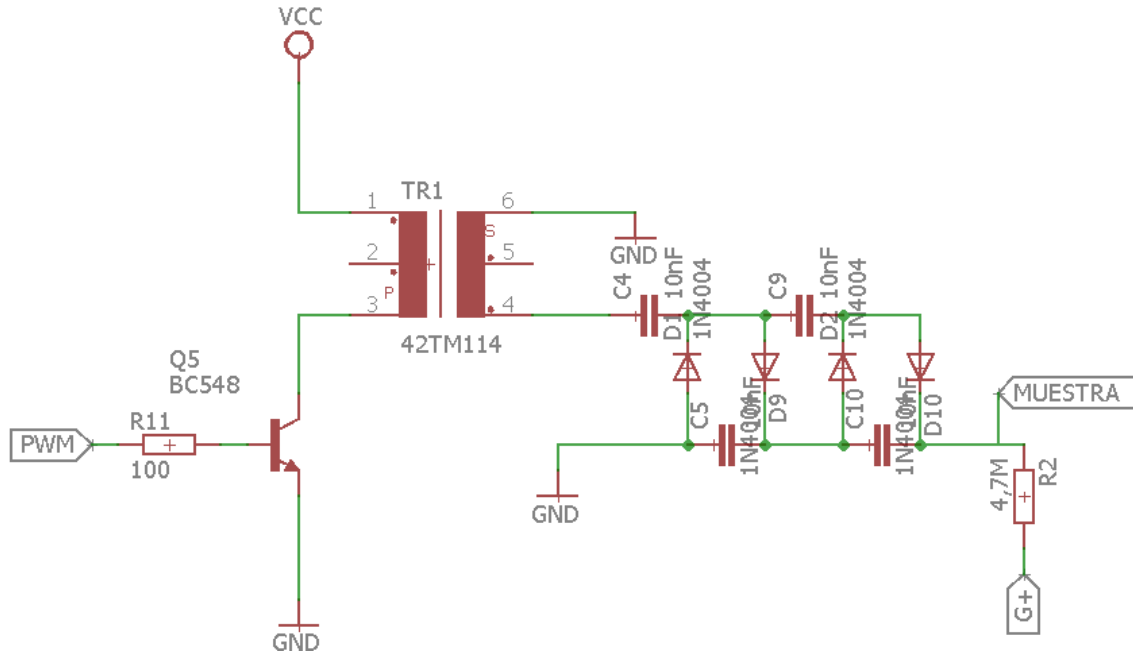


Fig. 3.4 - Esquemático de la generación de alta tensión

Este circuito elevador de tensión ofrece el siguiente comportamiento para una señal PWM de 50.000 Hz (frecuencia consignada) en función de la relación de trabajo de dicha señal.

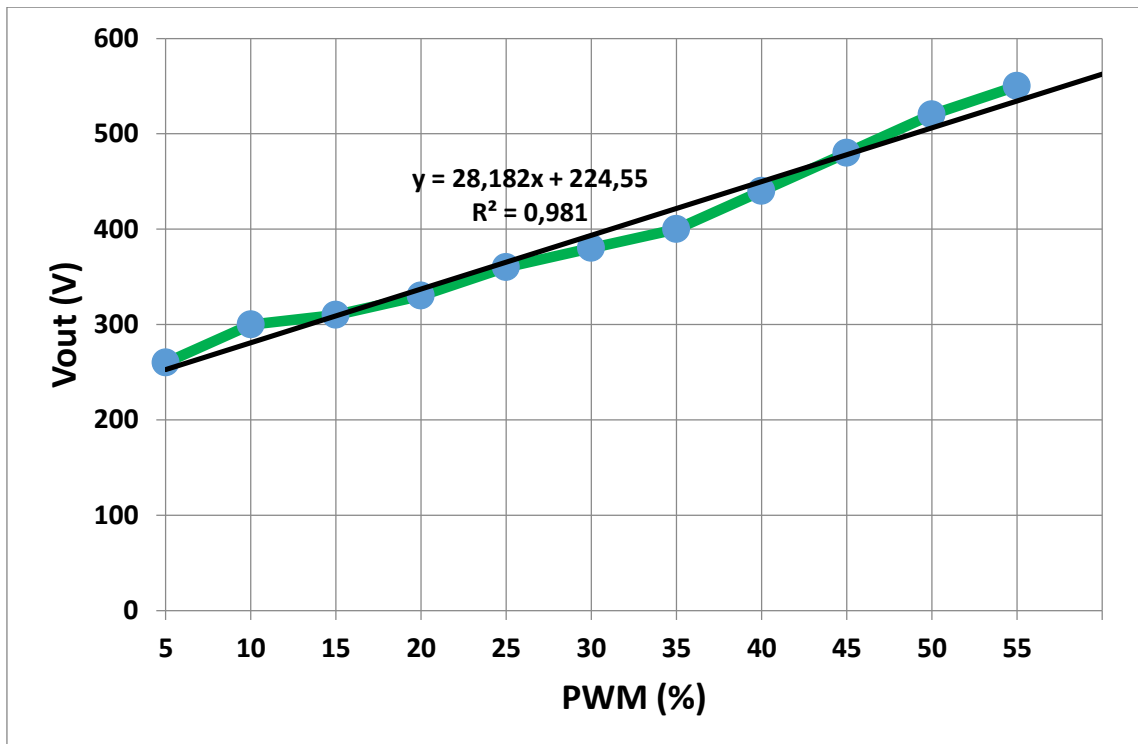


Fig. 3.5 - Comportamiento generador de tensión para PWM de 50KHz

✚ Importante. Medición del alto voltaje.

El tubo GM requiere una gran diferencia de potencial, pero sólo una pequeña cantidad de corriente. Esto origina que medir el voltaje sea un poco complicado, ya que no se puede medir de manera directa.

Al medir el voltaje, un polímetro típico supone una carga de aproximadamente  $10\text{M}\Omega$  en el circuito que está midiendo. Esta carga es excesiva para la pequeña cantidad de corriente disponible, y el polímetro lee un valor mucho más bajo que el voltaje real. Para el caso de un medidor con una impedancia de entrada de  $10\text{M}\Omega$  puede leer 214V cuando en realidad el voltaje está cerca de 420V. Se necesita alrededor de un giga-ohmio de impedancia de entrada para obtener valores precisos en la medición del voltaje.

Una forma de aumentar la impedancia de entrada del polímetro es colocar grandes resistencias en serie con la sonda y multiplicar la lectura que se obtenga, por ejemplo, añadir 9 resistencias de  $10\text{M}\Omega$  en serie añade  $90\text{M}\Omega$ . Una vez que se han añadido las resistencias en serie con el medidor, hay que multiplicar la lectura por un factor, calculado con la fórmula siguiente:

$$V_{real} = \frac{(R_{polímetro} + R_{anexa})}{R_{polímetro}} \times V_{medido}$$

Por ejemplo, si se construye una "sonda" de  $90\text{M}\Omega$  para un medidor típico de  $10\text{M}\Omega$ , se tiene que  $90 + 10/10 = 10$ , por lo tanto la lectura ha de multiplicarse por 10. Si se utiliza una resistencia de  $1\text{G}\Omega$ , con el mismo dispositivo de medida, se tiene  $1000 + 10/10 = 101$ , así que la lectura por debe ser multiplicada por 101.

### 3.3.1 - Inversor CC/CA

El funcionamiento de este inversor CC/CA se fundamenta en el paso de corriente o no a través del transistor a causa de una señal de PWM generada por el microcontrolador.

Estas oscilaciones de corriente hacen que se excite el transformador elevador convirtiendo una señal de 0 a 5V de corriente continua a señales de corriente alterna de un potencial mayor que la señal original del microcontrolador.

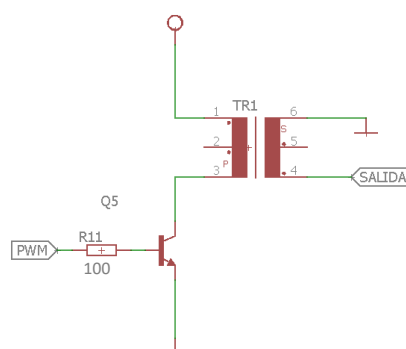


Fig. 3.6 - Circuito inversor CC/CA

Este circuito inversor presenta el siguiente comportamiento de Voltaje pico-pico ( $V_{pp}$ ) en función de la frecuencia de la señal PWM con que se ataca el transistor. Esta señal posee, en todas las frecuencias, una relación de trabajo del 40%.

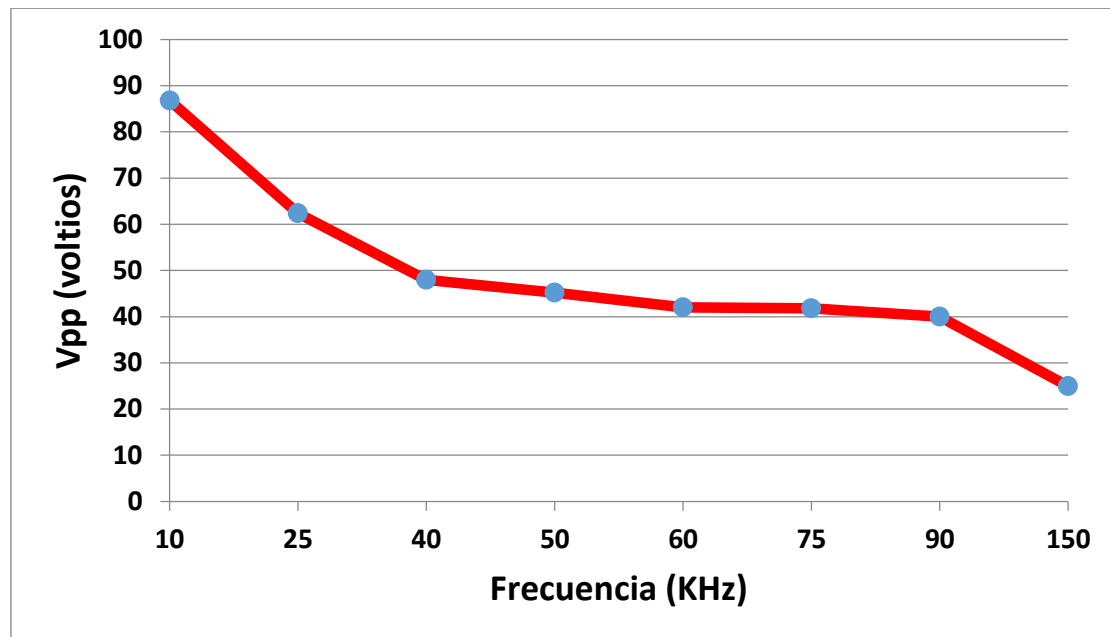


Fig. 3.7 -  $V_{pp}$  en función de la frecuencia

### 3.3.2 - Multiplicador de tensión

El multiplicador de tensión es un circuito que convierte tensión desde una fuente de corriente alterna, generada en la etapa inversora CC/CA, a otra de corriente continua de mayor voltaje mediante una serie de etapas de diodos y condensadores. En este trabajo se ha realizado un circuito cuadruplicador de tensión como el que se muestra en la imagen 3.8.

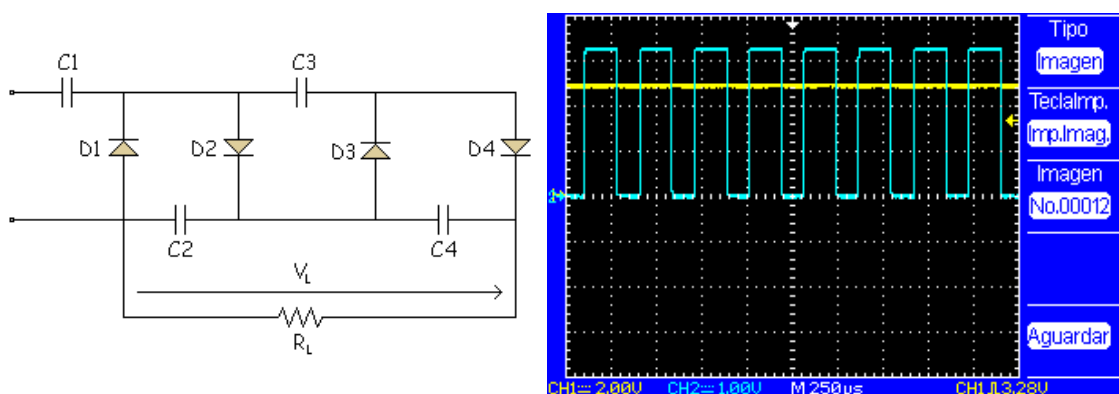


Fig. 3.8 A) Cuadruplicador de tensión B) Señal obtenida

### 3.4 - Circuito de conformación de pulsos

El funcionamiento de este circuito de adquisición de pulsos se basa en impedir el paso de corriente o no a través del transistor debido a los impulsos originados por el tubo Geiger-Muller, que entran al circuito por la señal "G-".

Estas oscilaciones de la señal son capturadas por el microcontrolador, mediante una interrupción de flanco de bajada, para cuantificar la cantidad de pulsos que produce el tubo por magnitud de tiempo.

En este circuito, cuando se produce un flanco de bajada en la base del transistor, la señal medida (PULSO) pasa de estado alto (5V) a estado bajo (0V), es decir, funciona con lógica inversa.

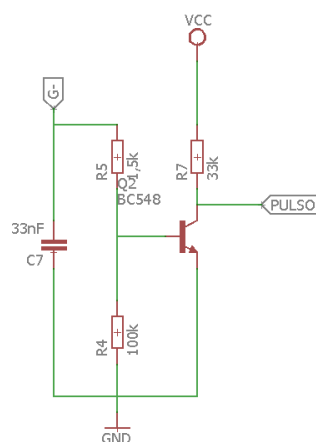


Fig. 3.9 - Circuito de adquisición de pulsos

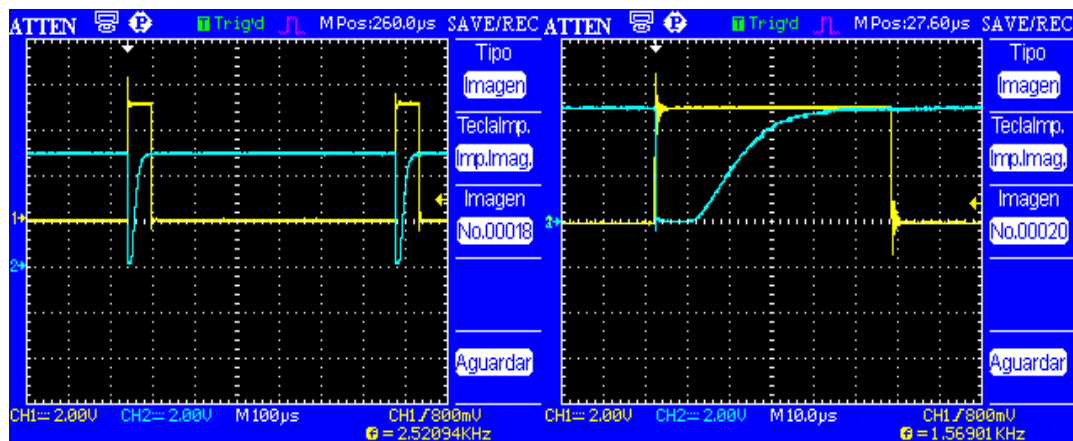


Fig. 3.10 - Señal obtenida al producirse un pulso

### 3.5 - Características principales del PIC 18F14K50

El microcontrolador que se ha elegido para la realización del proyecto es el PIC18F14K50, de la casa Microchip. Esta familia ofrece las ventajas de los microcontroladores PIC18, es decir, alto rendimiento de computación a un precio económico, con la adición de alta resistencia y memoria

'flash'. Además de estas características, la familia PIC18F1XK50/PIC18LF1XK50 presenta mejoras de diseño que hacen a estos microcontroladores una opción lógica para este proyecto, ya que se requiere un alto rendimiento.

Las características principales del microcontrolador son las siguientes:

**Compatibilidad USB:**

- a) USB 2.0
- b) Baja velocidad (1.5 Mb/s) y alta velocidad (12 Mb/s)
- c) Soporta transferencias de tipo control, interrupción, asíncrona y Bulk
- d) 256 bytes de memoria RAM para USB
- e) Detección de conexión USB mediante D+/D-
- f) Soporta hasta 16 Endpoints (8 bidireccionales)

**Modos de administración de energía:**

- a) Run: CPU on, periféricos on
- b) Idle: CPU off, periféricos on
- c) Sleep: CPU off, periféricos off
- d) Oscilador de arranque (2 velocidades)

**Periféricos:**

- a) 3 interrupciones externas programables
- b) 7 interrupciones independientes de cambios de voltaje de entrada
- c) 7 resistencias de pull-ups programables
- d) Módulo ECCP ( Enhanced Capture/Compare/PWM)
- e) Módulo MSSP (Master Synchronous Serial Port) que soporta comunicación SPI de 3 hilos y comunicación I2C™ con modo esclavo y maestro
- f) Módulo EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter).
- g) 9 canales de conversión Analógica-Digital, 10 bits de resolución
- h) Comparador analógico dual

**Estructura del oscilador**

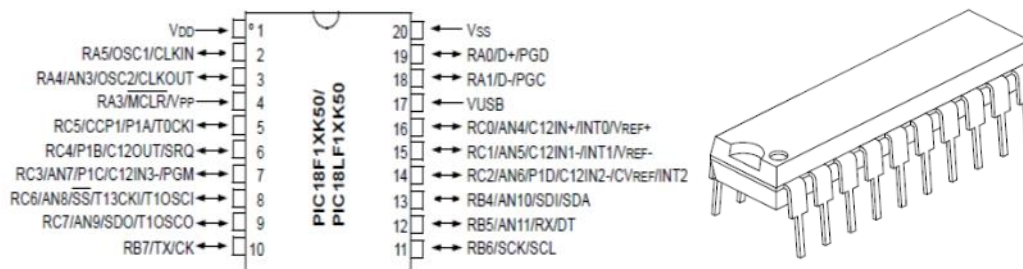
- a) Divisor de frecuencia para CPU
- b) 4 tipos de cristales resonadores (4 – 48 MHz)
- c) Oscilador interno (31 KHz – 16 MHz)
- d) 4X Phase Lock Loop (PLL)
- e) Oscilador secundario utilizando timer1 a 32 KHz
- f) Respaldo de frecuencia de reloj



### Características especiales del microcontrolador

- Arquitectura optimizada para compilación en C
- Niveles de prioridad para las interrupciones
- Perro guardián (WDT) de 4ms hasta 131s
- Auto programable bajo software de control
- Multiplicador hardware de 8 bits
- Alimentación 3V In-Circuit Serial Programming™ (ICSP™) usando 2 pines
- Rango de voltaje de alimentación (1,8V – 5,5V)
- Brown-out Reset (BOR) programable
- Dos velocidades de arranque

Device	Program Memory		Data Memory		I/O <sup>(1)</sup>	10-bit A/D (ch) <sup>(2)</sup>	ECCP (PWM)	MSSP		EUSART	Comp.	Timers 8/16-bit	USB
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I <sup>2</sup> C™				
PIC18F13K50/ PIC18LF13K50	8K	4096	512 <sup>(3)</sup>	256	15	11	1	Y	Y	1	2	1/3	Y
PIC18F14K50/ PIC18LF14K50	16K	8192	768 <sup>(3)</sup>	256	15	11	1	Y	Y	1	2	1/3	Y

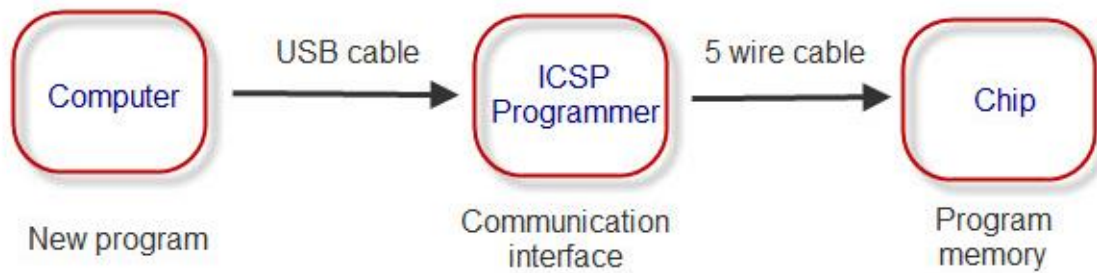


**Fig. 3.11 - Microcontrolador 18F14K50**

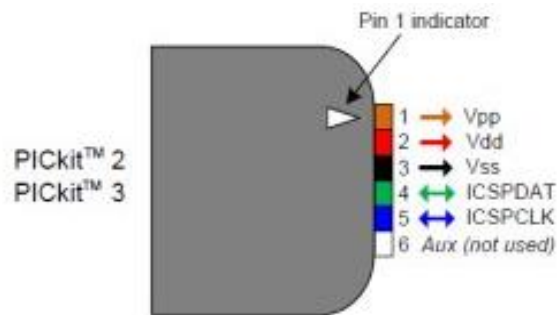
Para su conexión al PC a través del puerto USB se hace preciso seleccionar el conector adecuado según las recomendaciones del estándar.

De forma similar, para hacer uso de la funcionalidad ICSP, se hace preciso definir el conector adecuado.

Los PICs se programan utilizando cinco señales, dos de ellas dedicadas a la comunicación síncrona, y las tres líneas restantes destinadas a proporcionar la energía al microcontrolador. La señal de reloj está siempre controlada por el programador.



### Comunicación de programación típica



**Fig. 3.11 - Conectores ICSP**

Líneas que componen la programación por ICSP:

- *VPP* (PIN 1) es el voltaje necesario para preparar el PIC para que éste funcione en modo programación. Esta señal debe estar conectada al pin MCLR del micro; dependiendo de si su valor es GND o VDD, el microcontrolador se encuentra en modo programación o no.
- *VDD* (PIN 2) es la entrada de energía positiva al PIC.
- *VSS* (PIN 3) es la entrada de energía negativa al PIC y la referencia de 0 voltios para las señales restantes.
- *ICSPCLK* (PIN 5) es la línea de reloj de la comunicación serie. El cometido de esta línea es producir oscilaciones entre GND y VDD, cuya frecuencia es elegida por el programador.
- *ICSPDAT* (PIN 4) es la línea de datos serie. Su interfaz es bidireccional, por lo que los datos pueden ser enviados del PIC al programador y viceversa. Los cambios que se producen en la línea son entre GND y VDD, siendo producida la transmisión en el flanco descendente.

### 3.6 - Esquemático de la solución propuesta

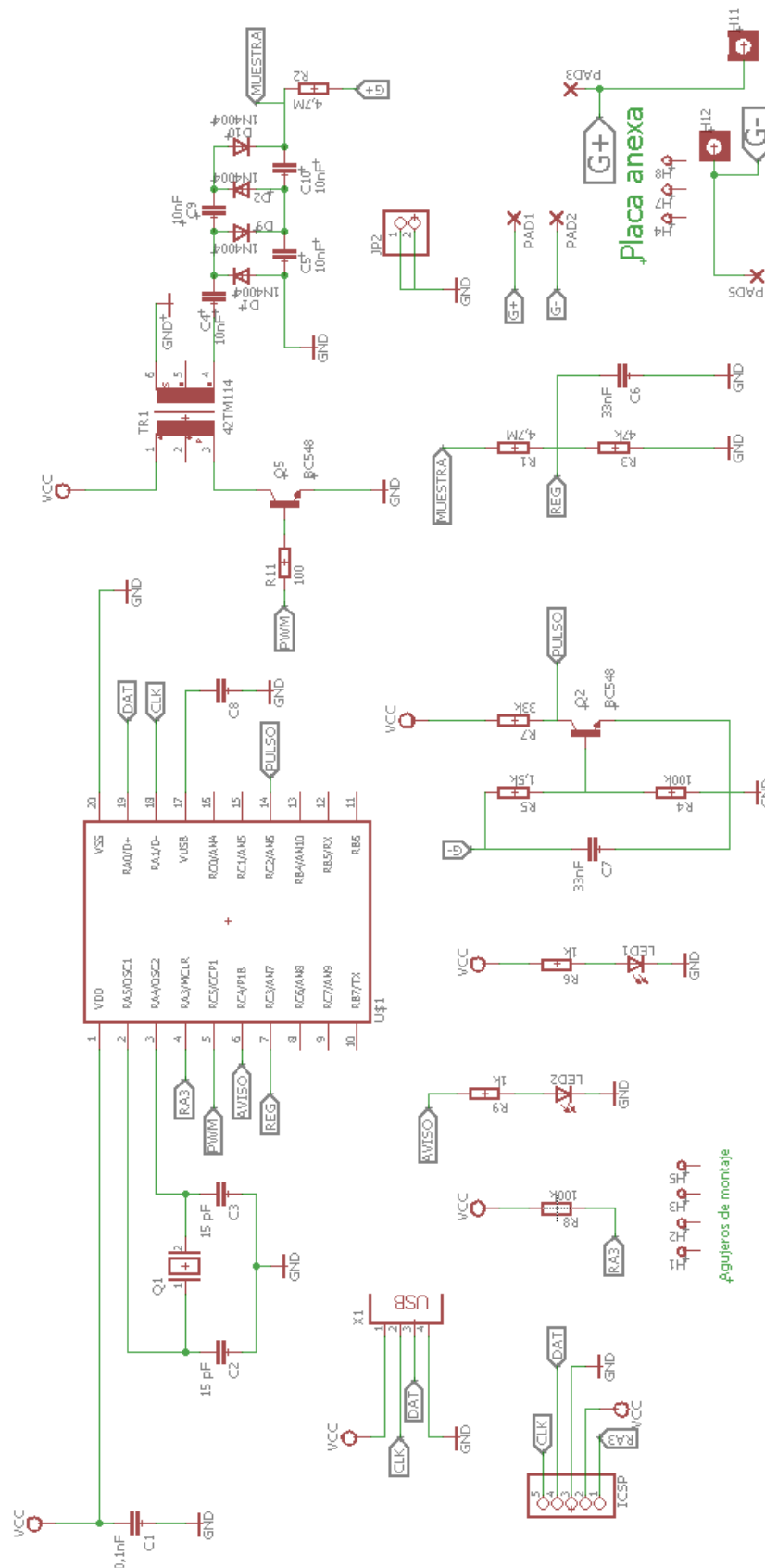


Fig. 3.12 - Esquemático de la solución propuesta

Como se puede observar en el esquema, el microcontrolador tiene conectado un cristal de 48 Mhz, que marca la frecuencia principal de trabajo, con sus respectivos condensadores.

La patilla /MCLR se encuentra en todo momento en nivel alto, con lo cual la placa ha de desconectarse de la fuente de alimentación si se desea resetearla. Por lo tanto, el PIC sólo es programable por ICSP.

Se ha configurado la patilla RC2 como interrupción de flanco de bajada para la adquisición de los pulsos, y la patilla RC5 como generadora de la señal PWM.

Se ha dispuesto la patilla RC4 como señal de aviso, la cual se activa cuando se exceda un número determinado de pulsos en un espacio de tiempo fijado. La patilla RC3 se ha establecido como convertidor A/D, función para ser utilizada en trabajos futuros.

Se ha colocado una tira de dos pines conectada a masa de la tarjeta, con objeto de poder realizar medidas dentro de la placa, y un diodo LED de funcionamiento.

El resto de los pines individuales se encargan de las conexiones de las dos tarjetas, así como del alojamiento del tubo en su posición.

# Capítulo 4

## Desarrollo software

### 4.1 - Entorno de desarrollo

Para la generación del firmware a incluir en el microcontrolador se ha utilizado una configuración similar a la que puede verse en la figura 4.1.

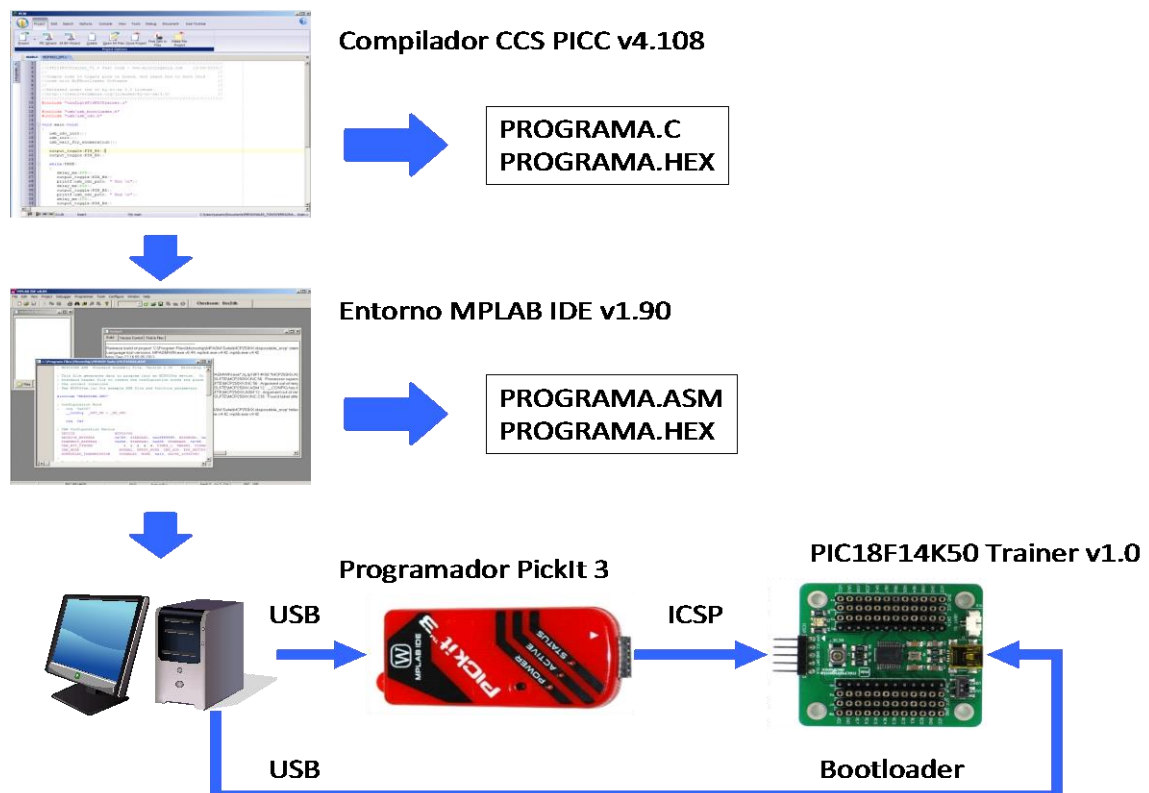


Figura 4.1 - Entorno de desarrollo

Para la programación del PIC se puede proceder de dos métodos diferentes.

- Primer procedimiento:

1º. Se genera el archivo *.ASM* con el programa MPLAB.

2º. El Pickitt 3 se conecta con el PC via USB y se carga en el dispositivo el archivo *.ASM*.

3º. El Pickitt se conecta a la placa de desarrollo mediante el conector ICSP y se vuelca el programa.

Una vez instalado el firmware del bootloader por primera vez en el microcontrolador, se puede proceder del segundo modo.

- Segundo procedimiento

1º. Se genera el archivo *.HEX* con el compilador CCS de C o con MPLAB.

2º. Se conecta el PC con la placa de desarrollo por USB.

3º. Se vuelca el archivo *.HEX* empleando el programa Bootloader.

#### 4.1.1 - Compilador CCS C

Para la programación del microcontrolador se ha empleado el compilador C de CCS, ya que éste incluye bibliotecas que incorporan determinados comandos que no son estándar, sino específicos de la familia de microcontroladores PIC, puesto que el lenguaje C estándar es demasiado genérico.



Fig. 4.2 - Logotipo CCS

Este compilador posee varias directivas de preprocesado, que no dispone el lenguaje genérico C:

- Directivas relacionadas con la especificación del dispositivo.
- Directivas de cualificación de funciones.
- Directivas de control del compilador.
- Directivas de control de la memoria del microcontrolador.

Este compilador proporciona, además, una gran cantidad de funciones incorporadas para acceder y manipular los periféricos del micro, esto hace mucho más sencillo su configuración, sin verse obligado el usuario a trabajar con los registros asociados a cada funcionalidad.

Además de todo lo señalado anteriormente, este entorno de desarrollo incluye una selección de dispositivos (Device Selection Tool), cuya base de datos incluye todos los dispositivos que puede programar el compilador, incluyendo todas sus características hardware.

Por estas razones se ha decidido utilizar este entorno de desarrollo para realizar todo el software programado en el PIC.

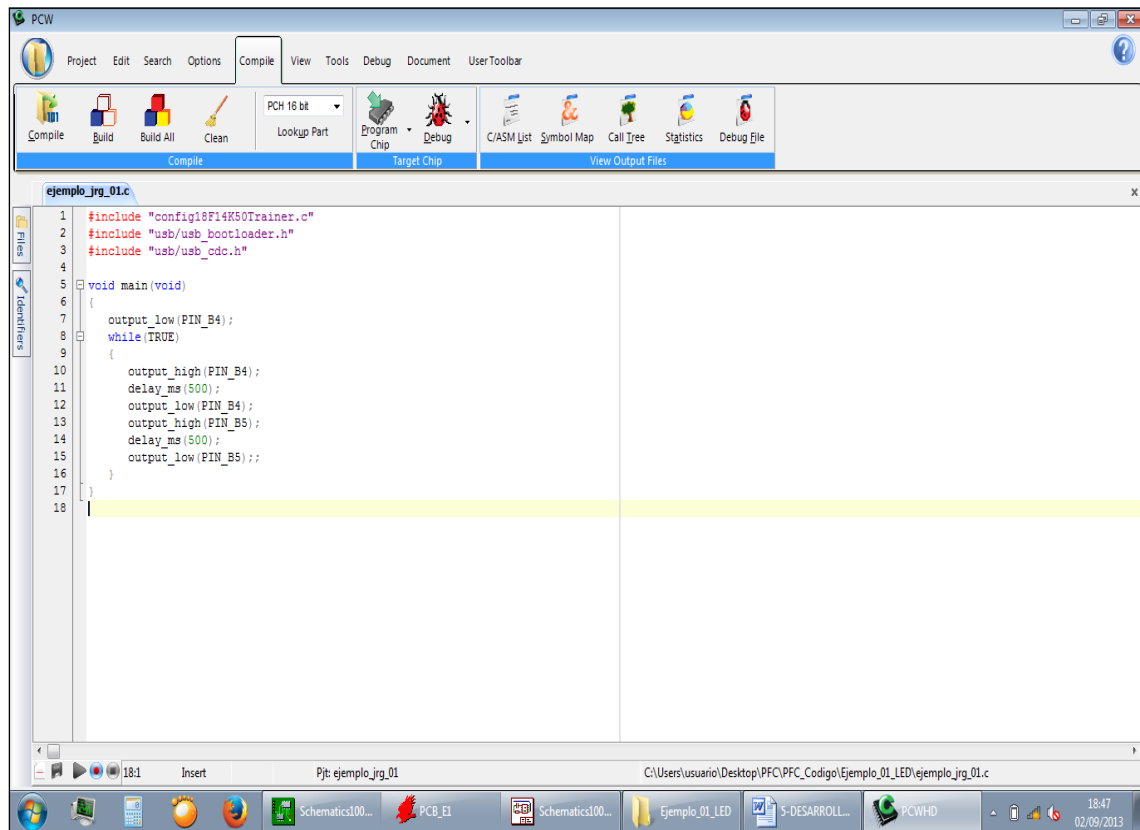


Fig. 4.3 - Entorno de desarrollo CCS

#### 4.1.2 - Placa de desarrollo 18F14K50

Para la toma de contacto con el microcontrolador y el diseño del software se ha empleado la placa de desarrollo 18F14K50 Trainer1.0, realizada por Microingenia S.L., cuyo fin es la implementación rápida de diferentes sistemas en el que se requiera el manejo de este microcontrolador.

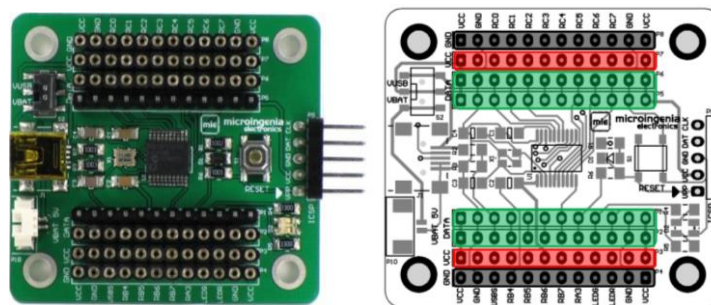


Fig. 4.4 - Placa de desarrollo 18F14K50 Trainerv1.0

➤ **Características generales**

- a) PIC 18F14K50
- b) 48 MHz CPU
- c) 16 KB de FlashROM
- d) 256 Bytes de EEPROM
- e) 768 Bytes de RAM
- f) 17 Entradas/Salidas Digitales
- g) 9 Entradas Analógicas
- h) LED bicolor
- i) Conector ICSP para conexión con el grabador/depurador Pickit 2/3
- j) Conector USB, para alimentación y programación
- k) Interruptor de Reset

➤ **Especificaciones**

- a) Alimentación: 5 VCC (USB o Vbat)
- b) Dimensiones: 50,8 x 43,18mm (2" x 1.7")
- c) Peso: 15,9g

## 4.2 - Pruebas iniciales

Con el fin de poder compilar programas para la plataforma elegida (PIC18F14K50), se ha precisado incorporar un fichero de configuración especialmente programado a los fines de este proyecto. Para ello se han incluido los ficheros de cabecera propios del microcontrolador, obteniéndose el siguiente código de configuración resultante.

```
#include <18F14K50.h>
#fuses
HS, NOWDT, NOPROTECT, NOLVP, NODEBUG, NOBROWNOUT, USBDIV1, PLEN, CPUDIV1, PUT, MCLR
#use delay(clock=4800000)
#use PWM(OUTPUT=PIN_C5, DUTY=0)
```

### Configuración del PIC

La estructura de este fichero es:

- ✓ La primera línea del código hace referencia al microcontrolador que va a ser utilizado en este proyecto (18F14K50).
- ✓ La segunda línea se encarga de las directivas de preprocesado del microcontrolador.



Comando	Explicación
<b>HS</b>	Habilita el uso de reloj externo de alta velocidad
<b>NOWDT</b>	Deshabilita el uso del perro guardián
<b>NOPROTECT</b>	Deshabilita la protección del código
<b>NOLVP</b>	Deshabilita Low Voltage ICSP Programming
<b>NODEBUG</b>	Deshabilita la depuración en línea
<b>NOBROWNOUT</b>	Deshabilita reset del PIC por caída de voltaje
<b>USBDIV1</b>	Divisor de frecuencia USB igual a 1
<b>PLLEN</b>	Habilita el uso de PLL
<b>CPUDIV1</b>	Divisor de frecuencia CPU igual a 1
<b>PUT</b>	Power Up Timer
<b>MCLR</b>	Habilita pin de reset

- ✓ La tercera línea configura la frecuencia a la que trabaja el microcontrolador, en este caso trabaja a 48MHz, velocidad necesaria para lo operación de las comunicaciones USB.
- ✓ La cuarta línea configura la patilla de la señal de PWM.

#### 4.2.1 - Encendido/apagado de LED

Como primer paso para la puesta a punto del entorno de desarrollo CCS, se ha planteado un problema clásico en la programación de microcontroladores, el encendido y apagado temporizado de un LED. Para ello, se han empleado los diodos LEDs incluidos en la placa de desarrollo del microcontrolador 18F14K50.

Con este ejercicio se ha pretendido comprobar la correcta configuración del microcontrolador, así como de las características básicas de la implementación del lenguaje C bajo el compilador CCS.

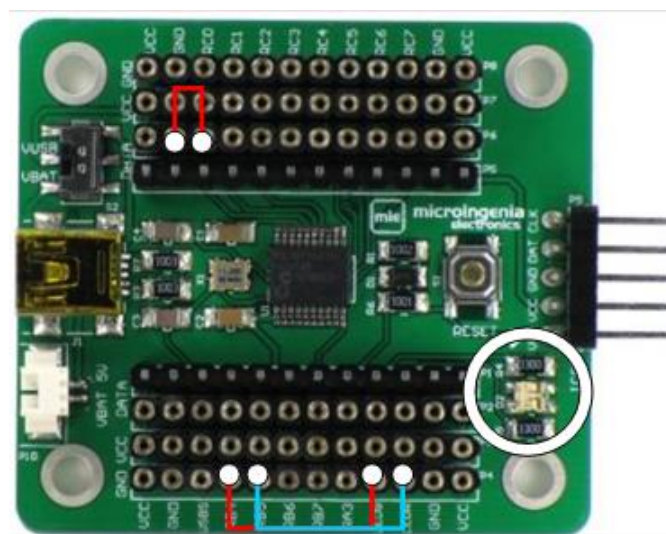
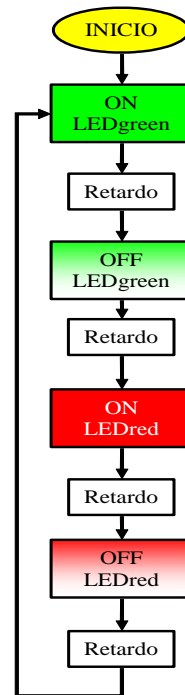


Fig. 4.5 - Conexiones en placa 4.2.1



El funcionamiento del programa consiste en el encendido y apagado de cada uno de los citados LEDs de forma sucesiva. El sistema seguirá ejecutándose hasta que se presione el botón de reset de la placa de desarrollo, o bien se desconecte de su fuente de energía.

```

#include "config18F14K50Trainer.c"
#include "usb/usb_bootloader.h"

void main(void)
{
  while (TRUE)
  {
    output_high(PIN_B4);
    delay_ms(500);
    output_low(PIN_B4);
    output_high(PIN_B5);
    delay_ms(500);
    output_low(PIN_B5);
  }
}

```

#### Primer programa encendido/apagado LED

Esta primera forma de programación, la más sencilla, temporiza el retardo mediante el uso de la función 'delay', en este caso, el argumento es de 500ms.

```

#include "config18F14K50Trainer.c"
#include "usb/usb_bootloader.h"

void main(void)
{
  disable_interrupts(GLOBAL);
  disable_interrupts(INT_TIMER0);
  setup_timer_0(RTCC_DIV_256);
  set_timer0(18661); //Valor de precarga para que el contador timer0
cuente 0.5 segundos
  enable_interrupts(GLOBAL);
}

```

```

enable_interrupts (INT_TIMER0);

output_low (PIN_B4);
output_high (PIN_B5);

while (TRUE) {}
}
#int_timer0
void isr_timer0 (void)
{
output_toggle (PIN_B4);
output_toggle (PIN_B5);
}

```

### Segundo programa encendido/apagado LED

Este segundo programa, algo más complejo, hace uso de contadores e interrupciones.

El programa comienza con un LED apagado y otro encendido; cuando el contador se desborda se produce la interrupción por desbordamiento del contador timer0, pasando a ejecutarse su rutina de interrupción. En esta interrupción se lleva a cabo un 'toggle' (pasar de nivel alto a nivel bajo y viceversa) de los dos pines a los que están conectados los LEDs.

El cálculo del valor de precarga es el siguiente:

Tiempo = 0,5 s

Frecuencia del  $\mu\text{C}$  = 48 MHz

Divisor frecuencia =  $128 \cdot 4$

Frecuencia resultante = 93750 Hz

Periodo resultante = 10,67  $\mu\text{s}$

Conteo necesario =  $500000 / 10,67 \mu = 46.875$

Temporización efectiva =  $46875 \cdot 10,67 \mu\text{s} = 500156,25 \mu\text{s} = 500,156\text{ms}$

Valor de precarga =  $65536 - 46875 = 18661$

## 4.2.2 - Comunicación USB

El objetivo de este programa ha sido la comprensión de la comunicación USB, así como de los comandos necesarios para su uso de forma óptima.

Los comandos propios de la comunicación USB son:

usb_cdc_init ()	Configura los baudios, bit paridad etc, de la comunicación USB
usb_init ()	Inicializa el hardware del USB
usb_wait_for_enumeration ()	Espera hasta que el PicUSB sea configurado por el host

Se utiliza como fichero base el algoritmo de programación del punto 4.2.1, de encendido de diodos LEDs (primer ejemplo).

Tomando como base dicho programa, se lleva a cabo el envío, vía USB, de diferentes cadenas de caracteres dependiendo de la acción que se lleva a término en cada momento.

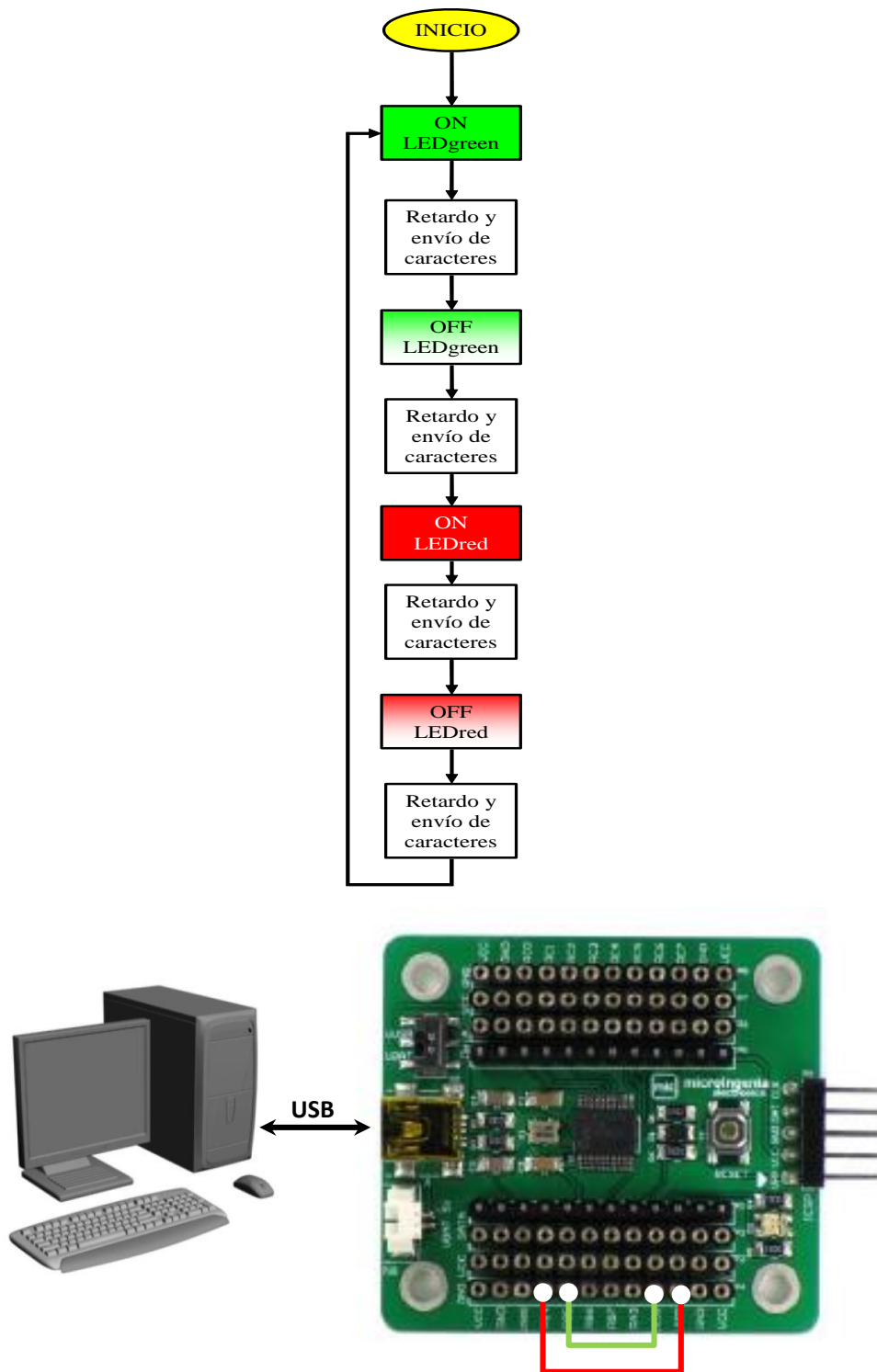


Fig. 4.6 - Conexiones placa de desarrollo 4.2.2

```

#include "config18F14K50Trainer.c"
#include "usb/usb_bootloader.h"
#include "usb/usb_cdc.h"
  
```

```
void main()
{
    usb_cdc_init();
    usb_init();
    usb_wait_for_enumeration();

    output_low(PIN_B4);
    while(TRUE)
    {
        output_high(PIN_B4);
        printf(usb_cdc_putc, "Encendido LED verde\n\r");
        delay_ms(500);
        output_low(PIN_B4);
        printf(usb_cdc_putc, "Apagado LED verde\n\r");
        output_high(PIN_B5);
        printf(usb_cdc_putc, "Encendido LED rojo\n\r");
        delay_ms(500);
        output_low(PIN_B5);
        printf(usb_cdc_putc, "Apagado LED verde\n\r");
    }
}
```

### Programa de comunicación USB

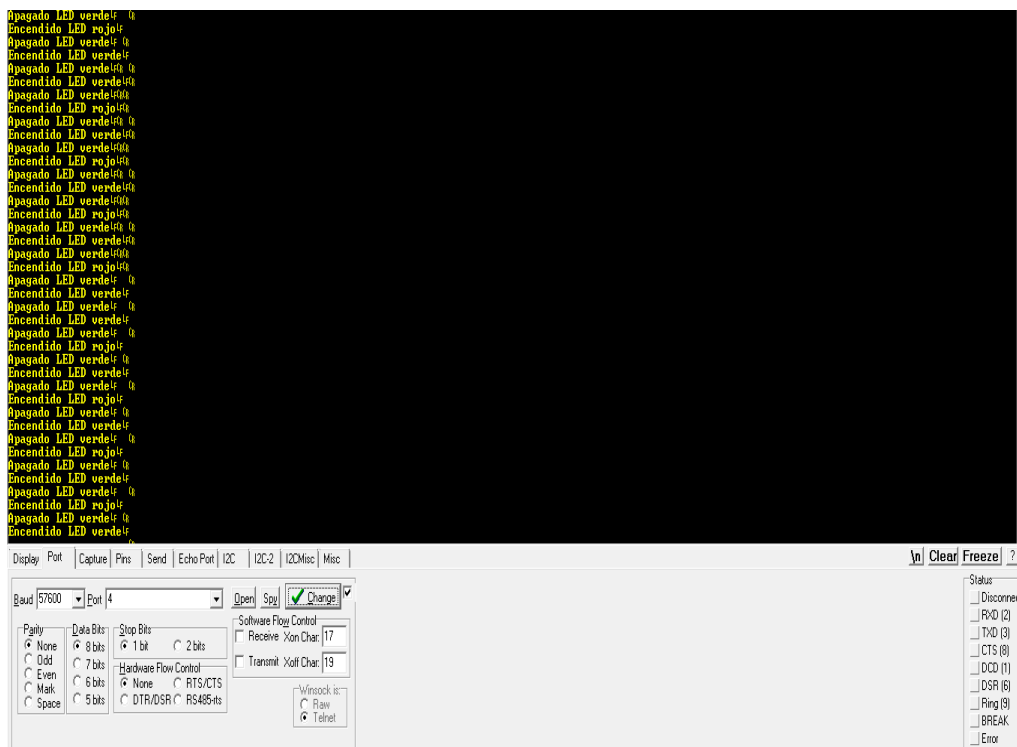


Fig. 4.7 - Envío de cadena de caracteres por USB

En esta imagen (fig. 4.7), se puede apreciar que el programa ha enviado la cadena de caracteres, encendido 'x' o apagado 'y', según la acción que esté realizando el microcontrolador en ese instante.



**Fig. 4.8 - Logotipo de Realterm®**

La imagen 4.7 es una captura de pantalla hecha en un momento cualquiera de la ejecución del software, que está siendo recibida por el PC mediante el programa Realterm ([www.realterm.sourceforge.net](http://www.realterm.sourceforge.net)), el cual refleja los datos en formato ASCII.

### 4.2.3 - Generación de señal PWM

Con este programa se ha pretendido generar la señal PWM necesaria para hacer funcionar el tubo Geiger-Muller a través del circuito elevador de tensión.

Esto se ha llevado a cabo mediante el uso de las funciones PWM (Pulse Width Modulation) del módulo CCP (Capture Compare PWM). Primero, se ha configurado el módulo CCP en modo PWM, segundo, se ha establecido el funcionamiento del timer\_2, timer asociado al PWM, y por último, se ha habilitado el funcionamiento del módulo como PWM.

La señal que se ha pretendido conseguir es de 50KHz de frecuencia con una relación de trabajo del 30%, como puede comprobarse en la señal del osciloscopio mostrada en la imagen 4.10.

```
#include "config18F14K50Trainer.c"
#include "usb/usb_bootloader.h"
#include "usb/usb_cdc.h"
#USE PWM(OUTPUT=PIN_C5, DUTY=30)

void main(void)
{
    setup_ccp1(CCP_PWM);
    setup_timer_2(T2_DIV_BY_1, 119, 2);
    pwm_on();

    while(TRUE)
    {
    }
}
```

**Programa de generación de PWM**

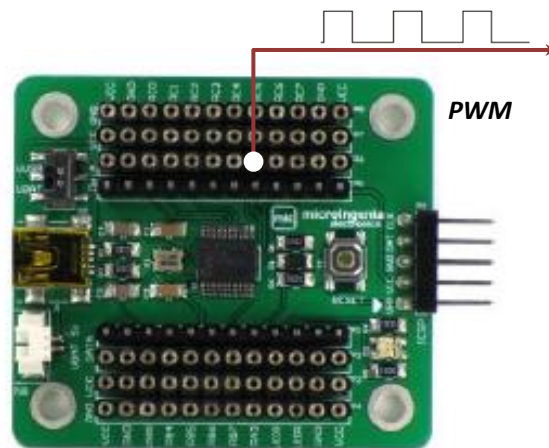


Fig. 4.9 - Conexiones placa de desarrollo 4.2.3

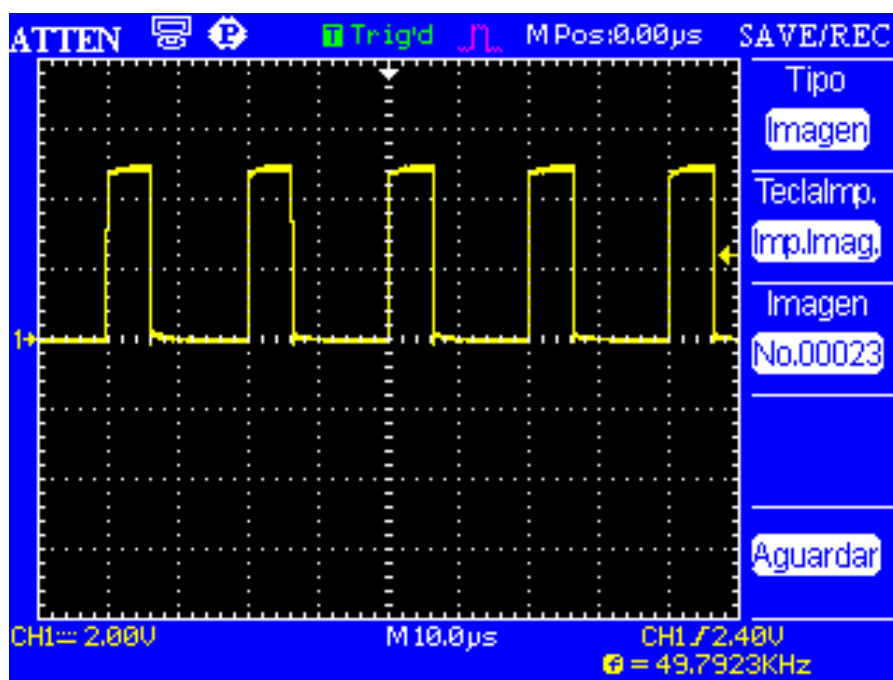


Fig. 4.10 - Captura de la señal PWM

#### 4.2.4 - Medición de CPS (Ciclos por segundo) y CPM (Ciclos por minuto)

El objetivo de este programa ha sido la actualización y el envío de los dos arrays usados por el programa, el array de ciclos por segundo y el de ciclos por minuto.

EL funcionamiento del programa consiste en cuantificar las veces que se produce una interrupción por flanco de bajada en 100 milisegundos en C2. Este valor es almacenado en un array de 10 posiciones que contabiliza los pulsos medidos dentro de un segundo. Este parámetro se almacena y se sobrescribe en el interior del array de manera sucesiva y, una vez alcanzado la última posición se vuelve a sobrescribir el primer valor. Cada vez que transcurre un segundo (10 posiciones), la magnitud de la variable que contabiliza los pulsos medidos dentro de un segundo es a su vez almacenado en un array de 60 posiciones, utilizado para medir los pulsos medidos en

un tiempo de un minuto. Este valor, al igual que el anterior, se almacena en el interior del array de manera sucesiva y, una vez alcanzado la última posición, se vuelve a sobrescribir la primera posición.

Siempre que se produce una interrupción por haber transcurrido 100 milisegundos se envían la cantidad de pulsos dentro de esos 100 ms, los pulsos del segundo anterior y los pulsos del minuto anterior.

El cálculo del valor de precarga del timer0:

Tiempo = 0,1 s

Frecuencia del  $\mu\text{C}$  = 48 MHz

Divisor frecuencia =  $128 \cdot 4$

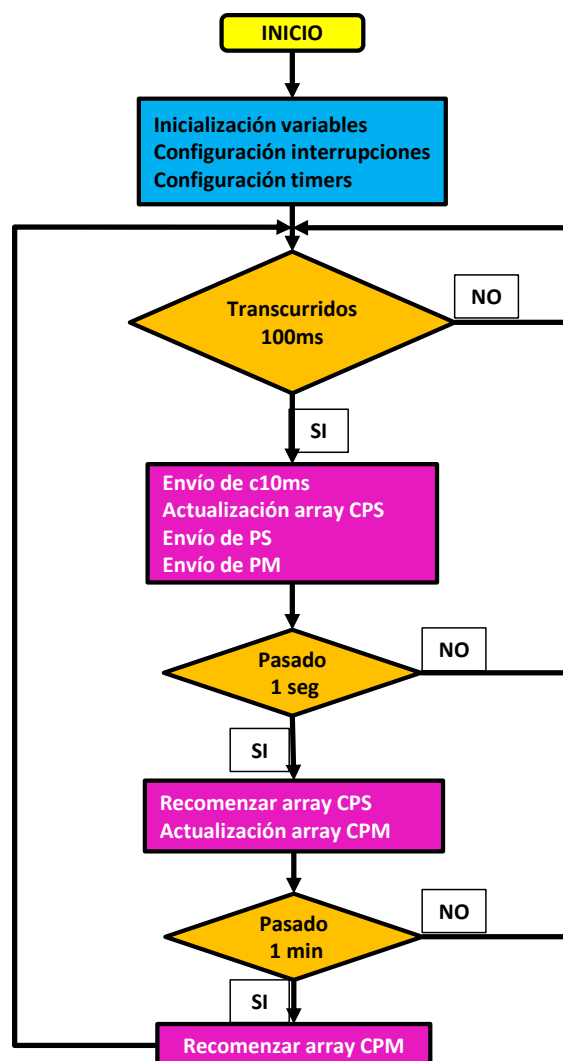
Frecuencia resultante = 93750 Hz

Periodo resultante = 10,67  $\mu\text{s}$

Conteo necesario =  $100000 \mu / 10,67 \mu = 9.375$

Temporización efectiva =  $9375 \cdot 10,67 \mu\text{s} = 100031,25 \mu\text{s} = 100,031 \text{ms}$

Valor de precarga =  $65536 - 9375 = 56161$



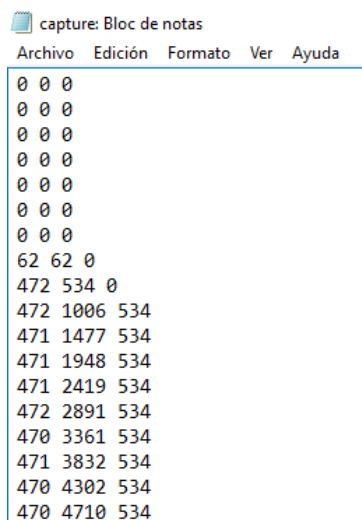




```
PS=PS+CMS[j];
//Se evalua si se enciende LED rojo
if(c100ms>100)output_high(PIN_C4);
else output_low(PIN_C4);
//Código para enviar variables
printf(usb_cdc_putc,"%Lu %Lu %Lu \r\n",c100ms,PS,PM);
c100ms=0;
a=0;
j++;
//Cada 10 veces 100ms se ejecuta esta parte del - 41 -ódigo
//Se elimina el valor antiguo de cuentas por segundo del array
//Se sustituye por el nuevo valor de cuenta por segundo del array
//Se actualizan variables
if (j==10)
{
    j=0;
    PM=PM-CPS[i];
    CPS[i]=PS;
    PM=PM+CPS[i];
    i++;
    if (i==60) i=0;
}
}
}
#int_timer0
void isr_timer0(void)
{
    a=1;
    set_timer0(56161);
}
#INT_EXT2
void interrupt_RC2()//llega pulso a patilla C2
{
    c100ms=c100ms+1;
}
```

#### Programa de medición de CPS y CPM

La recepción de los datos generados por el programa se ha realizado con Realterm, el cual ha generado un archivo de texto (imagen 4.12) donde han sido almacenados dichos datos. Estos datos se han exportado a un archivo Excel para su representación en forma gráfica, tal como se puede observar en la imagen 4.13.



capture: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
62 62 0
472 534 0
472 1006 534
471 1477 534
471 1948 534
471 2419 534
472 2891 534
470 3361 534
471 3832 534
470 4302 534
470 4710 534
```

Fig. 4.12 - Muestra del archivo de texto generado

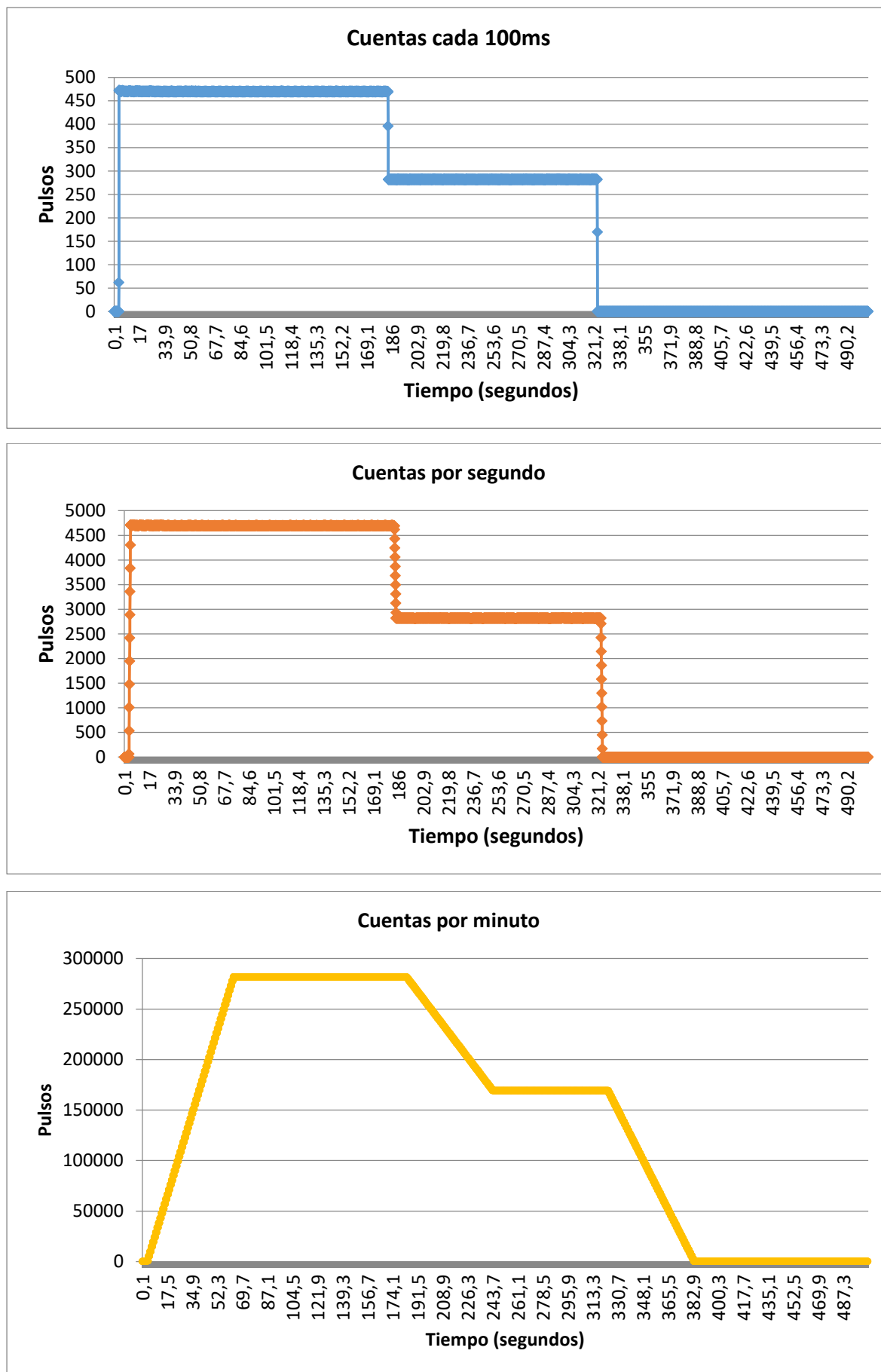


Fig. 4.13 - C100ms, CPS y CPM

## 4.2.5 - Comunicación USB de control

El objetivo de este ejercicio ha sido controlar por teclado la señal de PWM generada por el microcontrolador, tanto en frecuencia como en relación de trabajo.

Es posible (por estructura del programa) regular la frecuencia entre los 47KHz a 150KHz, pues de esta manera el transformador se encuentra siempre trabajando fuera del espectro de frecuencias audible. La señal PWM puede oscilar entre el 0% y el 90%, no recomendado trabajar con porcentajes superiores.

El funcionamiento del programa se basa en el gobierno de la señal PWM mediante una estructura de control de 4 caracteres ASCII. Esta estructura se compone de un array de 4 posiciones, el cual se modifica rotando los caracteres a la izquierda cada vez que se introduce un carácter por USB, almacenando en la última posición del array el carácter introducido por teclado. El primer valor y el último del array son valores de control, los cuales se ha establecido que si su cuantía equivale al carácter "ú" (250 en ASCII) se actualizan los valores de frecuencia (segundo valor) y de relación de trabajo (tercer valor), siempre que estos sean diferentes de los ya previamente fijados.

En el anexo F se muestra la forma en la que se calculan los valores enteros que controlan la señal PWM, así como de una tabla con ejemplos ya calculados. En el anexo G se dispone de una tabla de valores ASCII para la consulta de los respectivos caracteres con su equivalente en entero decimal.

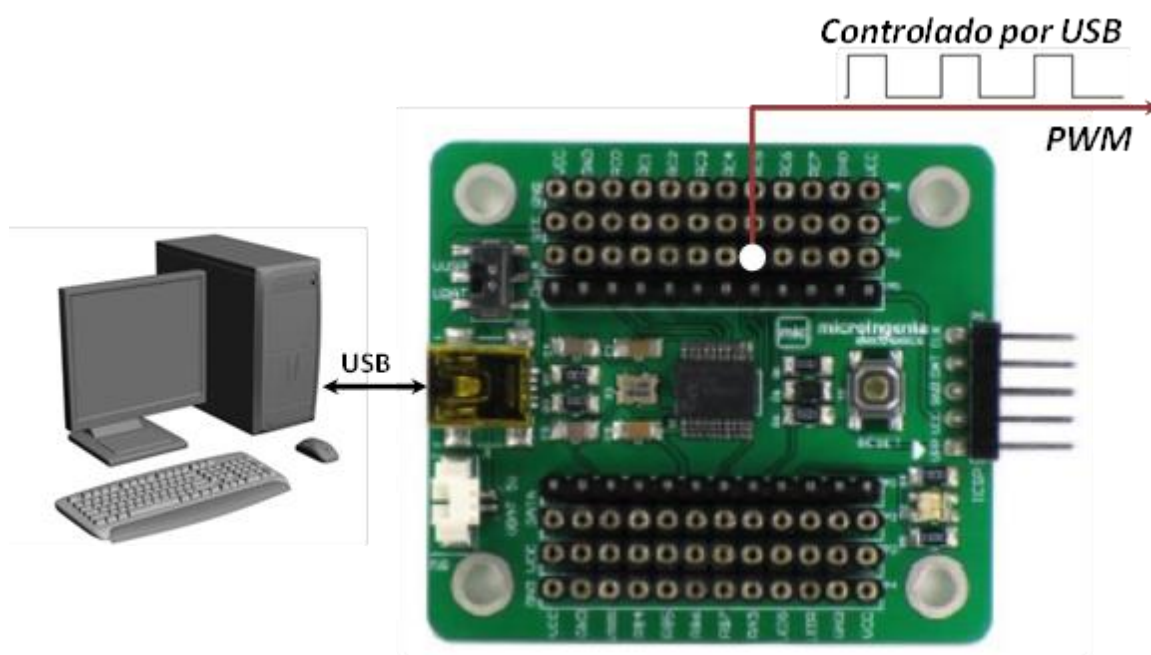
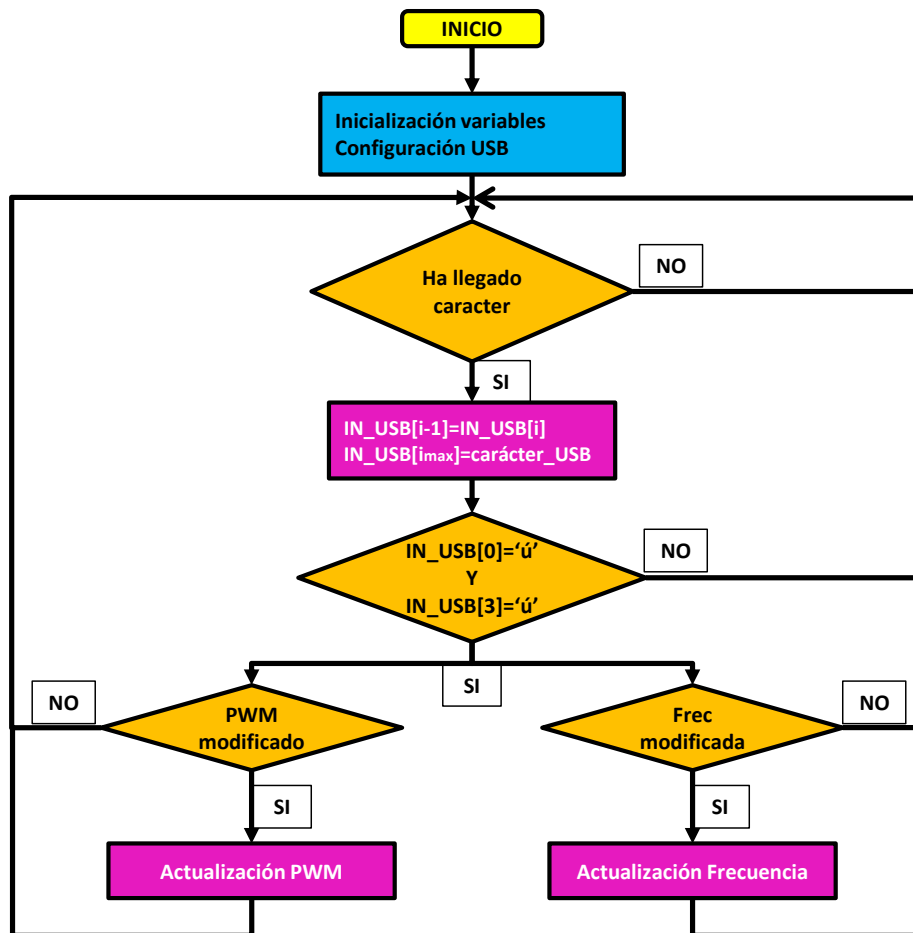


Fig. 4.14 - Conexiones placa de desarrollo 4.2.5



```

#include "config18F14K50Trainer.c"
#include "usb/usb_bootloader.h"
#include "usb/usb_cdc.h"
#USE PWM(OUTPUT=PIN_C5, DUTY=0)

//Variable control PWM
int8 pwm=0;
//variable control frecuencia
int8 periodo=0;
//variables comunicacion USB
int8 in_USB[4]={0,0,0,0};

void main(void)
{
  usb_cdc_init();
  usb_init();
  usb_wait_for_enumeration();

  setup_ccp1(CCP_PWM);
  setup_timer_2(T2_DIV_BY_1,0,1);
  pwm_on();

  while(true)
  {
    if(usb_cdc_kbhit())
    {

```

```

in_USB[0]=in_USB[1];
in_USB[1]=in_USB[2];
in_USB[2]=in_USB[3];
in_USB[3]=usb_cdc_getc();

if(in_USB[0]=='ú' && in_USB[3]=='ú')
{
    //Si ha cambiado el valor de la frec se actualiza
    if(- 45 -ódigo != in_USB[1])
    {
        - 45 -ódigo = in_USB[1];
        setup_timer_2(T2_DIV_BY_1,- 45 -ódigo,1);
    }
    //Si ha cambiado el valor del pwm se actualiza
    if(pwm != in_USB[2])
    {
        pwm = in_USB[2];
        set_pwm1_duty(pwm);
    }
    printf(usb_cdc_putc, "- 45 -ódigo = %u\r\n" ,in_USB[1]);
    delay_ms(1);
    printf(usb_cdc_putc, "pwm = %u\r\n" ,in_USB[2]);
}
}
}
}

```

#### Programa de comunicación USB de control

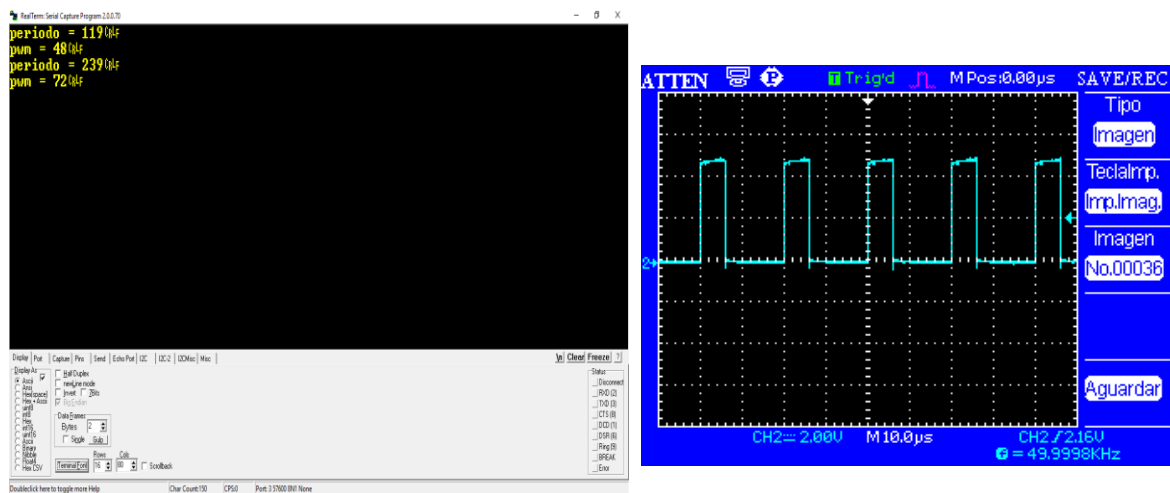
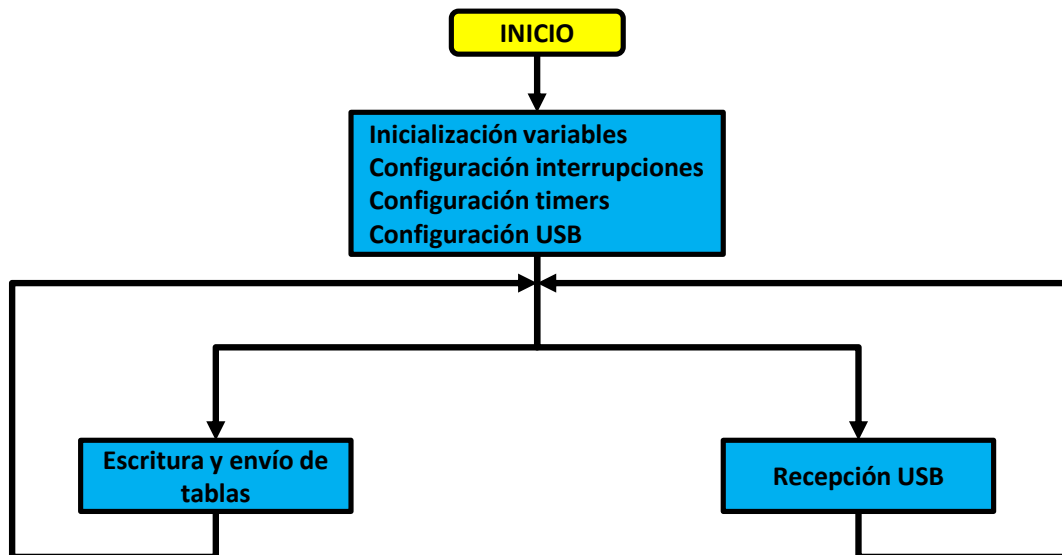


Fig. 4.15 - Control de la señal de PWM

### 4.3 - Integración del software final

Este programa ha sido el utilizado en la programación del microcontrolador, ya que realiza todas las tareas necesarias en el proyecto.

El proyecto posee la siguiente estructura de programación:

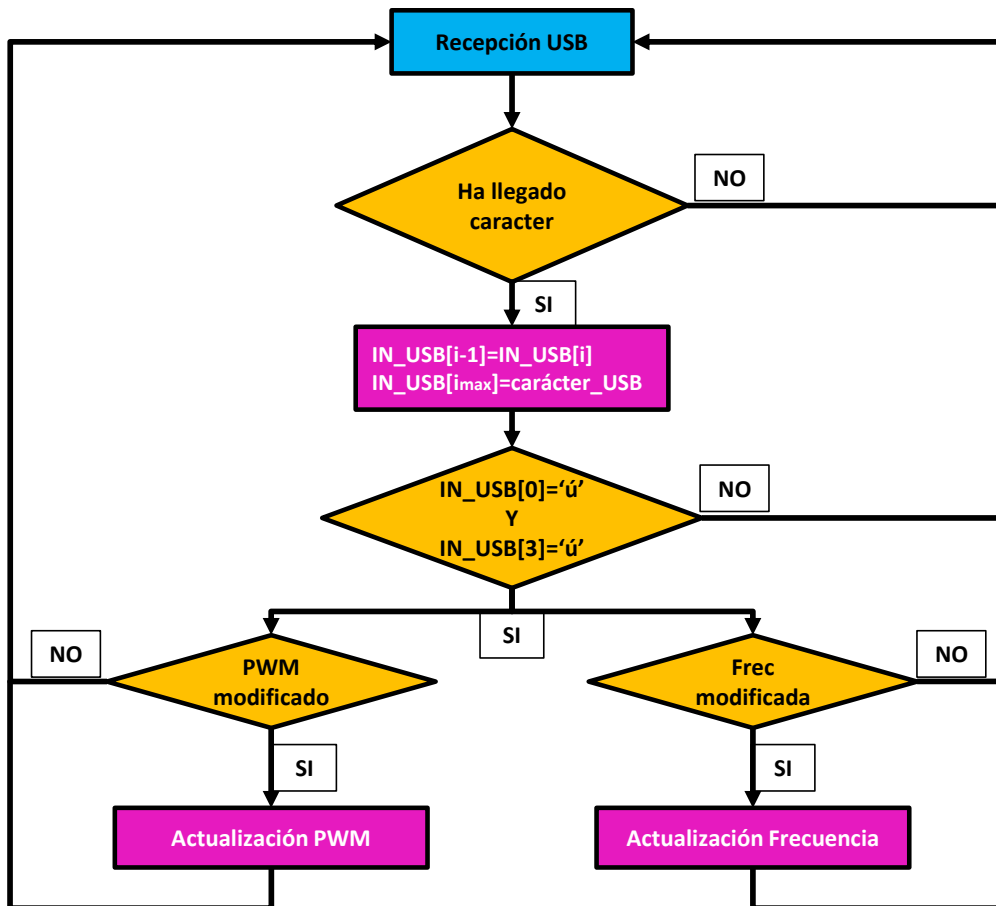


El programa comienza por la configuración de todos los parámetros presentes en el microcontrolador y la inicialización y declaración de todas las variables utilizadas en el programa del PIC.

Acto seguido se ejecuta la recepción de caracteres por USB y la escritura y envío de las tablas de valores generadas por el tubo en paralelo y de forma simultánea.

Hasta que no se finaliza la escritura de un valor en las tablas correspondientes no es posible la recepción de un carácter para la configuración de la señal PWM, y viceversa, no es posible la escritura de un valor en las tablas hasta que no se ha gestionado la recepción de un carácter.

El funcionamiento de 'Recepción USB' es el que se contempla en el siguiente diagrama de flujo.



En primer lugar se evalúa si se ha recibido un carácter por USB, y si es así, se rota a la izquierda los valores del array de control (4 posiciones) y se almacena el carácter recibido en la última posición de este array.

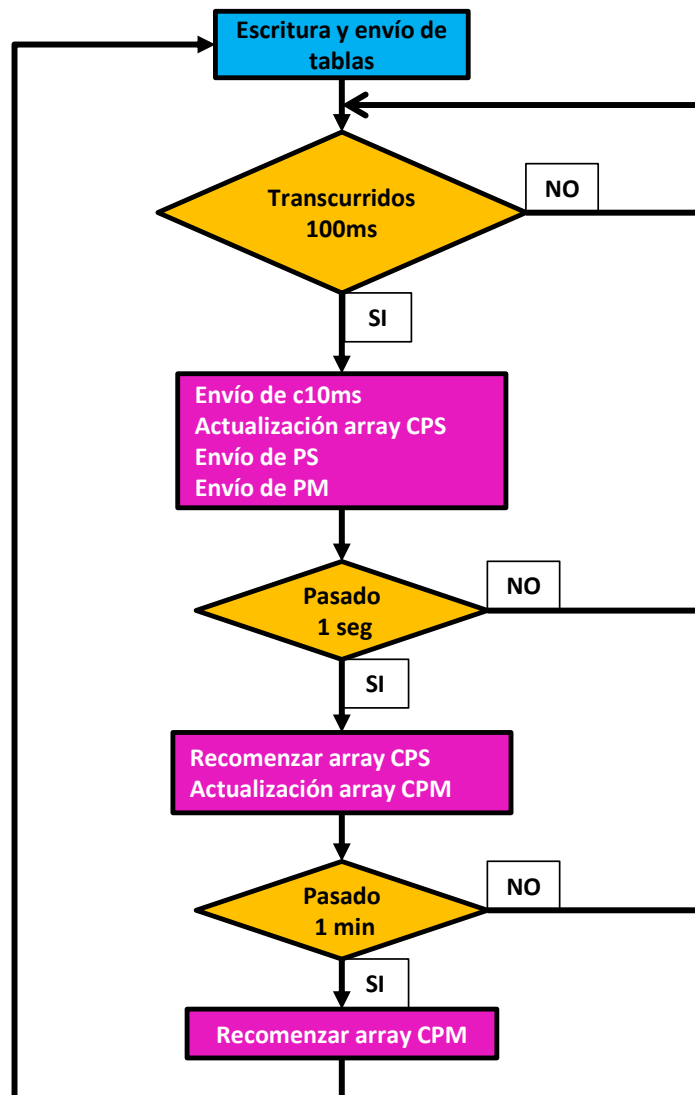
Acto seguido se evalúa si la estructura del array es la correcta (primer y último valor del array control iguales a 250), y de ser la adecuada, se procede a actualizar la frecuencia (segundo valor) y la relación de trabajo de la señal de PWM generada por el PIC (tercer valor), siempre que estos sean diferentes de los ya previamente fijados.

Se ha fijado que la frecuencia de la señal sea en todo momento de 50.000 Hertzios y que la relación de trabajo no supere el 45%, ya que de esta forma el voltaje a la salida de la etapa de elevación de tensión no se supera el valor máximo de voltaje que puede soportar este tubo Geiger.

En el anexo F se muestra la forma en la que se calculan los valores enteros que controlan la señal PWM, así como de una tabla con ejemplos ya calculados. En el anexo G se dispone de una tabla de valores ASCII para la consulta de los respectivos caracteres con su equivalente en entero decimal.



El modo de funcionamiento de 'Escritura y envío de tablas', incluido en el diagrama de flujo principal, se explica en el siguiente flujograma.



El objetivo de este programa es la actualización de los dos arrays usados por el programa, el array de ciclos por segundo y el de ciclos por minuto, y el envío de los pulsos dentro de 100 ms, dentro de 1 segundo y dentro de 1 minuto.

En primer lugar se evalúa si han transcurrido 100 milisegundos, y si es así, se almacenan las veces que se produce una interrupción por flanco de bajada en C2 en este tiempo.

El cálculo del valor de precarga del timer0 para temporizar 100 milisegundos es:

Tiempo = 0,1 s

Frecuencia del  $\mu\text{C}$  = 48 MHz

Divisor frecuencia =  $128 \cdot 4$

Frecuencia resultante = 93750 Hz

Periodo resultante = 10,67  $\mu\text{s}$

Conteo necesario =  $100000 \mu / 10,67 \mu = 9.375$



```

//Valor de precarga para que el contador timer0 cuente 0.1 segundos
setup_timer_0(RTCC_DIV_128|RTCC_INTERNAL);
set_timer0(56161);

//Configuracion de la señal PWM
setup_ccp1(CCP_PWM);
setup_timer_2(T2_DIV_BY_1,0,1);
pwm_on();

while(TRUE)
{
//Si se - 50 -ódigo caracteres y se guardan en array in_buffer
//Si el formato de la - 50 -ódigo- 50 -n- 50 -ión es correcto
//Se actualiza la señal PWM en frecuencia y - 50 -ódigo- 50 -n de
trabajo
if(usb_cdc_kbhit())
{
in_buffer[0]=in_buffer[1];
in_buffer[1]=in_buffer[2];
in_buffer[2]=in_buffer[3];
in_buffer[3]=usb_cdc_getc();
if(in_buffer[0]==250&&in_buffer[3]==250)
{
//Si ha cambiado el valor de la frec se actualiza
if(- 50 -ódigo!=in_buffer[1] && in_buffer[1]==239)
{
- 50 -ódigo= in_buffer[1];
setup_timer_2(T2_DIV_BY_1,- 50 -ódigo,1);
}
//Si ha cambiado el valor del pwm se actualiza
if(pwm!=in_buffer[2] && in_buffer[2]<95)
{
pwm=in_buffer[2];
set_pwm1_duty(pwm);
}
}
}
//Cada 100ms se ejecuta esta parte del - 50 -ódigo
if (a==1)
{
//Se elimina el valor antiguo de c100ms del array
//Se sustituye por el nuevo valor de c100ms del array
//Se actualizan variables
PS=PS-CMS[j];
CMS[j]=c100ms;
PS=PS+CMS[j];
//Se evalua si se enciende LED rojo
if(c100ms>100)output_high(PIN_C4);
else output_low(PIN_C4);
//Código para enviar variables
printf(usb_cdc_putc,"%Lu %Lu %Lu \r\n",c100ms,PS,PM);
c100ms=0;
a=0;
j++;
//Cada 10 veces 100ms se ejecuta esta parte del - 50 -ódigo
//Se elimina el valor antiguo de cuentas por segundo del array
//Se sustituye por el nuevo valor de cuenta por segundo del array
//Se actualizan variables
if (j==10)

```

```

        {
            j=0;
            PM=PM-CPS[i];
            CPS[i]=PS;
            PM=PM+CPS[i];
            i++;
            if (i==60) i=0;
        }
    }
}

#int_timer0
void isr_timer0(void)
{
    a=1;
    set_timer0(56161);
}

#INT_EXT2
void interrupt_RC2()//llega pulso a patilla C2
{
    c100ms=c100ms+1;
}

```

Software final

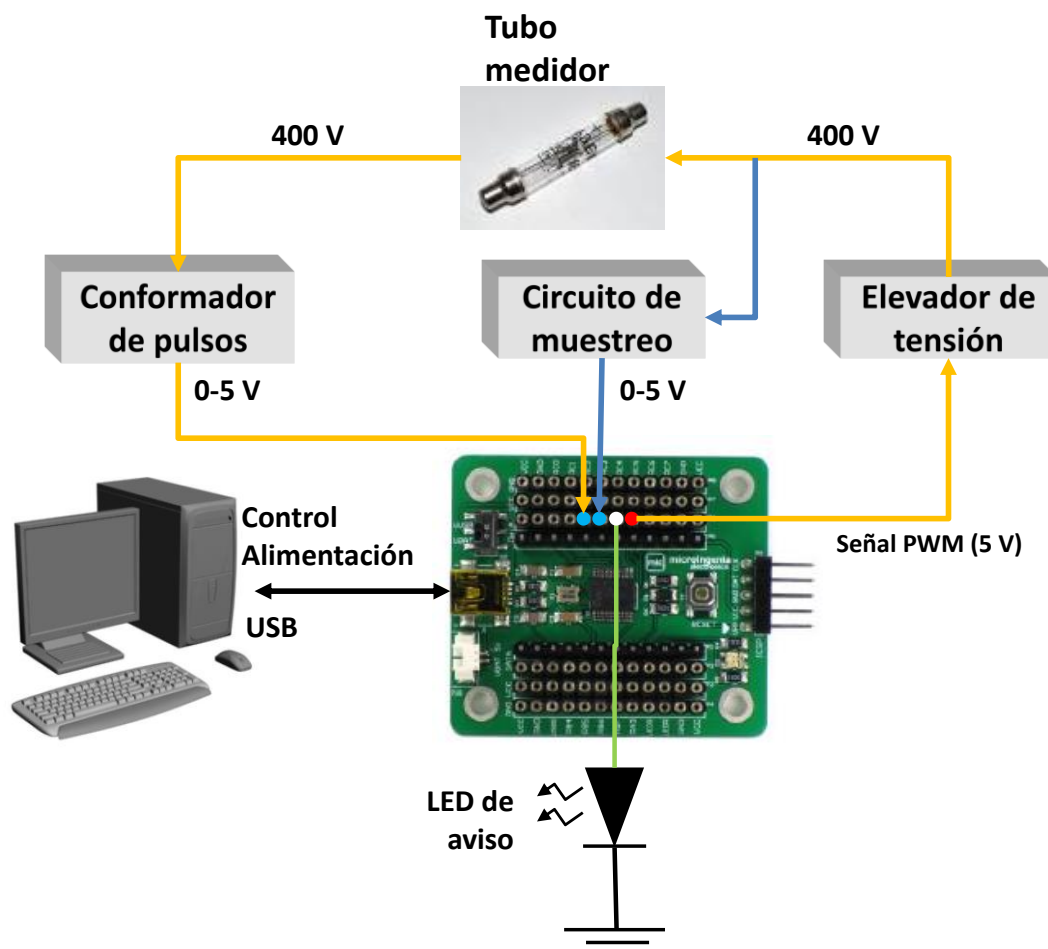


Fig. 4.16 - Conexiones placa de desarrollo para el software final

El programa comienza con una serie de configuraciones específicas para la placa entrenadora utilizada en el proyecto. Tras la configuración, inicializa las variables globales y las tablas de valores pertenecientes a cada una de las tablas de valores con las cuales se trabaja.

A continuación se procede a implementar la función 'main' que se ejecuta en el PIC. Dentro de ella, en primer lugar, se configura la comunicación USB, la interrupción producida por la llegada de un pulso (RC2) y los dos timers utilizados, uno asociado a la señal PWM (timer2) y el otro encargado de determinar si han transcurrido 100 milisegundos (timer0). En segundo lugar se ejecuta la parte del código encargada de la configuración de la señal PWM mediante teclado. En tercer y último lugar, se ejecuta el fragmento de código que administra las tablas, lo que ocurre cada 100 milisegundos (desbordamiento timer0).

En último lugar se establece la acción que se ejecuta cuando se produce la interrupción por desbordamiento del timer0 y por la aparición de un flanco de bajada en la patilla C2.

# Capítulo 5

## Elaboración del PCB

### 5.1 - Planteamiento del desarrollo

Toda la implementación física del sistema propuesto se ha llevado a cabo mediante el paquete Eagle v8.0.1 para el diseño de un prototipo hardware final. Así, se ha partido de un esquemático inicial (Figura 3.12) fruto de las pruebas realizadas en la etapa de desarrollo hardware y se ha confeccionado una placa de circuito impreso haciendo uso de la herramienta incluida en Eagle a tal efecto.

### 5.2 - Easily Applicable Graphical Layout Editor (EAGLE)

Dentro de los múltiples programas de diseños de PCB en el mercado, se ha optado por la utilización de EAGLE (Easily Applicable Graphical Layout Editor). Todo el desarrollo hardware que compone este proyecto, tanto esquemáticos como PCBs, ha sido realizado con este programa.

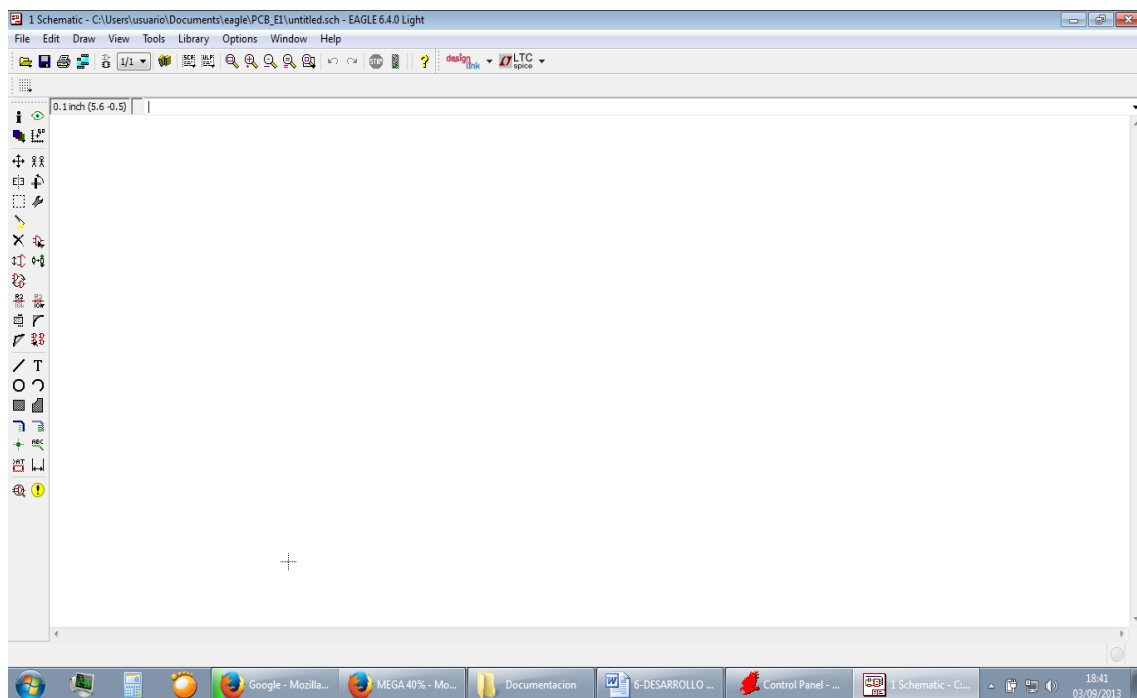


**Fig. 5.1 - Logotipo de Eagle®**

Este software de diseño PCB es capaz de diseñar un esquema electrónico y una placa de circuito impreso (PCB) que consta de un editor de esquemas, un editor de PCB y un módulo de trazador. El software viene con una amplia biblioteca de componentes, pero ésta puede

ampliarse, puesto que está incluido un editor para el diseño de nuevas piezas o para modificar las ya existentes. Está desarrollado por CadSoft ([www.cadsoftusa.com](http://www.cadsoftusa.com)), y está disponible en tres versiones, tanto para - 54 - windows como para Linux. En este proyecto se hace uso de la versión 'light', puesto que sus restricciones son suficientes para poder ejecutar este proyecto.

- Restricciones de la licencia Eagle Light Edition.
  - Área de diseño de placa limitada a 150 x 100 mm.
  - Sólo dos capas de diseño (top y bottom).
  - Tan sólo se puede crear un esquema por proyecto.
  - Soporte limitado vía e-mail o mediante la utilización de los foros.
  - Uso limitado a aplicaciones que no produzcan beneficios o para placas de evaluación.
  - Existe el permiso de distribución de EAGLE con distribuciones de Linux con colecciones de software o CD-ROMs, mediante información a la empresa desarrolladora.



**Fig. 5.2 - Entorno de desarrollo de Eagle**

## 5.3 - PCB, montaje y puesta en marcha

Se ha decidido realizar la elaboración del PCB mediante una placa fotosensible y un posterior revelado y ataque con ácido.

### 5.3.1 - Elaboración del layout del PCB

Tras haberse comprobado que el diseño de la placa y el programa son los adecuados, se ha procedido a diseñar el layout para construir la placa de circuito impreso.

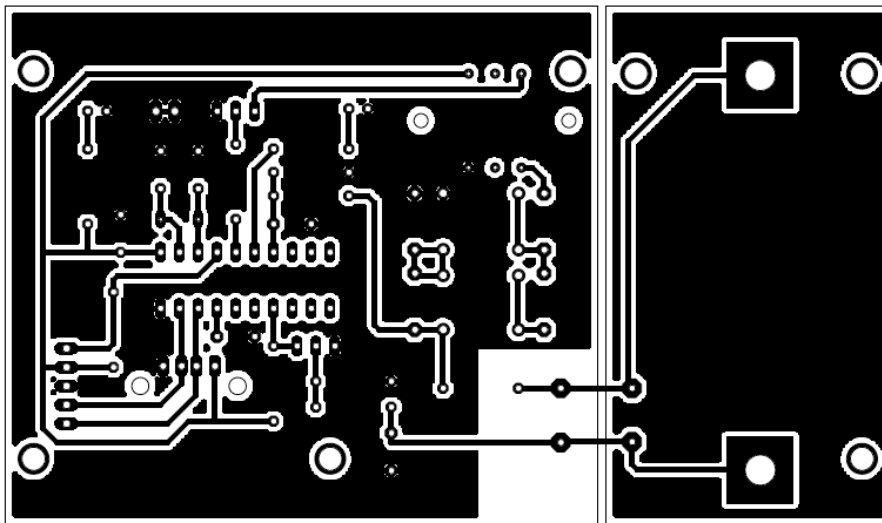


Fig. 5.3 - Layout del PCB

Como se puede observar en la figura 5.3, se han dispuesto todos los componentes buscando bastante similitud con el esquemático, a fin de facilitar la comprensión entre ambos.

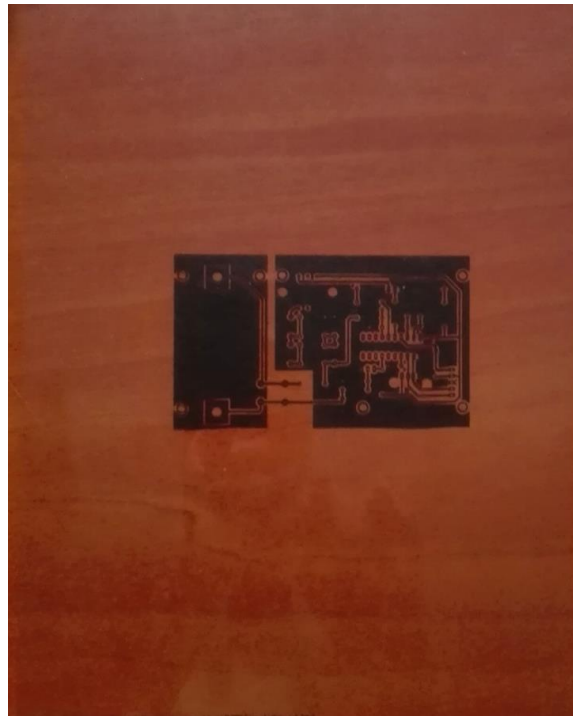
Se ha intentado utilizar el suficiente espacio como para que la distribución de los componentes sea compacta pero limpia. El ruteado de la placa se ha realizado a una sola cara, con pistas de 0.024 pulgadas de ancho.

### 5.3.2 - Elaboración del PCB por insolación y ataque químico

El proceso de fabricación de la placa de circuito impreso se ha basado en la fotolitografía y en el ataque por ácido, y consta de los siguientes pasos:

*Impresión del fotalito de la placa.* El diseño del PCB realizado anteriormente (Fig. 5.3) se ha impreso en un papel de transparencia (acetato). Se recomienda la inclusión de algún texto o referencia para evitar la impresión de manera incorrecta.





**Fig. 5.4 - Fotelito del PCB**

*Placa con resina fotosensible positiva* (Fig. 5.5). Se trata de una placa de material plástico cubierta de cobre por una sola de sus caras, y tratada con una resina fotosensible, protegida de la luz por un adhesivo opaco.



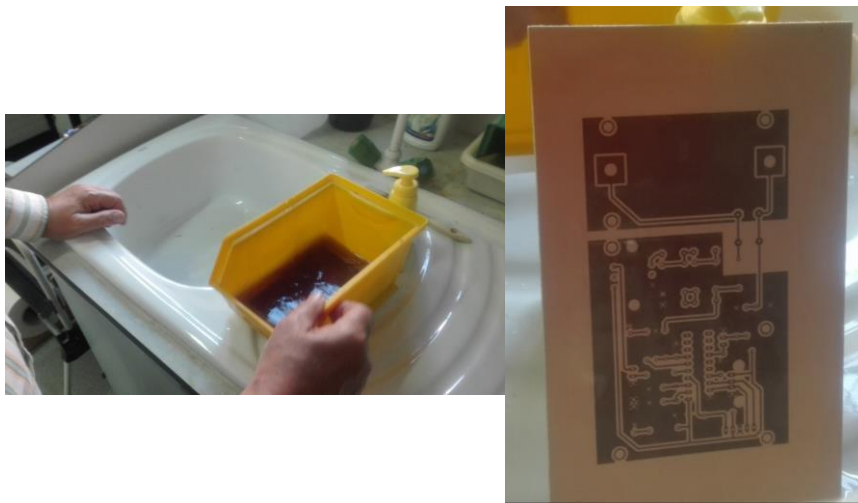
**Fig. 5.5 - Placa fotosensible**

*Insolación.* El fotolito y la placa se han introducido en la insoladora para exponer la zona que no se encuentra cubierta por la tinta a la radiación ultravioleta. El tiempo de exposición ha sido de 35 segundos. El fotolito debe mantenerse unido a la placa para evitar que se desplace durante la insolación, por ello la insoladora realiza el vacío durante la insolación.



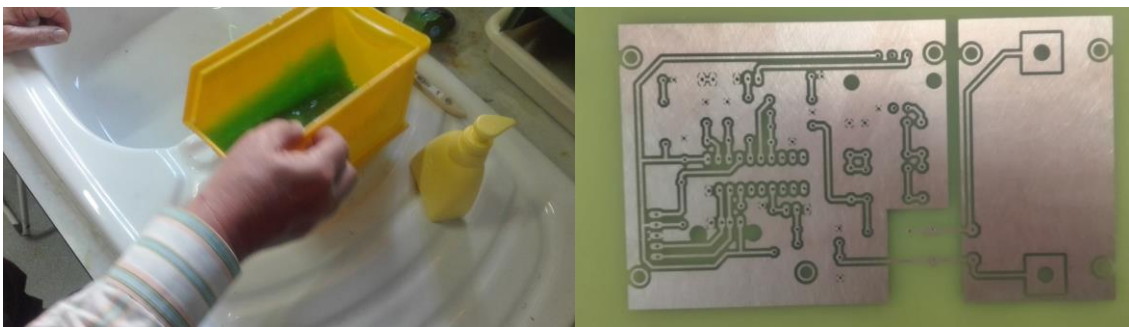
**Fig. 5.6 - Insoladora**

*Revelado.* La placa se ha introducido en un baño con revelador hasta que se ha apreciado que los diseños del fotolito se han transferido a la resina.



**Fig. 5.7 - Revelado del PCB**

*Ataque del cobre.* Se ha sumergido la placa en la disolución atacante hasta que el cobre no protegido por la resina se ha disuelto. La solución atacante está compuesta por dos partes de agua, una parte de agua oxigenada, y una parte de aguafuerte.



**Fig. 5.8 - Ataque del cobre del PCB**

Como último paso se ha procedido al taladrado de los pads de la tarjeta, con broca de 0,8 mm y 1 mm, y a la posterior soldadura con estaño de todos los elementos que componen el circuito.



**Fig. 5.9 - Placa final**

Una vez construida la tarjeta se han implementado dentro de ella los distintos programas de software del capítulo 4 para verificar que el funcionamiento de la misma es el adecuado. Se ha comprobado que la placa funciona correctamente.

Se ha dejado para futuros trabajos la utilización de la tarjeta como elemento de medición en un entorno en el que se disponga de una fuente radiactiva.

# Capítulo 6

## Conclusiones y trabajos futuros

### 6.1 - Conclusiones

En este proyecto se ha realizado la implementación software, el diseño y la construcción de una placa de circuito impreso que tiene como función la medición de radiación ionizante procedente de un elemento radiactivo.

Finalmente, destacar que con el trabajo realizado en este proyecto, aparte de suponer una aplicación real, se han adquirido conocimientos en el diseño electrónico, y más específicamente, en el diseño de contadores de radiación, en concreto, en los que la medida se realiza mediante el tubo Geiger-Muller.

### 6.1 - Trabajos futuros

Como ocurre siempre que se aborda un proyecto de este tipo, siempre es posible plantear líneas para la mejora y ampliación del trabajo elaborado. Entre éstas, cabe destacar las siguientes por su interés:

- La implementación de la solución en un chip ESP32 con capacidad para soportar comunicación WIFI para la transmisión de datos y el control de la placa, que ocasionaría la viabilidad de utilizar la placa medidora de manera portátil. Como factor en contra destacar que para poder adoptar esta alternativa se hace necesaria la inclusión de una batería de alimentación externa.

- Como ya se ha comentado al final del capítulo 5, se deja para posibles trabajos en el futuro la utilización *in situ* de la tarjeta como elemento de medida de una fuente radioactiva.
- Como ya se ha expuesto en el capítulo 3, se dispone de un circuito de muestreo de la señal de alta tensión, otorgando la posibilidad de que la placa funcione en lazo cerrado, regulando ella misma la tensión de alimentación del tubo. Esto no se ha abordado, dejándose su utilización y desarrollo para posibles trabajos posteriores.
- Asimismo, también sería interesante optimizar el código fuente usado, haciéndolo más fluido y simplificado.

Como comentario final, señalar que la placa del presente proyecto es una versión 1.0, en la cual el objetivo era obtener un soporte para comprobar el correcto funcionamiento del ejercicio programado y para poder ver el resultado de la programación. Como futuro trabajo se deja la creación de una versión 2.0 en la cual utilizando componentes SMD (*Surface Mount Device*) se verá reducido el tamaño de la placa.

---

## **Anexo A. Bibliografía y páginas web**

---

- [1] Compilador C CCS y Simulador Proteus para Microcontroladores PIC.  
Eduardo García Breijo  
ISBN-10: 8426714293 ISBN-13: 978-8426714299
- [2] Bioinstrumentación  
John G. Webster  
ISBN-10: 0471263273 ISBN-13: 978-0471263272
- [3] Manual de radiología para técnicos. Física, Biología y Protección Radiológica (5ªed.)  
Bushong, S
- [4] Física e instrumentación médicas (2ª ed.)  
Zaragoza, J.R.
- [5] Design and development of medical electronic instrumentation  
David Prutchi y Michael Norris  
ISBN 0-471-67623-3
- [6] Tutorial EAGLE versión 6
- [7] <https://sites.google.com/site/diygeigercounter/>
- [8] Wikipedia

---

## **Anexo B. Presupuesto**

---

En el presupuesto del proyecto, se ha incluido el precio de todos los materiales necesarios para implementarlo, y, además, se ha tenido en cuenta el coste y horas de la mano de obra necesarias para llevar a cabo cada parte que conforma el proyecto.

El presupuesto es estimado, pues aún no se sabe con certeza algunos rasgos importantes de este proyecto.

 **Costes de materiales de ejecución.**

Descripción	Cant.	PVP <sub>unitario</sub>	Total(€)
Placa fotosensible, simple cara 100x160	1	4,35	4,35
Cristal 48.0 MHZ	1	0,42	0,42
Resistencias	10	0,10	1,00
Condensador (22pF,15pF,33nF)	6	0,25	1,50
Condensador 10nF	4	0,26	1,04
Conector USB	1	0,90	0,90
LEDs	2	0,15	0,30
Conectores macho 20	1	0,70	0,70
PIC 18F14K50	1	2,71	2,71
Diodo 4007	4	0,15	0,60
Transistor BC548	2	0,30	0,60
Transformador LT700	1	4,65	4,65
Tubo Geiger SI-3BG	1	7,40	7,40
		<b>Materiales ejecución</b>	<b>26,17</b>
			<b>Unidad</b>

 **Costes de mano de obra**

Descripción	€/hora	Horas	Total(€)
INGENIERO DISEÑO HARDWARE	15,00	65	975,00
INGENIERO DISEÑO FIRMWARE	15,00	35	525,00
INGENIERO DISEÑO SOFTWARE	15,00	10	150,00
		<b>Mano de obra</b>	<b>1650,00</b>

 **Costes de material de desarrollo**

Descripción	Cant.	PVP <sub>unitario</sub>	Total(€)
Estación de trabajo informatico	1	2000,00	2000,00
Kit programación µC	1	600,00	600,00
Kit de soldador	1	100,00	100,00
Licencia PICC	1	600,00	600,00
Revelado y ataque de placa	1	60,00	60,00
		<b>Materiales desarrollo</b>	<b>3360,00</b>

 **Costes totales**

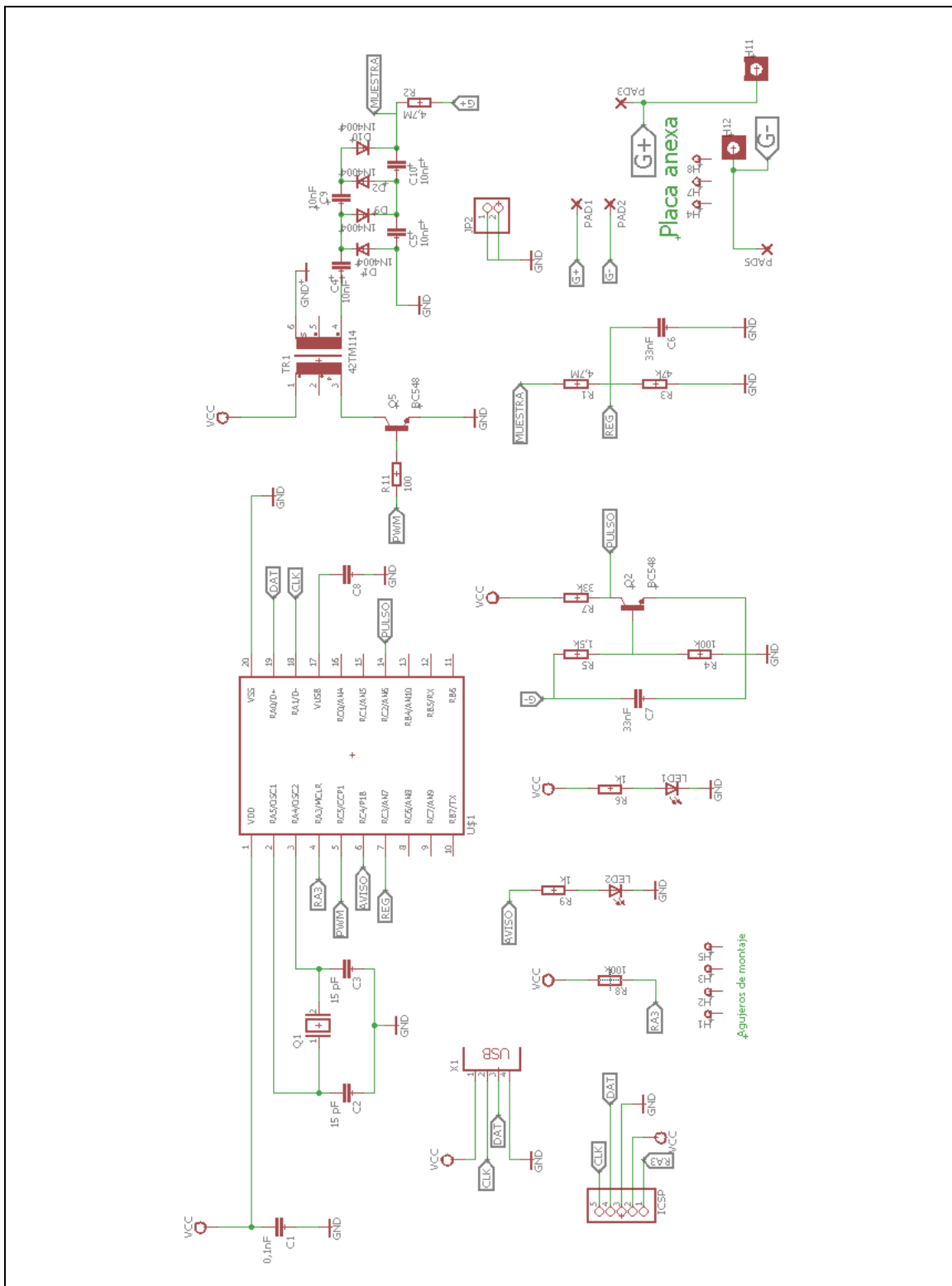
Descripción	Total (€)
MATERIALES DE EJECUCIÓN	26,17
MANO DE OBRA	1650,00
MATERIALES DE DESARROLLO	3360,00
	<b>5036,17</b>


 **Supuesto de fabricación**

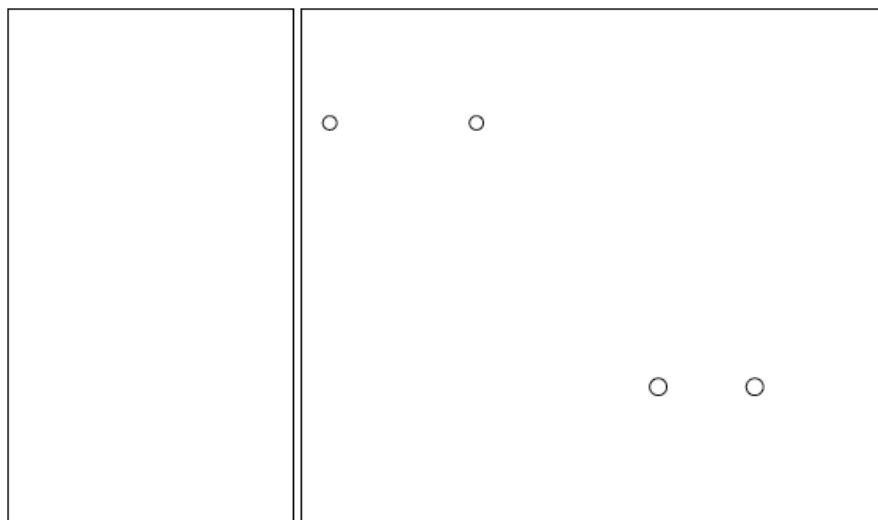
PCB Prototipo	Precio
Materiales de ejecución	26,17 €/unidad
Total 1 unidad prototipo	$26,17 \times 0,9 = 23,56 \text{ €}$
Total 100 unidades prototipo	2.356,00 €
Beneficio industrial (20%)	471,20 € - 261,70€
	2,11 €/unidad
PVP (IVA 21%)	<b>2.850,76 €</b> <b>28,51 €/unidad</b>




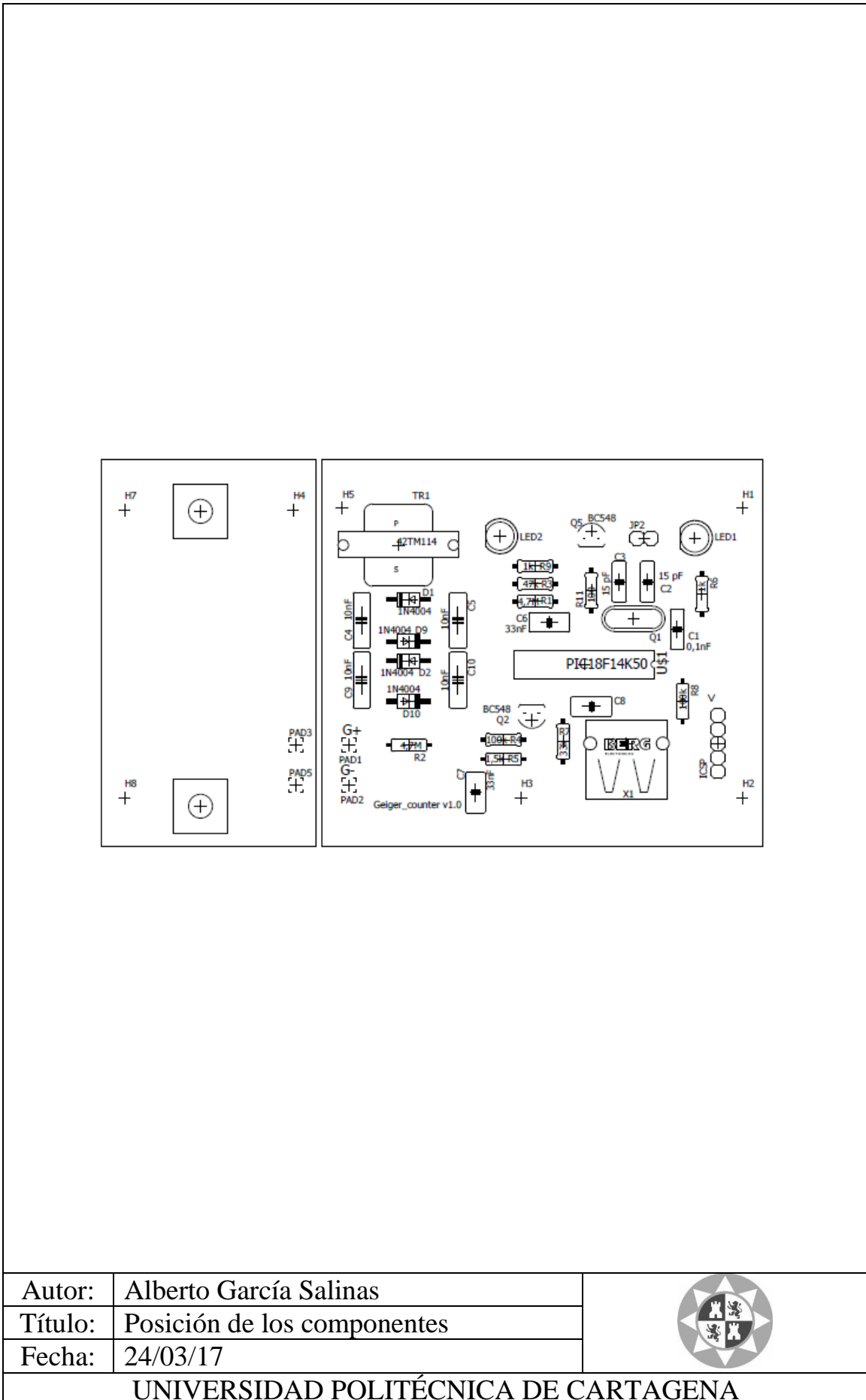
# Anexo C. Planos



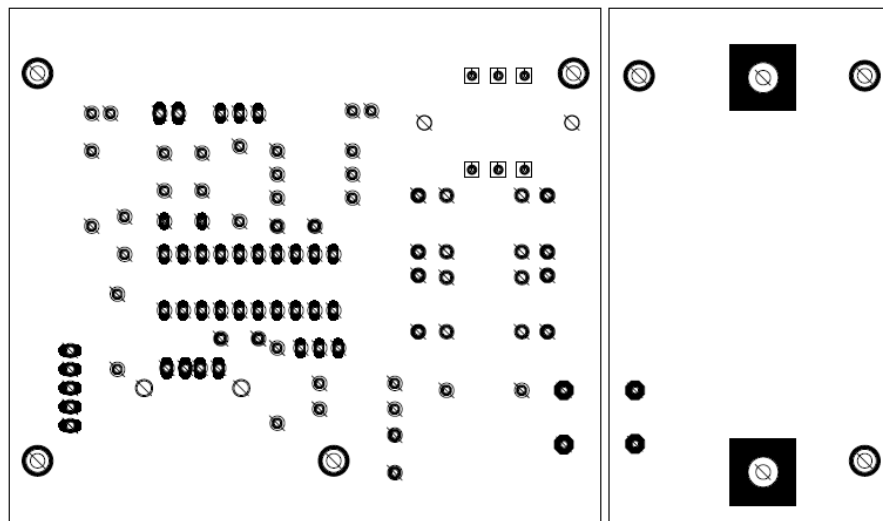
Autor:	Alberto García Salinas	
Título:	Esquemático	
Fecha:	24/03/17	
<b>UNIVERSIDAD POLITÉCNICA DE CARTAGENA</b>		




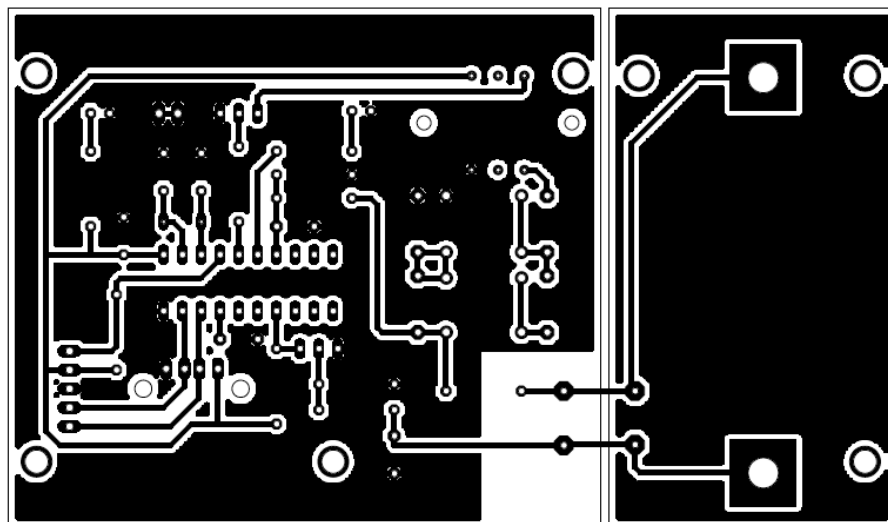
Autor:	Alberto García Salinas	
Título:	Gerber – Perímetro	
Fecha:	24/03/17	
UNIVERSIDAD POLITÉCNICA DE CARTAGENA		




Autor:	Alberto García Salinas	
Título:	Posición de los componentes	
Fecha:	24/03/17	
UNIVERSIDAD POLITÉCNICA DE CARTAGENA		



Autor:	Alberto García Salinas	
Título:	Gerber – Pads y taladros	
Fecha:	24/03/17	
UNIVERSIDAD POLITÉCNICA DE CARTAGENA		



Autor:	Alberto García Salinas	
Título:	Layout PCB	
Fecha:	24/03/17	
UNIVERSIDAD POLITÉCNICA DE CARTAGENA		



```

//Configuracion de la señal PWM
setup_ccp1(CCP_PWM);
setup_timer_2(T2_DIV_BY_1,0,1);
pwm_on();

while(TRUE)
{
//Si se - 70 -ódigo caracteres y se guardan en array in_buffer
//Si el formato de la - 70 -ódigo- 70 -n- 70 -ión es correcto
//Se actualiza la señal PWM en frecuencia y - 70 -ódigo- 70 -n de
trabajo
if(usb_cdc_kbhit())
{
in_buffer[0]=in_buffer[1];
in_buffer[1]=in_buffer[2];
in_buffer[2]=in_buffer[3];
in_buffer[3]=usb_cdc_getc();
if(in_buffer[0]==250&&in_buffer[3]==250)
{
//Si ha cambiado el valor de la frec se actualiza
if(- 70 -ódigo!=in_buffer[1] && in_buffer[1]==239)
{
- 70 -ódigo= in_buffer[1];
setup_timer_2(T2_DIV_BY_1,- 70 -ódigo,1);
}
//Si ha cambiado el valor del pwm se actualiza
if(pwm!=in_buffer[2] && in_buffer[2]<95)
{
pwm=in_buffer[2];
set_pwm1_duty(pwm);
}
}
}
//Cada 100ms se ejecuta esta parte del - 70 -ódigo
if (a==1)
{
//Se elimina el valor antiguo de c100ms del array
//Se sustituye por el nuevo valor de c100ms del array
//Se actualizan variables
PS=PS-CMS[j];
CMS[j]=c100ms;
PS=PS+CMS[j];
//Se evalua si se enciende LED rojo
if(c100ms>100)output_high(PIN_C4);
else output_low(PIN_C4);
//Código para enviar variables
printf(usb_cdc_putc,"%Lu %Lu %Lu \r\n",c100ms,PS,PM);
c100ms=0;
a=0;
j++;
//Cada 10 veces 100ms se ejecuta esta parte del - 70 -ódigo
//Se elimina el valor antiguo de cuentas por segundo del array
//Se sustituye por el nuevo valor de cuenta por segundo del array
//Se actualizan variables
if (j==10)
{
j=0;
PM=PM-CPS[i];
CPS[i]=PS;
}
}
}

```

```

        PM=PM+CPS[i];
        i++;
        if (i==60) i=0;
    }
}
}

#int_timer0
void isr_timer0(void)
{
    a=1;
    set_timer0(56161);
}

#INT_EXT2
void interrupt_RC2()//llega pulso a patilla C2
{
    c100ms=c100ms+1;
}

```

## Anexo F. Configuración PWM

			Parámetros T_2							1= VALOR INCORRECTO 0= VALOR CORRECTO	
F_RELOJ	F_trabajo	T_trabajo	T2_DIV	Post	F_pwm(Hz)	Pwm(%)	Period	Duty			
4,80E+07	1,20E+07	8,33E-08	1	1	47000	0,05	254,32	12,77	0	0	
4,80E+07	1,20E+07	8,33E-08				0,2	254,32	51,06	0	0	
4,80E+07	1,20E+07	8,33E-08				0,4	254,32	102,13	0	0	
4,80E+07	1,20E+07	8,33E-08				0,6	254,32	153,19	0	0	
4,80E+07	1,20E+07	8,33E-08				0,75	254,32	191,49	0	0	
4,80E+07	1,20E+07	8,33E-08				0,9	254,32	229,79	0	0	
4,80E+07	1,20E+07	8,33E-08	1	1	50000	0,05	239,00	12,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,2	239,00	48,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,4	239,00	96,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,6	239,00	144,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,75	239,00	180,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,9	239,00	216,00	0	0	
4,80E+07	1,20E+07	8,33E-08	1	1	75000	0,05	159,00	8,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,2	159,00	32,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,4	159,00	64,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,6	159,00	96,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,75	159,00	120,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,9	159,00	144,00	0	0	
4,80E+07	1,20E+07	8,33E-08	1	1	100000	0,05	119,00	6,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,2	119,00	24,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,4	119,00	48,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,6	119,00	72,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,75	119,00	90,00	0	0	
4,80E+07	1,20E+07	8,33E-08				0,9	119,00	108,00	0	0	

Formula para el period

$$\text{period}=(F\_trabajo)/(F\_PWM*T2\_DIV*POST))-1$$

Fórmula para el duty

$$\text{duty}=(PWM*F\_trabajo)/(F\_pwm*T2\_DIV*Post)$$



## Anexo G. Tabla ASCII

### El código ASCII

sigla en inglés de American Standard Code for Information Interchange (Código Estadounidense Estándar para el Intercambio de Información)

[www.elcodigoascii.com.ar](http://www.elcodigoascii.com.ar)

Caracteres de control ASCII				Caracteres ASCII imprimibles								ASCII extendido												
DEC	HEX	Símbolo ASCII		DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo						
00	00h	NULL	(carácter nulo)	32	20h	espacio	64	40h	@	96	60h	.	128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó
01	01h	SOH	(inicio encabezado)	33	21h	!	65	41h	A	97	61h	a	129	81h	ú	161	A1h	í	193	C1h	ł	225	E1h	ó
02	02h	STX	(inicio texto)	34	22h	"	66	42h	B	98	62h	b	130	82h	ë	162	A2h	î	194	C2h	ł	226	E2h	ô
03	03h	ETX	(fin de texto)	35	23h	#	67	43h	C	99	63h	c	131	83h	â	163	A3h	û	195	C3h	ł	227	E3h	ö
04	04h	EOT	(fin transmisión)	36	24h	\$	68	44h	D	100	64h	d	132	84h	ä	164	A4h	ü	196	C4h	ł	228	E4h	ø
05	05h	ENQ	(enquiry)	37	25h	%	69	45h	E	101	65h	e	133	85h	å	165	A5h	ñ	197	C5h	ł	229	E5h	õ
06	06h	ACK	(acknowledgement)	38	26h	&	70	46h	F	102	66h	f	134	86h	ä	166	A6h	*	198	C6h	ł	230	E6h	µ
07	07h	BEL	(timbre)	39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	°	199	C7h	ł	231	E7h	þ
08	08h	BS	(retroceso)	40	28h	(	72	48h	H	104	68h	h	136	88h	è	168	A8h	ˆ	200	C8h	ł	232	E8h	ÿ
09	09h	HT	(tab horizontal)	41	29h	)	73	49h	I	105	69h	i	137	89h	é	169	A9h	˜	201	C9h	ł	233	E9h	ÿ
10	0Ah	LF	(salto de línea)	42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	ê	170	AAh	˘	202	CAh	ł	234	EAh	ÿ
11	0Bh	VT	(tab vertical)	43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ë	171	ABh	½	203	CBh	ł	235	EBh	ÿ
12	0Ch	FF	(form feed)	44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	ì	172	ACH	¾	204	Ch	ł	236	ECh	ÿ
13	0Dh	CR	(retorno de carro)	45	2Dh	-	77	4Dh	M	109	6Dh	m	141	8Dh	í	173	ADh	⅔	205	CDh	ł	237	EDh	ÿ
14	0Eh	SO	(shift Out)	46	2Eh	.	78	4Eh	N	110	6Eh	n	142	8Eh	î	174	AEh	¼	206	CEh	ł	238	EEh	ÿ
15	0Fh	SI	(shift In)	47	2Fh	/	79	4Fh	O	111	6Fh	o	143	8Fh	ï	175	AFh	½	207	CFh	ł	239	EFh	ÿ
16	10h	DLE	(data link escape)	48	30h	0	80	50h	P	112	70h	p	144	90h	ê	176	B0h	¾	208	D0h	ł	240	F0h	ÿ
17	11h	DC1	(device control 1)	49	31h	1	81	51h	Q	113	71h	q	145	91h	ë	177	B1h	⅔	209	D1h	ł	241	F1h	±
18	12h	DC2	(device control 2)	50	32h	2	82	52h	R	114	72h	r	146	92h		178	B2h	¼	210	D2h	ł	242	F2h	
19	13h	DC3	(device control 3)	51	33h	3	83	53h	S	115	73h	s	147	93h		179	B3h	½	211	D3h	ł	243	F3h	
20	14h	DC4	(device control 4)	52	34h	4	84	54h	T	116	74h	t	148	94h		180	B4h	¾	212	D4h	ł	244	F4h	
21	15h	NAK	(negative acknowledge)	53	35h	5	85	55h	U	117	75h	u	149	95h		181	B5h	⅔	213	D5h	ł	245	F5h	
22	16h	SYN	(synchronous idle)	54	36h	6	86	56h	V	118	76h	v	150	96h		182	B6h	¼	214	D6h	ł	246	F6h	
23	17h	ETB	(end of trans. block)	55	37h	7	87	57h	W	119	77h	w	151	97h		183	B7h	½	215	D7h	ł	247	F7h	
24	18h	CAN	(cancel)	56	38h	8	88	58h	X	120	78h	x	152	98h		184	B8h	¾	216	D8h	ł	248	F8h	
25	19h	EM	(end of medium)	57	39h	9	89	59h	Y	121	79h	y	153	99h		185	B9h	⅔	217	D9h	ł	249	F9h	
26	1Ah	SUB	(substitute)	58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah		186	BAh	¼	218	DAh	ł	250	FAh	
27	1Bh	ESC	(escape)	59	3Bh	;	91	5Bh	[	123	7Bh	{	155	9Bh		187	BBh	½	219	DBh	ł	251	FBh	
28	1Ch	FS	(file separator)	60	3Ch	<	92	5Ch	\	124	7Ch		156	9Ch		188	BCh	¾	220	DBh	ł	252	FBh	
29	1Dh	GS	(group separator)	61	3Dh	=	93	5Dh	]	125	7Dh	}	157	9Dh		189	BDh	⅔	221	DDh	ł	253	FDh	
30	1Eh	RS	(record separator)	62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh		190	BEh	¼	222	DEh	ł	254	FEh	
31	1Fh	US	(unit separator)	63	3Fh	?	95	5Fh	_				159	9Fh		191	BFh	½	223	DFh	ł	255	FFh	
127	20h	DEL	(delete)																					

## Anexo H. Índice de figuras

Imagen	Descripción
Fig. 1.1	Diferentes tubos Geiger-Muller
Fig. 2.1	Tipos de radiación
Fig. 2.2	Relación energética de las radiaciones
Fig. 2.3	Factores de ponderación
Fig. 2.4	Esquema de efectos biológicos
Fig. 2.5	Dosis umbral para diferentes efectos y probabilidad de aparición de cáncer según la dosis
Fig. 2.6	Comportamiento del tubo frente a la tensión de entrada
Fig. 2.7	Esquema y funcionamiento de un tubo Geiger-Muller
Fig. 2.8	Número de pulsos vs dosis medida
Fig. 2.9	Mediciones de ciclos por segundo frente a determinadas radiaciones
Fig. 2.10	Primer circuito generador de tensión
Fig. 2.11	Segundo circuito generador de tensión
Fig. 2.12	Tercer circuito generador de tensión
Fig. 2.13	Primer circuito de adquisición
Fig. 2.14	Segundo circuito de adquisición
Fig. 2.15	Tercer circuito de adquisición
Fig. 3.1	Diagrama hardware
Fig. 3.2	Tubos SI-3BG
Fig. 3.3	Datasheet tubo SI-3BG
Fig. 3.4	Esquemático de la generación de alta tensión
Fig. 3.5	Comportamiento generador de tensión para PWM de 50KHz
Fig. 3.6	Circuito inversor CC/CA

Fig. 3.7	Vpp en función de la frecuencia
Fig. 3.8	A) Cuadruplicador de tensión B) Señal obtenida
Fig. 3.9	Esquemático del circuito de adquisición de pulsos
Fig. 3.10	Señal obtenida al producirse un pulso
Fig. 3.11	Microcontrolador 18F14K50
Fig. 3.12	Conectores ICSP
Fig. 3.13	Esquemático de la solución propuesta
Fig. 4.1	Entorno de desarrollo
Fig. 4.2	Logotipo CCS
Fig. 4.3	Entorno de desarrollo CCS
Fig. 4.4	Placa de desarrollo 18F2550 Trainerv1.0
Fig. 4.5	Conexiones placa de desarrollo 4.2.1
Fig. 4.6	Conexiones placa de desarrollo 4.2.2
Fig. 4.7	Envío de caracteres por USB
Fig. 4.8	Logotipo de Realterm
Fig. 4.9	Conexiones placa de desarrollo 4.2.3
Fig. 4.10	Captura de la señal PWM
Fig. 4.11	Conexiones placa de desarrollo 4.2.4
Fig. 4.12	Muestra del archivo de texto generado
Fig. 4.13	C100ms, CPS y CPM
Fig. 4.14	Conexiones placa de desarrollo 4.2.5
Fig. 4.15	Control de la señal de PWM
Fig. 4.16	Conexiones placa de desarrollo para el software final
Fig. 5.1	Logotipo de EAGLE
Fig. 5.2	Entorno de desarrollo EAGLE
Fig. 5.3	Layout del PCB
Fig. 5.4	Fotolito del PCB
Fig. 5.5	Placa fotosensible
Fig. 5.6	Insoladora
Fig. 5.7	Revelado del PCB
Fig. 5.8	Ataque del cobre del PCB
Fig. 5.9	Placa final