

UNIVERSIDAD DE CASTILLA LA MANCHA

ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS INDUSTRIALES

CIUDAD REAL

PROYECTO FIN DE CARRERA N° 13-12-200453

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL  
REMOTO INDEPENDIENTE PARA UN LÁSER DE USO  
CIENTÍFICO

AUTOR:

José Antonio Masegosa Zapata.

DIRECTOR DEL PROYECTO:

Gonzalo Rodríguez Prieto.

Febrero 2013.





---

*A mis padres León y M<sup>a</sup> Ángeles, por su apoyo incondicional.*

*A mis hermanos, tía y abuelos.*

*A la cúpula y a todas las personas que me han ayudado y han confiado en mi...*

*...muchas gracias a todos.*

# Índice general

<b>1. Introducción</b>	<b>9</b>
<b>2. Plataforma física</b>	<b>11</b>
2.1. Descripción del controlador . . . . .	12
2.2. Componentes . . . . .	15
2.2.1. Arduino Uno . . . . .	15
2.2.2. Potenciómetro digital DS1804 . . . . .	17
2.2.3. Potenciómetro rotatorio . . . . .	19
2.2.4. Pantalla LCD . . . . .	19
2.2.5. Tarjeta PCB . . . . .	20
2.3. Descripción de los circuitos . . . . .	21
2.3.1. Variador de tensión . . . . .	21
2.3.2. Pantalla LCD . . . . .	22
2.3.3. Lectores del láser . . . . .	22
2.3.4. Indicadores . . . . .	23
2.3.5. Tarjeta PCB . . . . .	24
2.4. Desarrollo de la tarjeta PCB . . . . .	26
2.4.1. Esquema del circuito . . . . .	26
2.4.2. Diseño del circuito impreso . . . . .	28
2.4.3. Fabricación de la PCB . . . . .	30
<b>3. Software</b>	<b>31</b>
3.1. Programación de la tarjeta Arduino Uno . . . . .	31
3.2. Diseño e implementación . . . . .	32
3.2.1. Declaración de variables . . . . .	32
3.2.2. Bloque setup . . . . .	34
3.2.3. Bloque loop . . . . .	35
3.2.4. Diagramas de bloque . . . . .	37
3.3. Software empleado para la calibración del controlador. . . . .	39



---

<b>4. Evolución del dispositivo</b>	<b>41</b>
4.1. Evolución del hardware . . . . .	41
4.2. Evolución del software . . . . .	44
<b>5. Datos experimentales</b>	<b>47</b>
5.1. Calibración de la tensión . . . . .	47
5.2. Medición de la intensidad lumínica del láser . . . . .	50
<b>6. Conclusiones</b>	<b>55</b>
<b>A. Potenciómetro digital DS1804</b>	<b>59</b>
<b>B. Transistor BC546</b>	<b>69</b>
<b>C. Pantalla digital. LCD</b>	<b>75</b>
<b>D. Arduino Uno</b>	<b>99</b>
<b>E. Lenguaje Arduino</b>	<b>109</b>
<b>F. Software principal desarrollado</b>	<b>111</b>
<b>G. Software para relación posición-tensión</b>	<b>121</b>
<b>H. Software inicial</b>	<b>125</b>
H.1. Lectura de potenciómetro rotatorio y conversión al digital. . . . .	125
H.2. Pontenciómetros y pantalla LCD . . . . .	127
<b>I. Tablas de intensidad lumínica</b>	<b>135</b>

# Índice de figuras

2.1.	Lateral del dispositivo con los conectores . . . . .	11
2.2.	Conector RS232 . . . . .	12
2.3.	Parte superior del prototipo . . . . .	14
2.4.	Componentes . . . . .	15
2.5.	Arduino Uno . . . . .	16
2.6.	DS1804 . . . . .	17
2.7.	Funcionamiento del puerto de tres terminales . . . . .	18
2.8.	Diagrama de bloques del DS1804 . . . . .	18
2.9.	LCD empleada en este proyecto . . . . .	19
2.10.	Renderización digital de la placa PCB . . . . .	20
2.11.	Esquema de conexiones lógicas del variador de tensión . . . . .	21
2.12.	Circuito de pantalla LCD . . . . .	22
2.13.	Circuito del transistor para lectura de corriente y pontecia . . . . .	23
2.14.	Circuito indicador . . . . .	24
2.15.	PCB . . . . .	25
2.16.	Boceto inicial del circuito de la PCB . . . . .	27
2.17.	PCB inicial . . . . .	28
2.18.	Circuito impreso, forma final . . . . .	29
2.19.	PCB tridimensional . . . . .	30
3.1.	Entorno de desarrollo integrado basado en java . . . . .	32
3.2.	Bloques principales del software . . . . .	37
3.3.	Diagrama del bucle <i>loop</i> . . . . .	38
4.1.	Primer prototipo del dispositivo de control . . . . .	42
4.2.	Prototipo con la LCD añadida . . . . .	43
4.3.	Dispositivo final montado y listo para su uso . . . . .	44
5.1.	Circuito para la calibración de la posición del potenciómetro digital según el voltaje de salida . . . . .	47
5.2.	Montaje real . . . . .	48
5.3.	Voltaje de salida respecto a la posición del potenciómetro digital . . . . .	50



## ÍNDICE DE FIGURAS



---

5.4. Sistema para la medida de la intensidad . . . . .	51
5.5. Intensidad lumínica del láser frente a voltaje del controlador . . . . .	52
5.6. Intensidad lumínica del láser respecto al voltaje de salida del controlador . .	53

# Índice de tablas

5.1.	Posición en el potenciómetro y voltaje de salida. . . . .	49
5.2.	Ecuaciones aproximadas; lineal y exponencial . . . . .	50
5.3.	Tensión frente a intensidad luminosa de las dos medidas . . . . .	52
5.4.	Ecuaciones aproximadas de la intensidad lumínica; lineal y exponencial . .	53
I.1.	Tensión frente a intensidad luminosa. Medida1. . . . .	136
I.2.	Tensión frente a intensidad luminosa. Medida2. . . . .	137



## ÍNDICE DE TABLAS

---



# Capítulo 1

## Introducción

El objetivo final de este proyecto es el diseño e implementación de un sistema de control de intensidad de un láser Excelsior 532-150, de 532 nm de longitud de onda y 150 mW de potencia máxima.

Actualmente, en el Instituto de Investigaciones Energéticas y Aplicaciones Industriales (INEI) se lleva a cabo un experimento científico que intenta estudiar las propiedades de una explosión eléctrica provocada por el paso de una gran cantidad de corriente por un conductor muy fino, fenómeno conocido desde hace tiempo [3] pero que sigue ofreciendo preguntas en nuestros días [4].

Para poder llevar a cabo los estudios, uno de los métodos más empleados es observar la imagen del plasma generado mediante cámaras electrónicas especiales para investigación. Como este plasma se mueve a muy alta velocidad, necesita fuentes de iluminación especiales, como las luces láser de las que dispone el laboratorio. Pero la potencia del láser debe ser controlada para no saturar la imagen, lo que precisa de un sistema de control por tensión externa al láser.

En el objetivo de este proyecto se demuestra la realización de un sistema físico y del software de control de este láser para poder variar su tensión de control, de tal forma que se pueda modificar su potencia de emisión. Además de cumplir con este objetivo, considerado principal, también se cubren otra serie de funciones de control y monitorización.

Para programar la electrónica del sistema de control, basada en la placa ARDUINO [8] se ha empleado el entorno de programación Processing, lo que ha obligado a dedicar una parte de tiempo del proyecto a su aprendizaje.

La comunicación entre ordenador e instrumento se ha realizado mediante un cable USB, conectado a la placa Arduino Uno, una de las múltiples versiones de las placas de control ARDUINO.

Esta memoria se estructura de la siguiente manera:

- En este primer capítulo se presenta una introducción al proyecto y el por qué del mismo.



- 
- En el segundo capítulo se realiza una descripción del sistema físico diseñado, así como de los distintos componentes utilizados. También se describen los circuitos desarrollados con el objetivo de poder variar la potencia de emisión del láser y de controlar otras funciones secundarias del mismo.
  - El lenguaje de programación utilizado, así como los programas de control del dispositivo, se tratan en el tercer capítulo.
  - El cuarto capítulo es un resumen de la evolución del desarrollo del dispositivo, tanto a nivel de hardware como de software.
  - En el quinto capítulo se recogen los datos obtenidos de las distintas mediciones, así como la interpretación de estos datos y una corta descripción del dispositivo empleado.
  - Al final del proyecto se presentan las conclusiones del proyecto y sobre el desarrollo del mismo, así como algunas posibilidades del trabajo futuro a realizar, para terminar el proyecto con los apéndices.

# Capítulo 2

## Plataforma física

En este capítulo se presenta una visión general del sistema electrónico desarrollado, describiendo sus componentes y funcionamiento. En esta sección del capítulo se describe el sistema de control y el láser asociado, para en la sección siguiente dar una descripción general del controlador. El capítulo finaliza con una sección dedicada al diseño y construcción de la placa electrónica necesaria en el dispositivo de control.

El dispositivo de control desarrollado tiene unas dimensiones exteriores de 175 x 45,5 x 126,2 mm. El volumen es más grande del necesario para los componentes actuales, pero esto permitirá futuros desarrollos como los mencionados en el capítulo 6.

En uno de los laterales del dispositivo se encuentran dos conectores que pueden servir como fuente de alimentación. Uno de estos conectores es un puerto USB (Ver fig. 2.1) que además de servir como fuente de alimentación, se usa para conectar el dispositivo a un ordenador. Así, si el usuario lo desea, puede realizar cambios en el software.



Figura 2.1: Lateral del dispositivo con los conectores. 1. Conector USB y posible alimentación, 2. Fuente de alimentación externa.

Del lateral opuesto sale el cable que conecta el dispositivo al láser siguiendo una conexión RS232 macho.

En la cara superior del dispositivo, se encuentran los distintos dispositivos luminosos,



## 2.1. DESCRIPCIÓN DEL CONTROLADOR



el botón de reinicio, la pantalla LCD y el potenciómetro rotatorio, todos ellos descritos en la sección 2.2.

El dispositivo controla un láser de 532 nm de longitud de onda, compuesto por una cabeza láser y una unidad de potencia. Está configurado para trabajar siempre a plena potencia, sin embargo, la unidad encargada de suministrar potencia puede configurarse mediante dispositivos externos de forma que la potencia de salida de la cabeza láser se modifique.

Para ello, el dispositivo de control envía una señal de tensión constante de amplitud arbitraria, de forma que la intensidad final del láser depende de esta señal de control, como se describe en capítulos posteriores.

Además de controlar la intensidad, se dispone de una serie de testigos luminosos que indican el proceso de modificación de la tensión, así como el momento en el cual el láser está preparado para ser utilizado.

La conexión entre la unidad de potencia del láser y el dispositivo de control se realiza con la conexión RS232 ya mencionada con un cable de 1,15 metros.

La tensión de control varía en función de la resistencia señalada por el usuario en el potenciómetro rotatorio. El esquema lógico que relaciona estos componentes se explicará al final de este capítulo, en la sección 2.4.

### 2.1. Descripción del controlador

La conexión entre el láser y el controlador se realiza a través de un cable RS232 de 15 pines, figura 2.2, de forma que cada uno de estos pines se encargará de recibir o transmitir la información necesaria entre el láser y el dispositivo según las especificaciones del fabricante del láser, de forma que en este proyecto se utilizan 9 de estos pines, de los cuales 2 están instalados pero no se ha desarrollado ninguna programación para sus señales.



Figura 2.2: Conector RS232. [<http://www.preciolandia.com/co/usb-2-0-to-rs232-serial-db9-cable-db25-a-6n4dsm-a.html>].

A continuación se detallan los pines utilizados y sus funciones:

**PIN1:** Salida. Indica cuando la cabeza láser ha alcanzado la potencia necesaria y está dentro del rango de temperatura de funcionamiento adecuada.



## 2.1. DESCRIPCIÓN DEL CONTROLADOR



**PIN3:** Salida. Si la temperatura de la cabeza láser es la adecuada para trabajar, está activo.

**PIN4:** Salida. Suministra una señal proporcional a la corriente de la salida del láser.  
(Escala:  $100 \text{ mV/Amp}$  según el fabricante)<sup>1</sup>

**PIN5:** Salida. Presenta una señal proporcional a la potencia de la salida del láser.  
(Escala:  $0.666 \text{ mV/mW}$  según el fabricante)<sup>1</sup>

**PIN6:** Tierra.

**PIN7:** Tierra.

**PIN8:** Entrada. Suministra a la unidad de potencia la tensión necesaria para poder utilizar el láser a una potencia arbitraria.

**PIN11:** Tierra.

**PIN12:** Tierra.

La parte superior del dispositivo de control muestra el potenciómetro y los LEDs de salida, figura 2.3.

---

<sup>1</sup>Desarrollado físicamente pero no a nivel de software



Figura 2.3: Parte superior del prototipo. 1. Entrada para el botón reset, 2. Pantalla LCD, 3. Potenciómetro rotatorio, 4. Indicadores LEDs.

Cada uno de los indicadores luminosos que se pueden observar en la figura 2.3, proporcionan información acerca del funcionamiento tanto del láser como del dispositivo de control objeto de este proyecto según el siguiente patrón:

**Led verde:** Conectado al pin  $U/\bar{D}$  del potenciómetro digital DS1804. Si está encendido, indica que el potenciómetro digital incrementa la resistencia. Por el contrario, si se encuentra apagado, la resistencia disminuye. En el prototipo está señalado como  $U/\bar{D}$ .

**Led amarillo:** Conectado al pin  $\overline{INC}$  del potenciómetro digital DS1804. Si se encuentra parpadeando, indica que el potenciómetro digital se encuentra en el proceso de cambiar la resistencia ofrecida. Cada parpadeo de este led indica un cambio de una unidad de 99 en los posibles valores de la resistencia. El LED señalado como  $\overline{INC}$  en el frontal.

**Led blanco:** Conectado al pin 1. Cuando la tensión del láser es la apropiada y la temperatura se encuentra dentro del rango de funcionamiento, se ilumina. Está etiquetado como «*tensión láser OK*».

**Led rojo:** Conectado al pin 3. Cuando la temperatura de la cabeza del láser es la adecuada para trabajar, está encendido. Es el etiquetado como «*Temp. Cabeza láser OK*».

Como se indica en la sección 2.2.1, uno de los componentes principales es la tarjeta Arduino Uno que suministra el control e interacción de los diversos componentes electrónicos del dispositivo de control, además de proporcionar la alimentación para los mismos. Según la hoja de datos de esta tarjeta (Anexo E), su propia tensión recomendada de alimentación



debe de estar dentro del rango de 7 a 12 V. Dispone de un conector para poder ser alimentada por una fuente de alimentación externa, empleada en este diseño como conector de alimentación externo general del prototipo. Los otros componentes empleados son circuitos electrónicos básicos y sus elementos asociados, como resistencias, condensadores, etc.

## 2.2. Componentes

El prototipo realizado está formado por varios componentes, unos exteriores (potenciómetro rotatorio y pantalla LCD), con los cuales el usuario podrá interactuar y otros interiores (DS1804, Arduino Uno y placa PCB) que servirán para poder llevar a cabo las distintas funciones a realizar, figura 2.4.

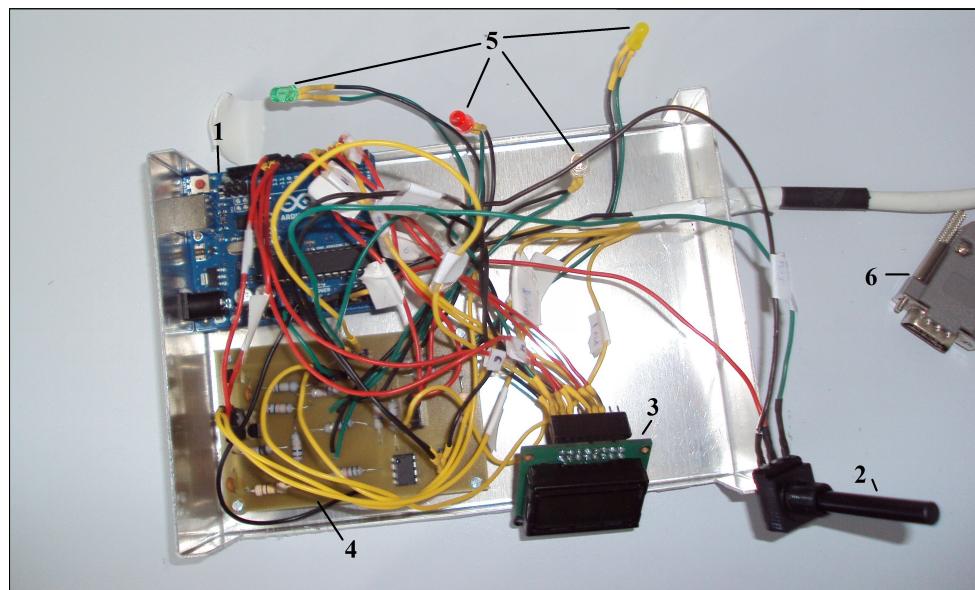


Figura 2.4: Componentes. 1. Tarjeta ARDUINO, 2. Potenciómetro rotatorio, 3. Pantalla LCD, 4. Tarjeta PCB, 5. LEDs, 6. Conector RS232.

### 2.2.1. Arduino Uno

La tarjeta Arduino es un dispositivo a través del cual los usuarios pueden controlar diversos mecanismos. Esta tarjeta es programable con la ayuda de un ordenador y realizada bajo las características del *open Source Hardware* [20].

Su principal componente es el microcontrolador ATmega328 y una de sus mayores ventajas es que dispone de un software libre, de forma que la creación de prototipos rápidos y baratos, fáciles de mantener, es más sencilla que con el uso de plataformas cerradas.

Los principales motivos por los que se ha escogido Arduino Uno son el precio, que funciona con un software libre y que es más fácil de manejar una herramienta ya creada



## 2.2. COMPONENTES



que realizar el diseño y la posterior fabricación de la placa electrónica (PCB) para los distintos circuitos de control que se necesitarían para realizar las múltiples funciones asociadas al dispositivo de control objeto de este proyecto.

Esta tarjeta proporciona la posibilidad de leer una gran variedad de instrumentos como interruptores, sensores, etc. En este proyecto se utiliza para leer datos de un potenciómetro rotatorio y transmitirlos a un potenciómetro digital tras realizar una transformación de los datos, a la par con otras tareas, como transmitir información a la pantalla LCD, leer datos procedentes del láser y transmitir información a la unidad de potencia del láser.

Hay que tener en cuenta que la tensión de entrada a los pines del Arduino no puede sobrepasar los 5V, ya que si esto ocurre se puede quemar por sobretensión. Durante el desarrollo de este proyecto, no hubo que preocuparse por ello ya que las entradas a la placa Arduino nunca exceden los cinco voltios. Estas entradas son tres. Una correspondiente a la lectura del potenciómetro rotatorio, que no excede los cinco voltios por estar alimentado el potenciómetro con el polo de cinco voltios de la propia placa y las otras dos entradas, correspondientes a las salidas de corriente y potencia del láser, cuyo orden de magnitud es de milivoltios. Las tres entradas se corresponden con conexiones analógicas en la placa Arduino Uno, los «*analog pins*» de la figura 2.5.

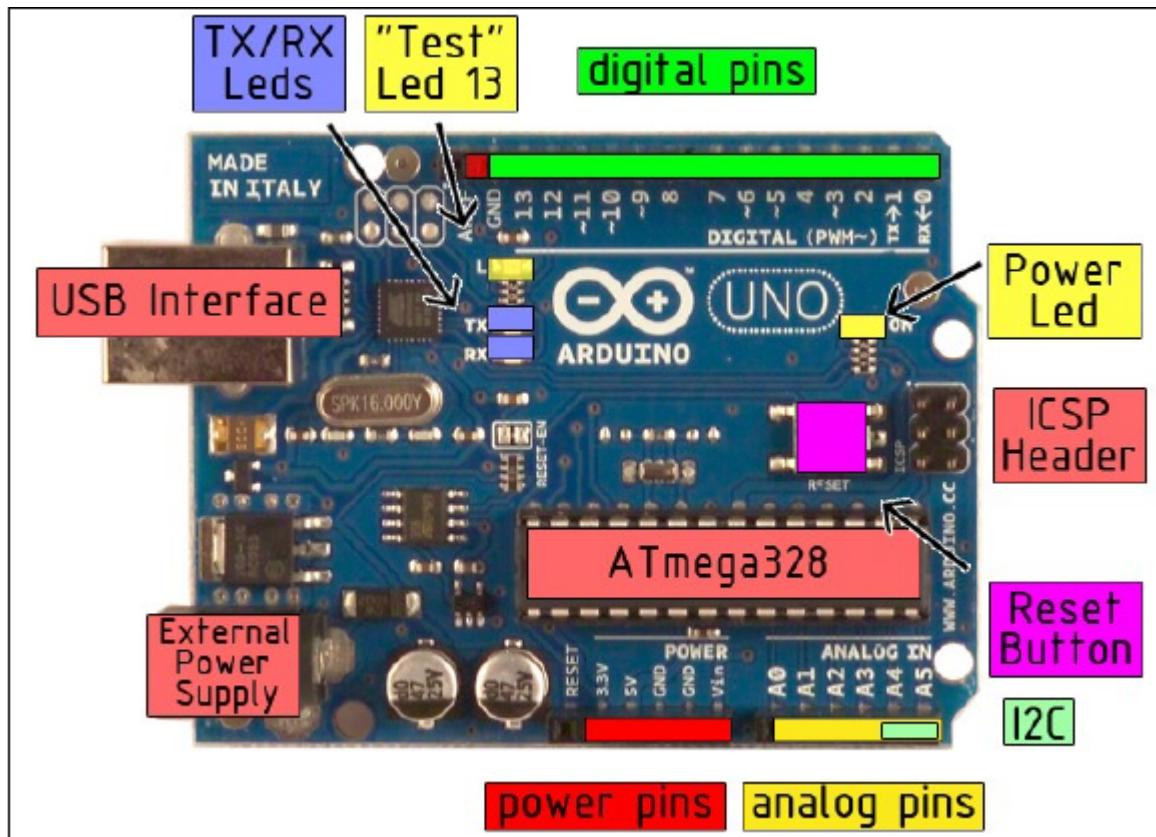


Figura 2.5: Arduino Uno (Anexo E).

La frecuencia de reloj de la tarjeta Arduino Uno es de 16 MHz, lo que implica que todos los chips empleados en este diseño tienen que tener una frecuencia de reloj compatible



con esta. Esta tarjeta cuenta también con un botón de reset que se ha comunicado con el exterior de la caja, para poder reiniciar el programa a través de este botón en el caso de que sea necesario.

### 2.2.2. Potenciómetro digital DS1804

Este componente se encuentra integrado en la tarjeta PCB, en la cual se engloban todos los componentes electrónicos del controlador como resistencias, etc. Es uno de los componentes claves en el desarrollo del proyecto, ya que se encarga de suministrar el intervalo de tensión necesario para la variación en la potencia del laser.

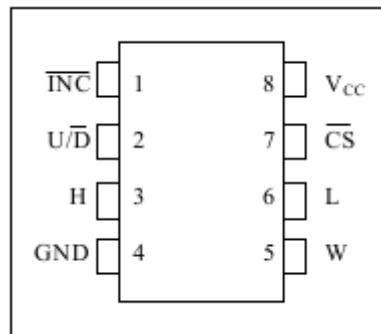


Figura 2.6: DS1804 (Anexo B).

Es un potenciómetro digital de 100 valores de resistencia, accesibles desde el terminal W. La resistencia del dispositivo es manipulada por un puerto de tres terminales que proporciona incrementos o decrementos en la resistencia en saltos de 1. Este puerto está compuesto por las entradas de control  $\overline{CS}$ ,  $\overline{INC}$  y  $U/\overline{D}$ , ver figura 2.6.

En el encendido, la resistencia del potenciómetro digital se carga en un tiempo máximo de  $500 \mu s$  una vez que la fuente de alimentación es estable. Además, el puerto de tres terminales está activo después de 50 ms. En la programación desarrollada se tienen en cuenta estos tiempos, así como la frecuencia de reloj del Arduino Uno.

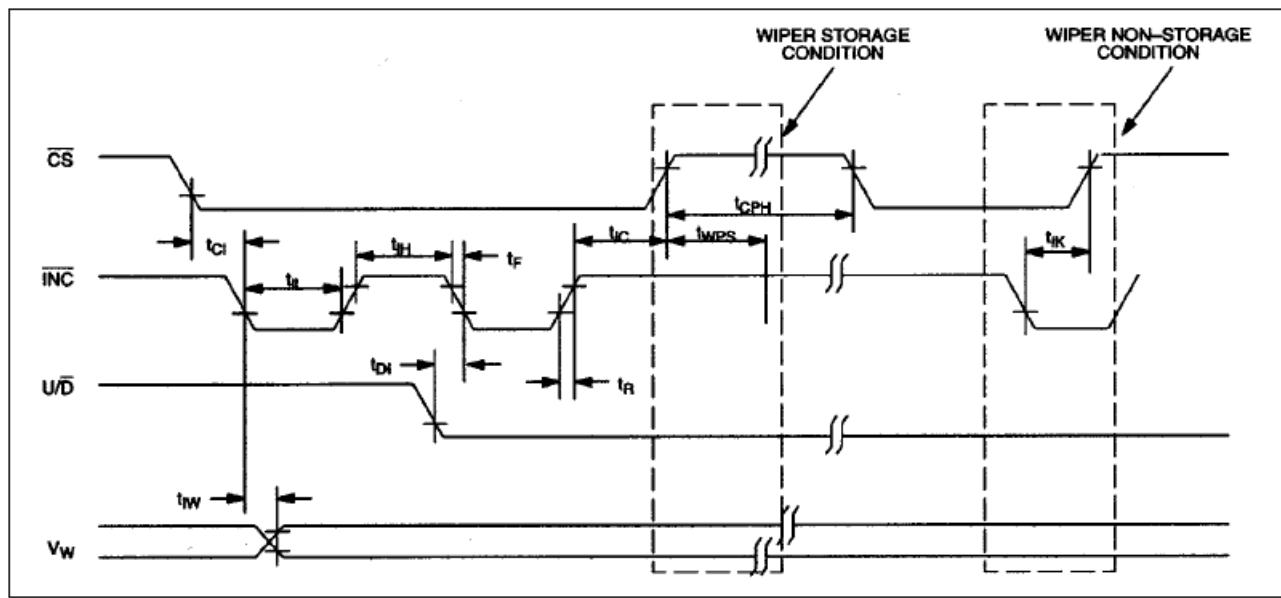


Figura 2.7: Funcionamiento del puerto de tres terminales (Anexo B).

Como se observa en la figura 2.7, para que se pueda realizar un cambio en la posición de potenciómetro digital DS1804 el terminal  $\overline{CS}$  debe de estar bajo (0) y el terminal  $\overline{INC}$  deberá llevar a cabo tres cambios en su estado (bajo-alto-bajo). El terminal  $U/\overline{D}$  indicará si el movimiento de la posición y por tanto el cambio de la resistencia es incremental (estado alto) o en decremental (estado bajo).

El diagrama de bloques lógico del CS1804 se muestra en la figura 2.8.

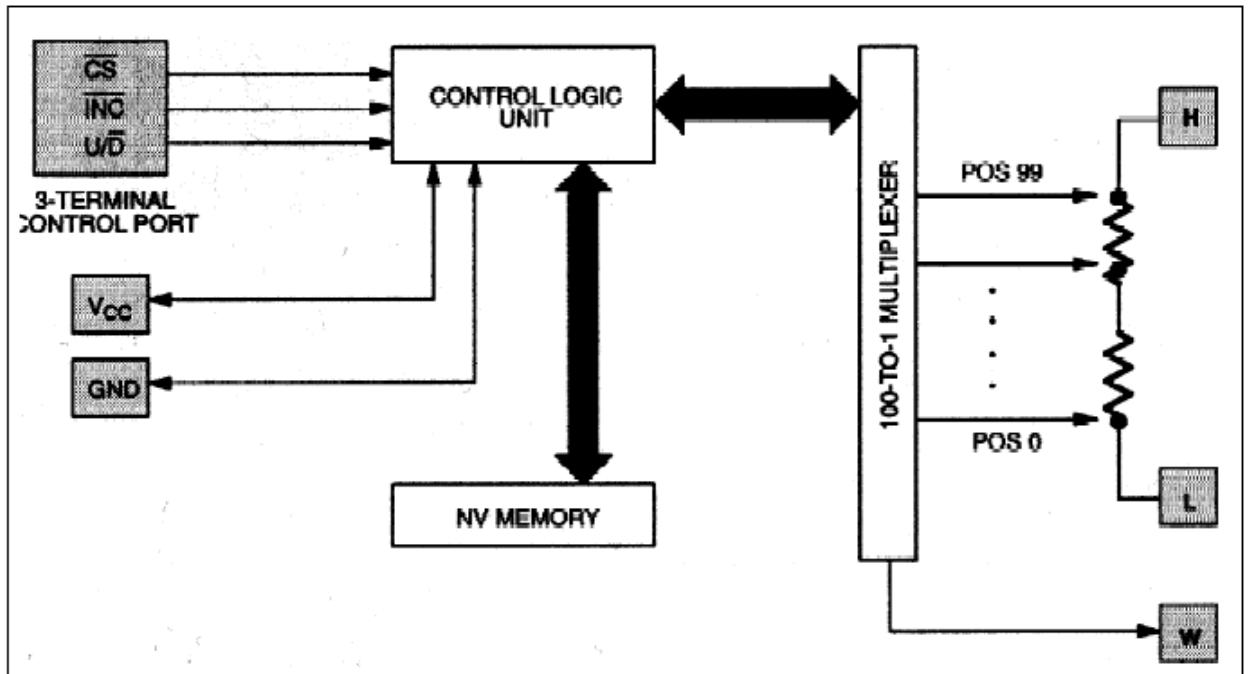


Figura 2.8: Diagrama de bloques del DS1804 (Anexo B).

Su parte principal es la unidad de control lógica, controlada por el puerto de tres



terminales descrito previamente. Conectada a tierra y alimentada por un terminal de 5V, esta unidad lógica cuenta también con una memoria interna que permite almacenar la última posición y resistencia a la que el potenciómetro digital DS1804 estuvo trabajando antes de ser apagado.

Como salida de esta unidad lógica, existen tres terminales (H, L y W). Con los terminales H y L se definen la tensión máxima y mínima (5V y 0V para este diseño) que el potenciómetro suministra. Este intervalo de tensión está dividido en 100 posibles posiciones de forma que a cada una de estas posiciones le corresponde un valor de tensión único. La posición en la cual se encuentra actualmente viene marcada por el terminal W.

Dentro del amplio número de resistencias que este chip ofrece, se ha escogido el modelo de  $50\text{ K}\Omega$ , por que según el circuito de la figura 2.2.1, es este el valor preciso para obtener el intervalo de tensión necesario para controlar la intensidad del láser.

### 2.2.3. Potenciómetro rotatorio

Es una resistencia variable de forma manual. A través de esta resistencia variable, se le comunican a la tarjeta Arduino Uno las resistencias que transladadas a tensión indican los cambios en las posiciones del DS1804.

### 2.2.4. Pantalla LCD

Dispositivo externo de 36,2 mm x 23,5 mm que es alimentado por la propia tarjeta Arduino, figura 2.9. Básicamente muestra información sobre los cambios de posición que se llevan a cabo en el potenciómetro digital DS1804, la tensión de salida que el usuario ha seleccionado, la intensidad lumínica correspondiente a esa tensión y al momento a partir del cual se pueden realizar cambios en el potenciómetro rotatorio para que el sistema responda según su diseño.

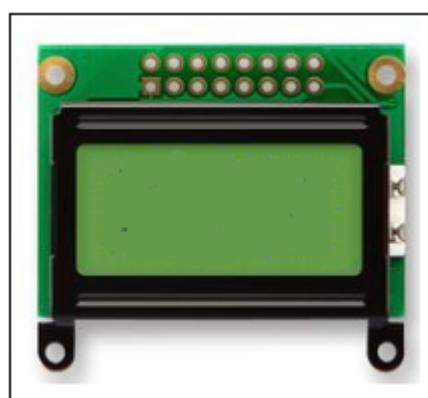


Figura 2.9: LCD empleada en este proyecto. [<http://uk.farnell.com/fordata/fdcc0802c-rnnybh-16le/disp-8x2-stn-lcd-3v-reflective/dp/1847923>].

Este modelo de pantalla LCD posee 16 patillas, las cuales se pueden emplear en su



programación siguiendo la interfaz proporcionada por el fabricante, ver apéndice D para los detalles. Así se tendría acceso a cada uno de los segmentos de la pantalla, pero para el uso necesario en este proyecto se prefirió emplear las librerías para programación de LCD disponibles para la tarjeta Arduino Uno [13]. Por que con su uso, en unas sencillas líneas de código se consigue que la pantalla muestre la información relevante sin necesidad de programar cada segmento de la pantalla.

### 2.2.5. Tarjeta PCB

Placa de 88,9 x 55,9 mm sobre la cual está impreso todo el circuito diseñado y utilizado en este proyecto, figura 2.10.

Esta placa está realizada únicamente sobre una de sus caras (cara top), ya que no se trata de un circuito muy complejo y se puede realizar perfectamente de esta manera, con el consiguiente ahorro tanto de materiales como económico que esto reporta. Está compuesta de varios componentes eléctricos (resistencias, condensadores, transistores y leds) que añadidos al circuito, se encargan de comunicar entre si las distintas partes del mismo. Una de sus partes más destacadas es el chip DS1804, cuyo funcionamiento se describe en la sección previa 2.2.2.

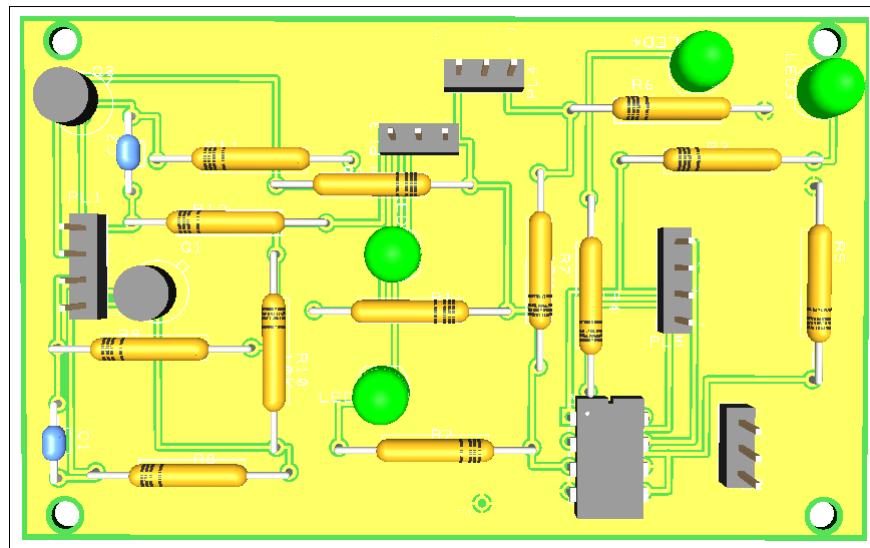


Figura 2.10: Renderización digital de la placa PCB. El chip del potenciómetro digital está situado en la esquina inferior derecha.



## 2.3. Descripción de los circuitos

En esta sección se comenta el desarrollo del circuito utilizado, las partes de las que este está compuesto y su funcionamiento. Para ello, se comenzará por describir el funcionamiento y diagramas lógicos de las diversas partes de la placa, para terminar la sección con una breve descripción de la tarjeta en su conjunto.

### 2.3.1. Variador de tensión

El circuito encargado de variar la tensión es el más importante de todos los circuitos diseñados, ya que será este el que se encargue de suministrar el rango de tensión que se busca. Este circuito esta compuesto por un potenciómetro rotatorio, un potenciómetro digital (DS1804), 3 resistencias (100KΩ, 2KΩ, 1,2KΩ) y dos leds (verde y amarillo), ver figura 2.11.

**Funcionamiento:** El usuario manipula el potenciómetro rotatorio, el cual envía la información a la placa Arduino. Esta información es empleada por la placa para ajustar el potenciómetro digital (DS1804) a través del puerto de tres terminales, colocándolo en la posición adecuada hasta alcanzar la tensión demandada por el potenciómetro analógico. Esta tensión se transmite a la unidad de control del láser, modificándose entonces la intensidad del mismo. La resistencia de 100KΩ actúa como divisor de tensión para que así la tensión saliente al PIN 8 del láser esté dentro del rango necesario.

La función de los led y las resistencias de 1,2 KΩ y 1KΩ es mostrar que se están realizando cambios en la tensión suministrada.

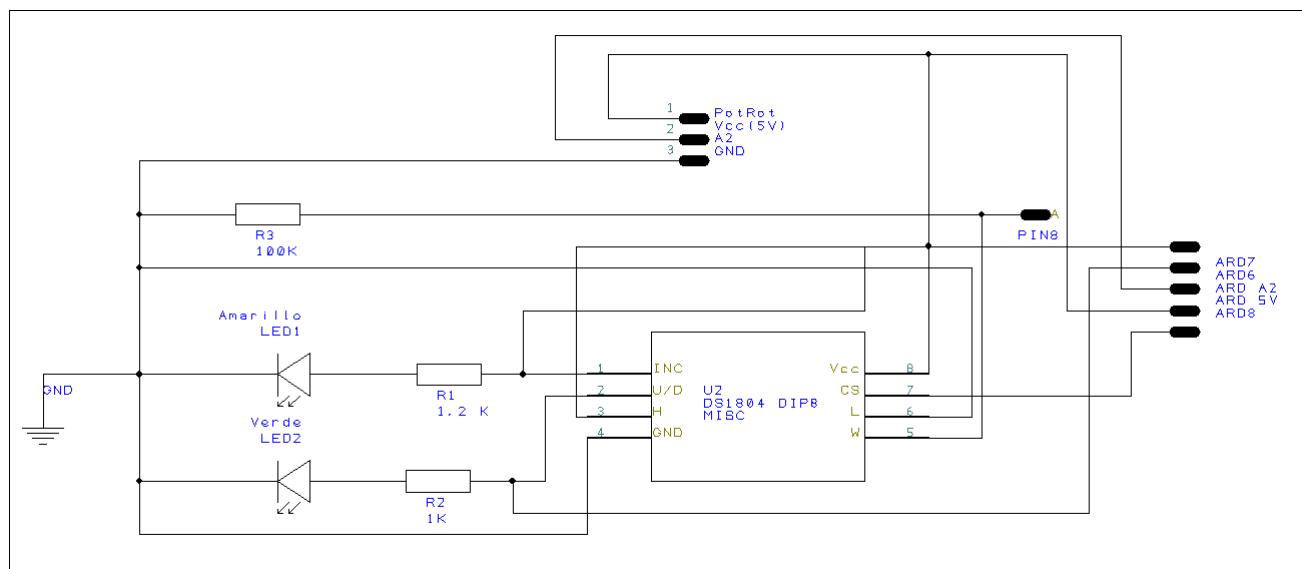


Figura 2.11: Esquema de conexiones lógicas del variador de tensión.



### 2.3.2. Pantalla LCD

Circuito compuesto de pocos elementos, ya que solo incluye dos resistencias ( $47\text{K}\Omega$  y  $1\text{K}\Omega$ ) y la propia pantalla LCD. Sin embargo, el número de conexiones es alto debido a la necesidad de controlar los segmentos de la LCD, figura 2.12.

**Funcionamiento:** La información que se desea presentar en la pantalla por la placa Arduino se envía mediante un bus de datos<sup>2</sup> a la pantalla LCD, para que muestre esta información.

Las resistencias de  $47\text{K}\Omega$  y  $1\text{K}\Omega$  forman un divisor de tensión encargado de regular el contraste de la pantalla LCD. Si se quiere disminuir el contraste, solo hay que disminuir la resistencia de  $47\text{K}\Omega$  o aumentar la resistencia de  $1\text{K}\Omega$ , ya que de ésta cae más tensión en la parte del divisor que va a tierra, disminuyendo así la tensión suministrada al pin de la pantalla LCD encargado del contraste de la misma, lo que se traduce en una disminución del contraste, y viceversa.

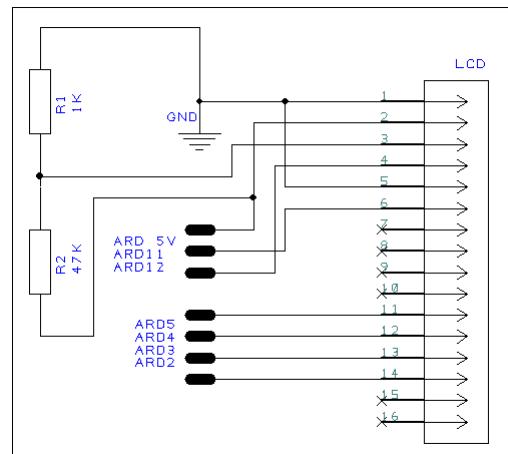


Figura 2.12: Circuito de pantalla LCD.

### 2.3.3. Lectores del láser

Se trata de dos circuitos idénticos, encargados de leer la corriente y la potencia de salida del láser. Cada uno de estos circuitos está compuesto por un transistor C546, un condensador de  $100\text{nF}$  y 3 resistencias, dos de  $10\text{K}\Omega$  y una de  $100\text{K}\Omega$ , ver figura 2.13.

La señales que informan de la corriente y potencia de salida del láser son suministradas en mV por el propio láser. El objetivo de estos circuitos es poder amplificar esta señal de tal forma que la tarjeta Arduino pueda leer los valores sin que el valor máximo sobrepase los cinco voltios.

Estos circuitos se han realizado físicamente, aunque la información que proporcionan a la placa Arduino no se ha empleado en este proyecto.

<sup>2</sup>Un conjunto de líneas digitales de la placa Arduino



## 2.3. DESCRIPCIÓN DE LOS CIRCUITOS



**Funcionamiento:** Se ha utilizado una configuración en emisor común para el transistor. Por lo tanto, la señal se aplica a la base del transistor y se extrae por el colector mientras que el emisor está conectado a masa.

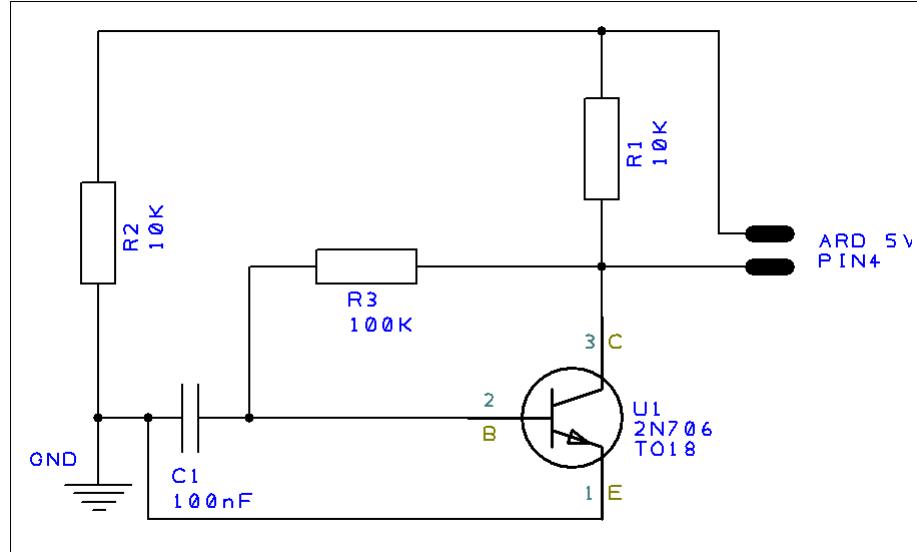


Figura 2.13: Circuito del transistor para lectura de corriente y potencia.

Si el circuito se emplea para leer la corriente del láser, el pin de entrada es el PIN 4 del láser y el de salida la entrada 9 del Arduino, mientras que si se lee la potencia los pines de entrada del láser y de salida del Arduino son el PIN 5 y la entrada 10, respectivamente.

### 2.3.4. Indicadores

Es un pequeño circuito, compuesto por dos resistencias ( $1\text{K}\Omega$ ) y dos led (rojo y blanco) encargados de indicar cuando la temperatura y potencia de funcionamiento del láser son las adecuadas para poder empezar a utilizarlo. Ambos circuitos son idénticos, cambia solo el color del LED, figura 2.14.

**Funcionamiento:** Las salidas de los pines PIN1 y PIN 3 del láser envían una señal cuando la temperatura y tensión del láser, respectivamente, son las adecuadas para funcionar. Esta señal es un voltaje que hace que los led se iluminen. La función de las resistencias es impedir que los led puedan ser dañados por una tensión alta.

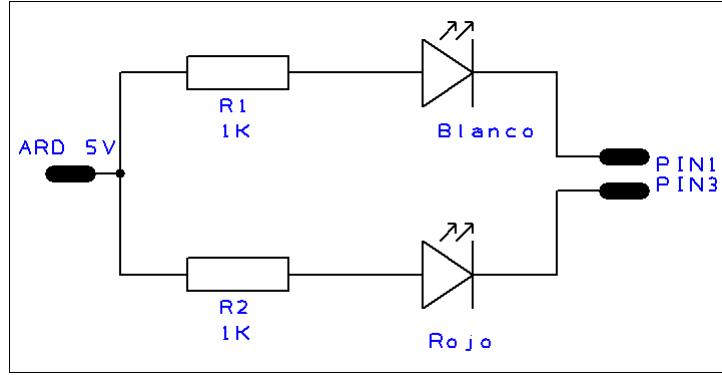


Figura 2.14: Circuito indicador.

### 2.3.5. Tarjeta PCB

Para terminar esta sección se presenta el diagrama lógico con todos los elementos previos unidos. Este circuito se diseñó para alojar en una sola tarjeta los circuitos anteriormente descritos. Está compuesto por todos los elementos citados anteriormente, salvo el potenciómetro rotatorio y la pantalla LCD que están instalados externamente a este circuito, aunque estos elementos tienen sus conectores asociados incluidos en la placa final.

La figura 2.15 muestra el cableado lógico de la tarjeta, y en la siguiente sección se desarrolla su diseño, construcción y las diversas etapas del mismo.



## 2.3. DESCRIPCIÓN DE LOS CIRCUITOS

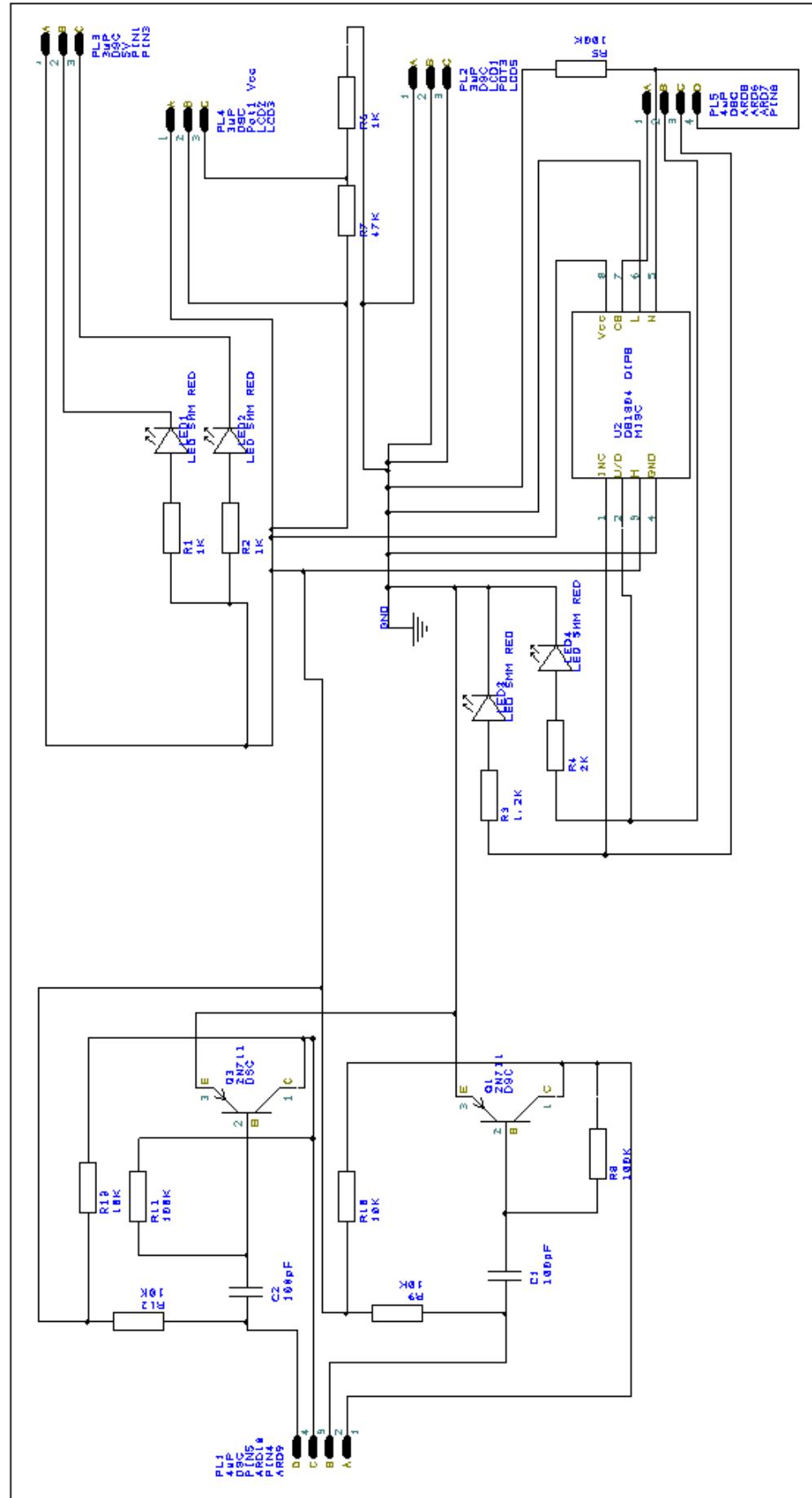


Figura 2.15: PCB.



## 2.4. Desarrollo de la tarjeta PCB

En esta sección se procede a dar un pequeño paseo por los distintos procesos que han transcurrido para la obtención final de la PCB, incluyendo el circuito con el que funciona el dispositivo encargado de regular la tensión suministrada al láser. El diseño de esta tarjeta ha sido realizado a través del programa *DesignSpark* [9] y para la obtención final de esta tarjeta, son necesarios una serie de pasos:

1. Diseño del diagrama esquemático del circuito.
2. Paso del diagrama esquemático al formato de circuito impreso.
3. Reestructurar el circuito, recolocando los elementos que lo forman y colocando las vías que conectan los nodos eléctricos de tal forma que éstas no se crucen entre ellas.
4. Fabricación física de la PCB.
5. Soldado de los elementos del circuito (resistencias, transistores, leds,...) a la PCB.

### 2.4.1. Esquema del circuito

Lo primero que se ha realizado es un boceto del circuito sobre papel, figura 2.16.



## 2.4. DESARROLLO DE LA TARJETA PCB

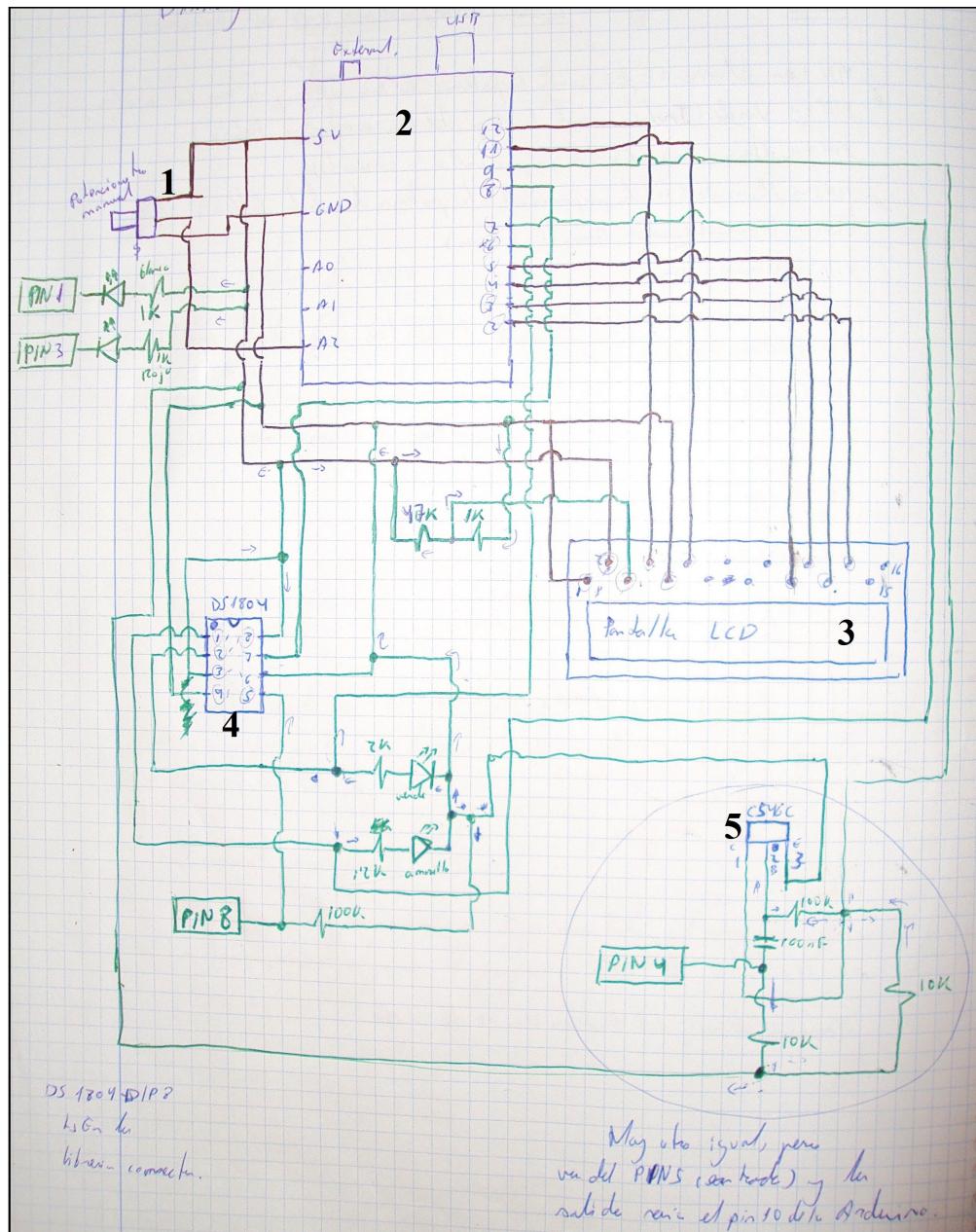


Figura 2.16: Boceto inicial del circuito de la PCB. 1. Pot. manual, 2. Arduino, 3. LCD, 4. Potenciómetro digital, 5. Transistor.

En el boceto de la figura 2.16 se puede observar el potenciómetro rotatorio y la pantalla LCD, elementos que como ya se indicó en apartado 2.3.5 no están incluidos en el diagrama esquemático, pero la presencia en el boceto sirve para saber dónde van conectadas las distintas vías y elementos. En el boceto solo se bosquejó uno de los circuitos correspondientes a la amplificación de las señales procedentes del láser que indican la corriente y la potencia con la que está trabajando, por que los dos circuitos son exactamente iguales. Por supuesto, ambos circuitos se escriben en el esquema.

Una vez que se han corregido los posibles errores, se procede a dibujar el esquema del circuito mediante el programa *DesignSpark* obteniendo el circuito de la figura 2.15, ya descrito anteriormente.



## 2.4.2. Diseño del circuito impreso

En este punto del proceso se inicia una etapa automática, ya que es el propio programa *DesignSpark* el encargado de hacer la primera PCB. Para ello, lo único que hay que hacer es indicarle las unidades métricas (mm) y el número de capas a emplear en la placa. En el circuito de este proyecto, con una sola capa para colocar componentes fue suficiente. En la figura 2.17 se puede ver el aspecto inicial de la PCB, en la cual todos los elementos y vías están todavía lejos de sus posiciones finales.

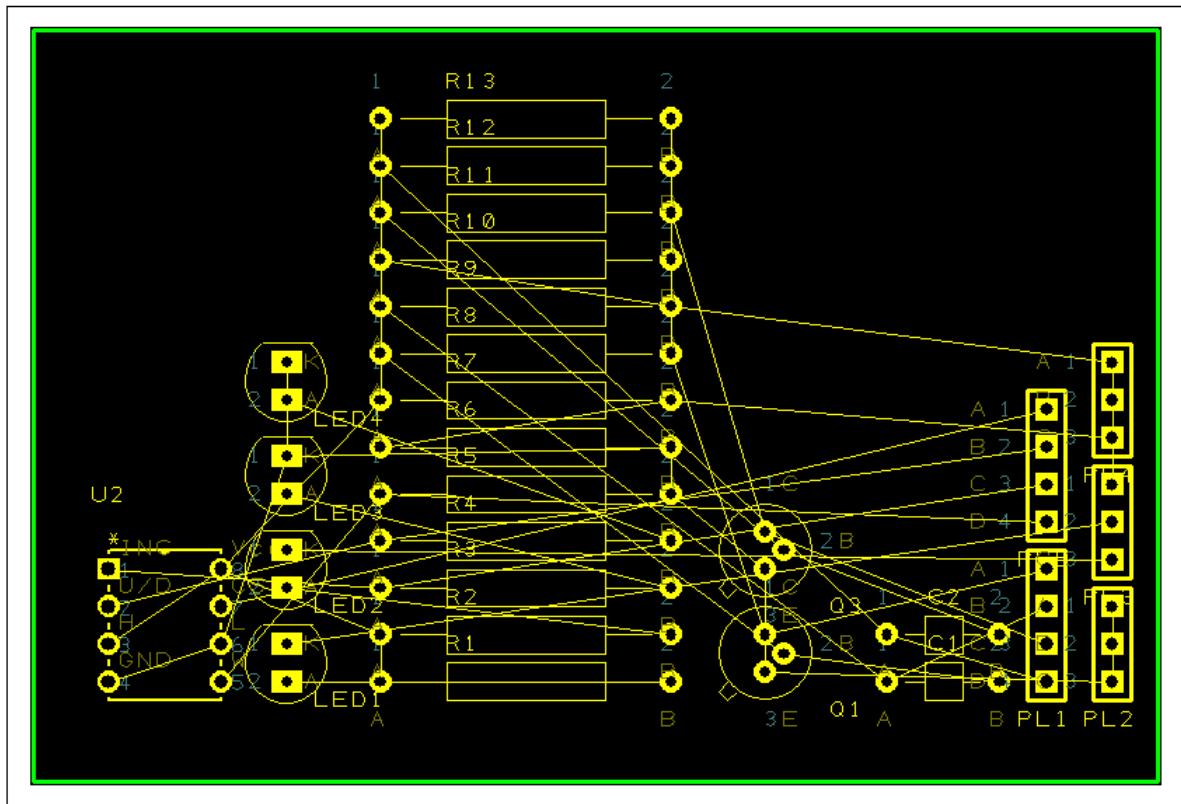


Figura 2.17: PCB inicial.

Como se puede observar en la figura, el circuito impreso inicial generado por el programa dista mucho de ser el ideal, de tal modo que se reestructura colocando los distintos elementos que lo componen y las conexiones entre ellos (vías), sin que existan cruces de vías que provoquen que el circuito impreso final no funcione.

Una vez modificadas la posiciones de los distintos elementos para tener un circuito impreso que se ajuste al diseño, se puede observar la diferencia en la figura 2.18. Como se observa, no hay cruces de vías que puedan impedir que la tarjeta PCB funcione correctamente.

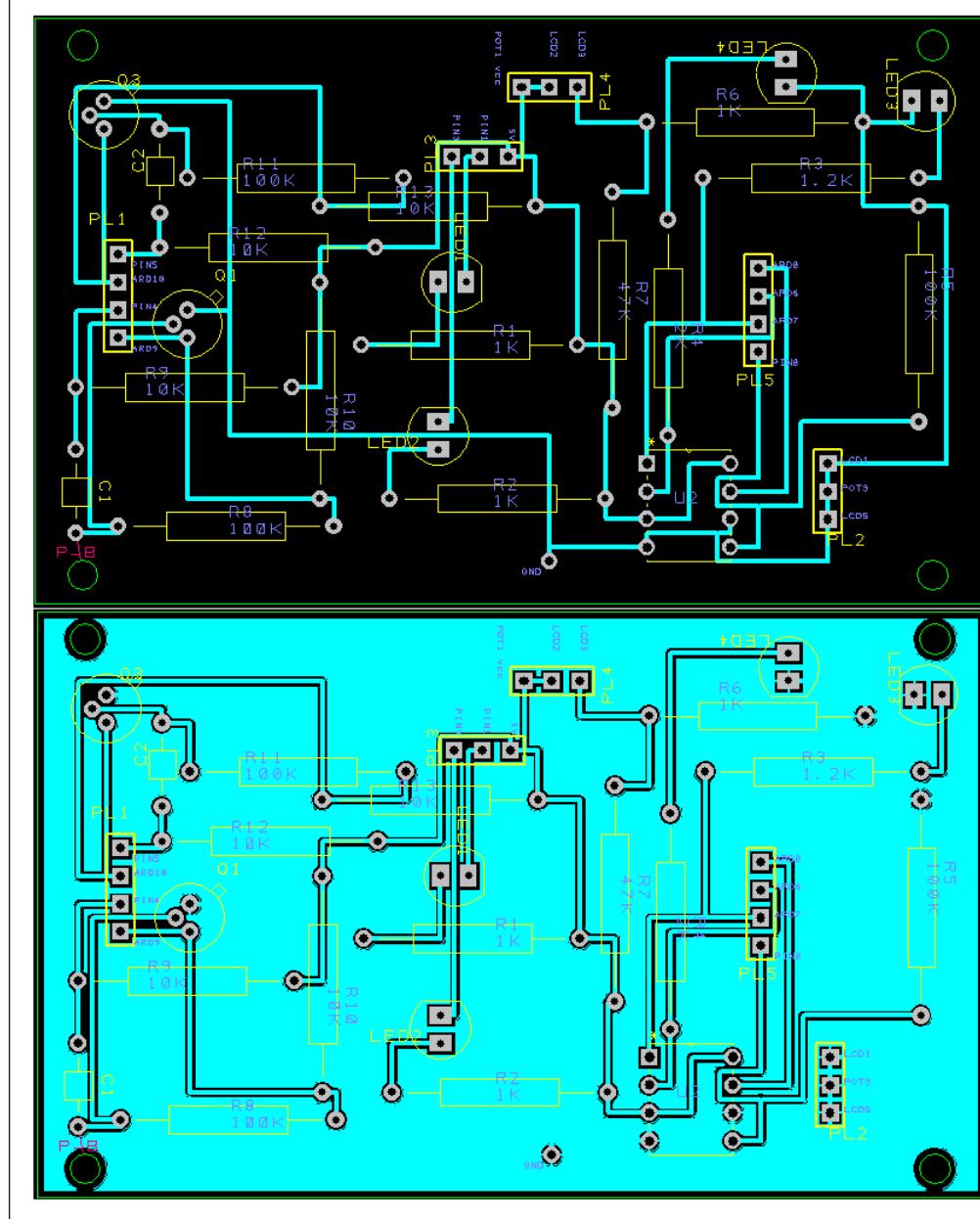


Figura 2.18: Circuito impreso, forma final (Sin recubrimiento y con recubrimiento de cobre).

Como último paso, se añade una conexión común a tierra para todos los elementos mediante una placa común de cobre otorgando a la PCB las siguientes características:

- Debido a que la gran mayoría de los elementos usados están conectados a tierra, la existencia de esta tierra común ofrece un ahorro en el número de líneas necesarias, lo que a su vez también facilita el diseño de la PCB.
- Al disponer de una tierra tan grande, el circuito es menos propenso a interferencias entre los distintos nodos ya que si existen frecuencias altas, éstas se irán a tierra sin llegar a interferir en el correcto funcionamiento del circuito.



## 2.4. DESARROLLO DE LA TARJETA PCB



Una vez realizado esto, el programa puede mostrar una vista tridimensional renderizada de la PCB con sus elementos correspondientes, figura 2.19.

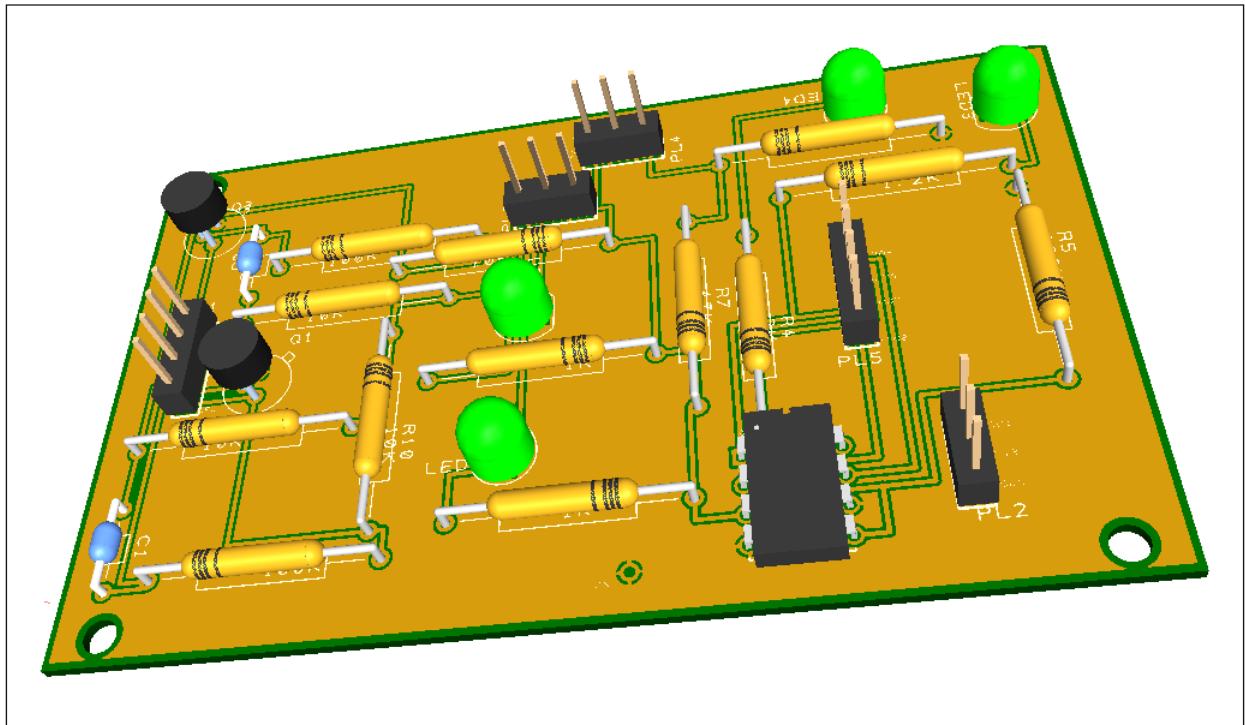


Figura 2.19: PCB tridimensional. Nótese la presencia de los conectores y de los leds pegados a la placa.

### 2.4.3. Fabricación de la PCB

Una vez desarrollada gráficamente la tarjeta PCB, se procede a su fabricación física, proceso llevado a cabo por los técnicos de la escuela a partir de los documentos correspondientes donde se detallan las distintas características de la PCB. Cuando la PCB se ha construido y comprobado, se soldaron sus diversos elementos a la misma, obteniendo la placa final.

# Capítulo 3

## Software

Para poder llevar a cabo las operaciones necesarias para conseguir un correcto funcionamiento del sistema, no sólo es necesaria la de una plataforma física, sino también la programación de un software que se encargue de manejar y controlar esta plataforma física para que funcione conforme a las especificaciones de diseño. El software programado en este proyecto se ejecuta sobre la plataforma Arduino, sobre la placa Arduino Uno, encargada de controlar los elementos integrantes en el proyecto, como ya se comentó en capítulos anteriores.

En este capítulo se explica el software desarrollado así como su funcionamiento y la programación del mismo.

### 3.1. Programación de la tarjeta Arduino Uno

En la programación de la tarjeta Arduino se utiliza el lenguaje de programación *wiring* [11] cuyas características son semejantes a las del lenguaje C++ [11] y está basado en el lenguaje *processing* (lenguaje de programación y entorno de desarrollo integrado basado en java) [11].

Este lenguaje de programación presenta una serie de ventajas, como pueden ser la utilización de bibliotecas ya creadas o la creación de nuevas bibliotecas, además de ser código abierto y de uso no muy complicado.

La compilación del software y su introducción en la tarjeta Arduino se hace a través del entorno de desarrollo integrado basado en java cuyo aspecto se puede ver la figura 3.1.



## 3.2. DISEÑO E IMPLEMENTACIÓN



The screenshot shows the Arduino IDE interface. The code in the main window is as follows:

```
// Final22 | Arduino 1.0.1
Archivo Editar Sketch Herramientas Ayuda

Final22

// a = (99-60)/1023; Cálculo de la constante a, necesaria para pasar las 1023 posiciones del potenciómetro rotatorio a las 40 posiciones que quedan del DS1804.
a = 0.038124; //Constante para la transformación lineal de las posiciones del potenciómetro manual al digital
valorPot = analogRead(pinPot); //Iguala la variable valorPot a la lectura hecha por el arduino de la entrada pinPot.
valorDig = (valorPot * a) + 60; /*Transformación lineal. Del manual al digital. Se le suma 60 para que la posición mínima en la que pueda empezar el potenciómetro digital sea la
                                posición 60*/
valorDigInt = (int)valorDig; //Pasa el valor del número de posiciones de la primera medida del potenciómetro digital a un número entero

digitalWrite(UD,HIGH);
for(contador = 0; contador <= valorDigInt; contador++)
{
    digitalWrite(INC,LOW);
    delay(10);
    digitalWrite(INC,HIGH);
    delay(10);
    digitalWrite(INC,LOW);
}
digitalWrite(UD, LOW);
digitalWrite(CS, HIGH);

/*Para mostrar la posición inicial en la pantalla del ordenador, instaurar las dos siguientes líneas
Serial.println("Posición inicial.");
Serial.println(valorDigInt);
*/



voltInic= 0.829 * exp(0.018*valorDigInt);
//Configuración de la pantalla LCD
lcd.begin(8, 2); //Ocho columnas y dos filas.

Compilación terminada

Tamaño binario del Sketch: 8.226 bytes (de un máximo de 32.256 bytes)
```

In the status bar at the bottom right, it says "Arduino Uno en COM3".

Figura 3.1: Entorno de desarrollo integrado basado en java.

Usando este entorno se escribe el software necesario, se verifica que esté libre de errores y se compila cargándolo a la plataforma física, en este caso la tarjeta Arduino Uno.

## 3.2. Diseño e implementación

El código fuente original se incluye completo en el anexo G. En esta sección se irán repasando las distintas divisiones que presenta y sus funciones principales. El programa se divide en tres bloques distintos:

- Declaración de variables.

Es el primer bloque y permite la definición de las diversas variables y librerías a emplear.

- Bloque setup.

Bloque en el que se definen los pines físicos de la tarjeta como de entrada o salida.

- Bloque loop.

Es el elemento principal del programa.

### 3.2.1. Declaración de variables

En este bloque inicial se lleva a cabo la declaración de todas las variables que se van a utilizar a lo largo del programa y también se realiza la carga de las bibliotecas necesarias. En este caso, las correspondientes a las operaciones matemáticas y control de la pantalla LCD.



Entre todas las variables declaradas, se pueden destacar las siguientes, ya que son las que interactúan con los elementos físicos:

- **int** pinPot
- **int** CS
- **int** U/D
- **int** INC

La variable **pinPot** es la encargada de leer en cuál de las 1023 posiciones posibles se encuentra el potenciómetro rotatorio o a qué posición ha sido cambiado.

Las variables **CS**, **U/D** e **INC**, son las encargadas de manejar el puerto de tres terminales que controla al potenciómetro digital. La variable **CS** se encarga de establecer si se realiza o no una modificación, estados alto y bajo, respectivamente. La **U/D** indica al potenciómetro digital si las posiciones que marcan el valor de la resistencia deben incrementarse o si disminuyen. La variable **INC** indica los cambios de posición mediante su cambio de estado. Como el cambio de posición en el potenciómetro se produce cuando la variable pasa del estado alto al bajo, para asegurar un cambio de estado se emplea la secuencia bajo - alto - bajo en el programa realizado para este proyecto.

Además de estas variables se definen un gran número de variables internas y de control, que se usan como constantes de transformación, para el almacenamiento de datos, limitadores, etc.

Entre estas variables, una a destacar es la variable **voltaje**, que suministra la salida de voltaje elegida por el usuario a través del potenciómetro rotatorio. Una variable importante para el funcionamiento del programa es la denominada **contador**, variable encargada de delimitar las posiciones máximas y mínimas en el programa que puede alcanzar el potenciómetro digital, así como marcar en el programa la posición en la que se encuentra el potenciómetro digital en cada momento.

Las variables utilizadas en el programa se listan en las siguientes líneas:

```
//Variables, de entradas y salidas de la tarjeta arduino
int pinPot = A2; //Salida del potenciómetro.
int CS = 8; //Interrumpor del DS1804.
int UD = 6; //Dirección del DS1804.
int INC = 7; //Posiciones a cambiar en el DS1804.

//Variables utilizadas por el programa, para realizar las distintas acciones.
float valorPot; //Primera lectura del potenciómetro manual.
float valorPot2; //Segunda lectura del potenciómetro manual.
float difPot; // Incremento de las lecturas del potenciómetro manual.
```



```
float valorDig; //Primera lectura del potenciómetro digital.  
int valorDigEnt; //Valor de valorDig en entero.  
float valorDig2; //Segunda lectura del potenciómetro digital  
int valorDig2Ent; //Valor de valorDig2 en entero  
float difDig; //Incremento de las lecturas del potenciómetro digital.  
int difDigEnt; //Valor de difDig en entero  
float voltInic; //Variable para almacenar el voltaje al inicio de la sesión.  
float DS1804; //Posición del DS1804, en la 2a lectura.  
int DS1804Ent; //Valor de DS1804 en entero  
int contador = 0; //Marca el límite en las posiciones del potenciómetro digital.  
float a; //Constante en la transformación lineal.  
float voltaje; //Valor final de la tensión(V)  
float intlum; //Valor de la intensidad lumínica(mW)
```

### 3.2.2. Bloque setup

El código escrito en este bloque se ejecuta una sola vez, al iniciarse el programa por primera vez como el bloque anterior.

En este bloque lo primero que se realiza es declarar las variables como de entrada o de salida según correspondan, como se observa en el apéndice G.

Según las especificaciones del diseño de este proyecto, la posición del potenciómetro digital debe ser función de la del potenciómetro rotatorio manual. Como se menciona en la subsección 2.2.2, el potenciómetro digital tiene una memoria interna que almacena la última posición en la cual ha sido utilizado y el potenciómetro manual puede ser manipulado mientras el sistema está apagado lo que implicaría que las posiciones de ambos potenciómetros no coincidan. Para evitar este posible desajuste, en esta sección del programa se realiza una primera lectura del potenciómetro manual y se lleva el digital a esa posición, evitando así que surja cualquier diferencia de posición entre ambos.

Para ello, se baja el potenciómetro digital 100 posiciones, lo que permite establecer que comience en la posición inicial independientemente de la posición que hubiera guardado por usos anteriores, a continuación se realiza una lectura del potenciómetro manual y mediante una relación de transformación se pasa esta posición a su equivalente del potenciómetro digital. Esta relación se midió en el transcurso del proyecto, y se puede escribir como:

$$y = 0,03225 x + 60, \quad (3.1)$$

donde  $y$  es la posición del potenciómetro digital y  $x$  la del manual.

Es importante resaltar aquí que el fabricante del láser indica en el manual que la tensión de entrada está en el rango de los 2,5 a 5 V, por lo que la tensión que se debe suministrar



al láser deberá estar siempre en este rango. Por ello, para evitar unas tensiones de salida inferiores a la mencionada, una vez hecha la transformación del valor del potenciómetro manual a posiciones del potenciómetro digital, expresión 3.1, a estas últimas se le suman 60 posiciones, ya que como se demuestra en el capítulo 5, es la posición anterior la que proporciona un voltaje de 2,5 V.

Como último paso de este bloque inicial, se transforma el valor de la resistencia suministrada en tensión y se muestra su resultado en la pantalla LCD del sistema de control.

### 3.2.3. Bloque loop

Este bloque como su propio nombre indica es un bucle<sup>1</sup> de forma que cada vez que llegue el final, volverá a ejecutar el bucle desde la primera instrucción de manera continua. A continuación se procede a describir la estructura de funcionamiento de este bloque.

Comienza con una primera lectura del potenciómetro rotatorio y su paso a la variable *pinPot*. Al ser un valor analógico, necesita transformarse en el equivalente digital para su uso con el potenciómetro digital. Para ello, se emplea la relación:

$$y = 0,03225 x, \quad (3.2)$$

donde *y* es la posición del potenciómetro digital y la *x* la del manual, como en la relación 3.1 anterior.

Los tipos de las variables empleadas está descrito en el anexo G, en el listado del programa.

Después de esta primera lectura, se vuelve a leer el potenciómetro rotatorio y si se ha modificado con una diferencia de posiciones mayor de 2 (debido a las posibles fluctuaciones del potenciómetro rotatorio) entre ambas, se considera que se está modificando su valor. Entonces se habilitan 3 segundos para que el usuario pueda modificar el potenciómetro rotatorio ajustándolo a la posición que busca. Es importante resaltar que fuera de este tiempo de establecimiento, indicado en la pantalla LCD por el programa, no se debe modificar el potenciómetro rotatorio, ya que al estar en el bloque loop, el sistema está continuamente leyendo el valor del potenciómetro manual y podrían mezclarse las lecturas, teniendo así que reiniciar el programa por falta de predictibilidad de los resultados.

Después de estos 3 segundos, se vuelve a leer el potenciómetro rotatorio considerando que el valor obtenido es el que el usuario decidió establecer finalmente. Esta lectura se almacena en la misma variable en la cual se almacenó el valor de la segunda lectura, *valorPot2*, sustituyendo el valor anterior.

En este momento el programa ya dispone de la posición inicial y la establecida por el usuario en el potenciómetro rotatorio. Restando estos valores, se obtiene el número de

---

<sup>1</sup> **Loop** es bucle en inglés.



posiciones que el potenciómetro digital tiene que cambiar para ajustarse al nuevo valor solicitado por el usuario.

Este cambio de posiciones en el potenciómetro digital se lleva a cabo a través del puerto de tres terminales controlado por las variables ***U/D***, ***CS*** e ***INC*** como ya se ha mencionado en el desarrollo del apartado 3.2.1. El número de veces que ***INC*** pasa por un ciclo bajo - alto - bajo se establece con un bucle de tipo ***for***. Cuando se ha terminado de cambiar el número de posiciones en el potenciómetro digital la variable ***CS*** vuelve a establecerse a nivel alto para evitar cualquier cambio en el potenciómetro digital.

Después de que el cambio de posiciones haya finalizado, las siguientes líneas de código transforman la posición final del potenciómetro digital en voltios siguiendo la expresión del cuadro 5.2, de la página 50.

Posteriormente se encuentran una serie de instrucciones que muestran en la pantalla LCD el número de posiciones que ha cambiado el potenciómetro digital y el valor correspondiente a la posición final establecida en voltios.

El valor del voltaje se transforma a su vez en intensidad del láser, ver sección 5.2, mediante la expresión del cuadro 5.4. Y como en el caso anterior, se emplea el mismo conjunto de instrucciones para mostrar esta intensidad en mW en la pantalla LCD.

Entonces se llega al final del bloque y este se volverá a reiniciar, pero si no existen cambios en la posición del potenciómetro rotatorio, no habrá ningún tipo de modificación y la pantalla LCD no mostrará ningún cambio, como muestran los diagramas de bloque especificados en el siguiente apartado.



### 3.2.4. Diagramas de bloque

En este apartado, se muestran los diagramas de bloques que representan el funcionamiento del programa descrito en las secciones anteriores.

Los bloques más generales del software vienen impuestos por la arquitectura de programación del Arduino y son tres, el inicial, el *setup* y el *loop*, figura 3.2.

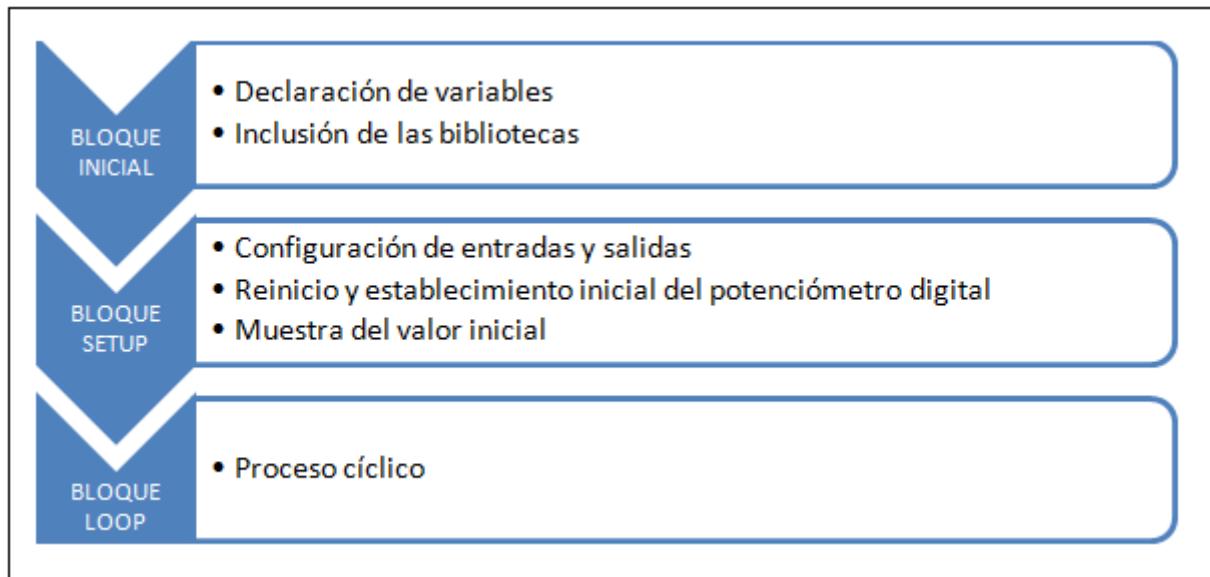
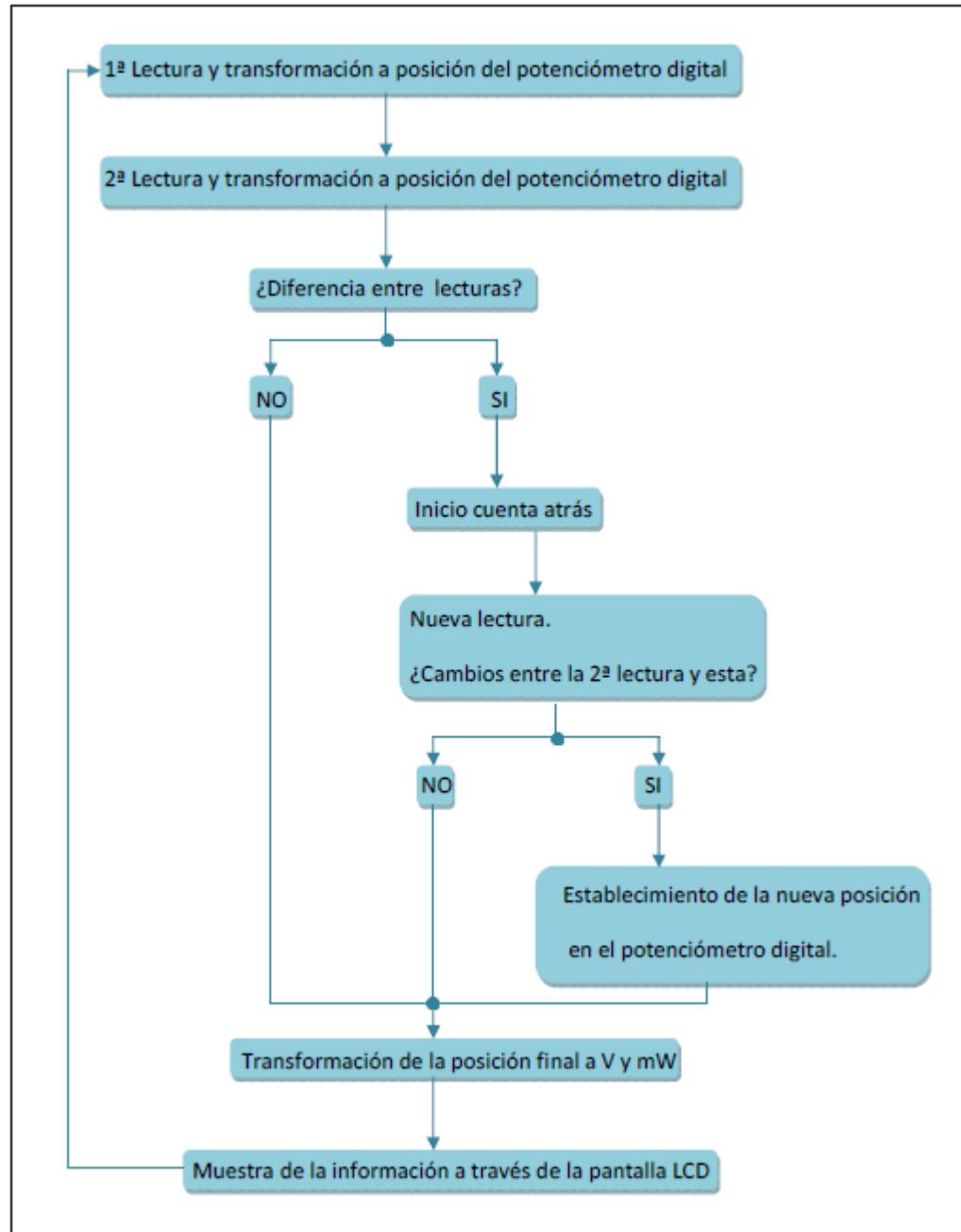


Figura 3.2: Bloques principales del software.

Como ya se ha indicado con anterioridad, el bloque *loop* es un proceso cíclico, donde se almacena el programa principal de este proyecto y cuyo diagrama de bloques de funcionamiento se puede apreciar en el esquema de la figura 3.3.

Figura 3.3: Diagrama del bucle *loop*.

Como muestra el diagrama anterior, el bloque *loop*, es un proceso cíclico encargado de llevar a cabo una serie de acciones en un orden preestablecido. Como se observa, el principal mecanismo utilizado es la comparación entre distintas lecturas, dependiendo de si existe o no diferencia entre estas, el programa cojerá un camino u otro. En el caso de no existir diferencias entre las lecturas, la información mostrada en la pantalla LCD no cambia. Y en el caso de que exista diferencias entre las lecturas, se sigue otro camino dentro del bloque *loop* que permite la realización del cambio en el potenciómetro digital, terminando este camino en el mismo punto de muestra de la información a través de la pantalla LCD. Una vez en este punto del bloque, se vuelve otra vez al principio volviendo a repetir el proceso.



### 3.3. Software empleado para la calibración del controlador.

Es un sencillo programa encargado de ir variando las posiciones del potenciómetro digital de una en una y en intervalos de 5 segundos de manera ascendente desde la posición 0 hasta la 99 y de forma descendente después, mostrando la en pantalla LCD cada cambio de posición. De tal forma que, como se ve en la sección 5.1, se puede observar experimentalmente el valor de tensión correspondiente a cada una de la posiciones mediante un polímetro acoplado a los pines que proporcionan este valor en el diseño.

Este software consta de los mismos tres bloques que el programa principal: declaración de variables, bloque *setup* y bloque *loop*, ver anexo H.

Debido a su uso, solo hay que declarar cuatro variables en el bloque inicial: las variables ***CS***, ***U/D***, ***INC***, que al igual que en el software principal se encargan de controlar el puerto de tres terminales del potenciómetro digital, y la variable ***contador***, que establece cuando el programa debe de dejar de seguir aumentando o disminuyendo posiciones porque se ha alcanzado el límite de posiciones, tanto inferior como superior.

En el bloque *setup*, las tres variables encargadas de controlar el puerto se declaran como variables de salida de la tarjeta y a continuación se procede a reiniciar el potenciómetro digital para que comience desde la posición 0. Primero se establece la variable ***CS*** a nivel bajo y la variable ***U/D*** a nivel alto y a continuación la variable ***INC*** se hace cambiar de nivel en tres ocasiones por cada posición que se desea cambiar, como se describió anteriormente. Una vez contadas 99 posiciones en la variable ***contador***, se procede al cambio de posiciones pero de manera descendente.

Para ello, el programa tiene el mismo código utilizado para ascender posiciones, pero en este caso la variable ***U/D*** se establece en *OFF* para que el cambio se haga de manera descendente. Entonces, el número de posiciones irá bajando hasta llegar a 0, que es el límite inferior de la variable ***contador***.

Este proceso de recorrer todas las posiciones existentes en el potenciómetro digital sólo interesa llevarlo a cabo una vez, es decir que no es necesario que el proceso esté ejecutándose en un ciclo infinito, por eso el grueso del programa se ha desarrollado en el bloque *setup*, de ejecución única. De esta manera, el bloque *loop* donde se programa el bucle que debe seguir la tarjeta Arduino está vacío en este programa.



# Capítulo 4

## Evolución del dispositivo

Este capítulo está dedicado a mostrar la evolución tanto del hardware del sistema como del software del dispositivo desde sus primeros prototipos hasta el resultado final presentado en este proyecto.

Merece la pena destacar en el desarrollo del dispositivo que ha medida que se iba avanzando en la parte física o hardware, se iba avanzando de forma semejante en programación del mismo software, un método de trabajo que permite ir comprobando el correcto funcionamiento y realización de la tarea designada a cada una de las partes instaladas.

### 4.1. Evolución del hardware

El cabezal de control del láser usa una señal analógica de voltaje para el control de la intensidad, por lo que lo primero que se hizo fue asegurarse de que la señal de tensión que le llegara al láser fuera analógica y no digital. Esta fue una de las razones de acudir a la tarjeta Arduino Uno, que es capaz de transformar las señales digitales en analógicas.

Hasta llegar al diseño final del hardware utilizado en el circuito, se pasó por distintas etapas evolutivas. En un primer diseño se pensó en usar el regulador de tensión *LM317* como el dispositivo modificador de tensión, sin embargo se descartó por su ineffectividad para el objetivo que se pretende conseguir con este proyecto, ya que el regulador de tensión lo único que hace es ajustar la tensión a un único valor, cuando lo que se necesita es poder suministrar distintos valores de tensión todos dentro de un rango determinado. Por ello, se decidió el empleo de un potenciómetro manual como medio de ajuste del voltaje de salida del dispositivo.

Con el fin de evitar inestabilidades y oscilaciones en el valor de la tensión que pueden surgir al usar directamente la tensión del potenciómetro, se decidió emplear el potenciómetro manual como sistema de control del operador y un potenciómetro digital para suministrar la señal de voltaje al cabezal del láser, consiguiendo de esta forma mayor precisión y estabilidad en el valor suministrado.

Para poder controlar el potenciómetro digital se incorpora al diseño la tarjeta Arduino



## 4.1. EVOLUCIÓN DEL HARDWARE



Uno, que además se empleará en el diseño final como elemento de control del resto de funciones del dispositivo de control, figura 4.1.

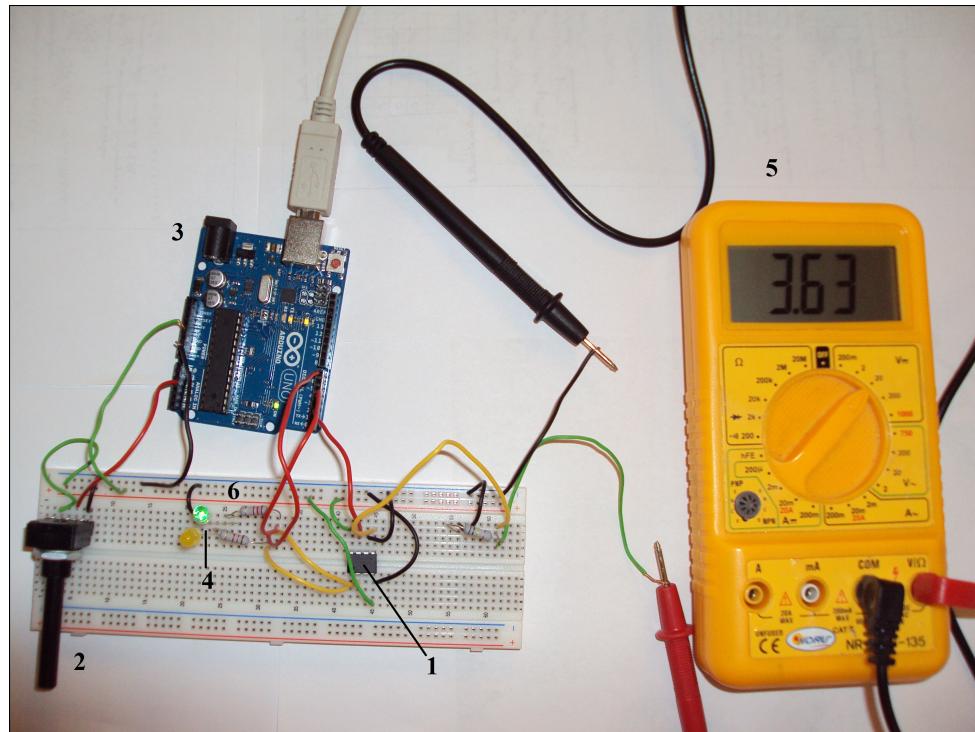


Figura 4.1: Primer prototipo del dispositivo de control. 1. Potenciómetro digital DS1804, 2. Potenciómetro manual, 3. Arduino Uno, 4. Leds, 5. Polímetro, 6. Resistencias.

Después de lograr con el diseño anterior que los valores de tensión se correspondan con los valores esperados, se determinó el valor de tensión que corresponde a cada una de las posiciones del potenciómetro digital, esto es, se realizó una calibración voltaje - posiciones. En el siguiente paso del diseño, se ideó el modo de que el usuario viera los cambios en el voltaje del dispositivo. Para esto se pensó en varias alternativas, como el uso del medidor de tensión compacto DPM 40. Esta opción se descartó por su falta de flexibilidad, ya que este aparato sólo permitiría indicar la tensión de salida, sin cubrir el resto de funciones. Por ello, para poder cubrir el mayor número de necesidades posibles, finalmente se optó por la instalación de una pantalla LCD, ver figura 4.2.

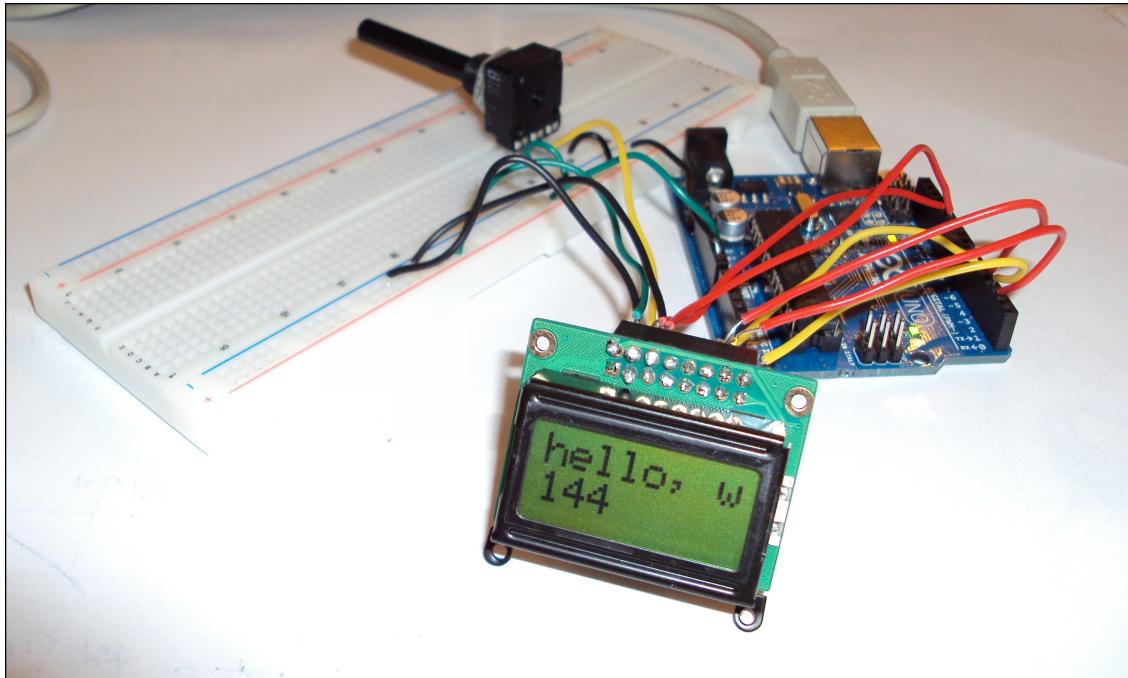


Figura 4.2: Prototipo con la LCD añadida.

Tras diseñar el circuito para la utilización de la pantalla LCD e integrarlo con el circuito para el funcionamiento del potenciómetro digital, seguían quedando algunos pines de conexión al láser por el puerto RS232 que facilitan el control del láser a través del dispositivo que se desarrolló. Dos de estos pines, el 4 y el 5 del puerto RS232 del cabezal del láser, corresponden a salidas de datos del orden de los milivoltios, proporcionales a la corriente y potencia emitidas por el láser. Dado que la tarjeta Arduino lee señales del orden de los voltios, es necesario amplificar estas señales hasta este nivel para su recepción y uso por parte de la tarjeta.

En la realización de este circuito de amplificación primeramente se pensó en el uso de amplificadores operacionales, pero los resultados obtenidos no eran los esperados ya que aunque la amplificación obtenida si era del orden deseado, la variabilidad de la tensión obtenida era demasiado pequeña. Debido a que los amplificadores no cumplían con las expectativas, se emplearon los transistores C546C en un circuito de base común, como muestra el circuito de la figura 2.13 ya comentado.

Después de integrar este circuito al circuito general, se instalaron otros dos dispositivos luminosos (leds), con sus correspondientes resistencias, para señalar los pines 1 y 3 del conector RS232, que informan cuando la cabeza del láser está a la temperatura y potencia correctas para ser utilizado.

Una vez fijado el diseño de todos los circuitos necesarios para el funcionamiento de las distintas partes del dispositivo, se procedió a su integración en un solo circuito. Es con éste con el que se diseña la tarjeta PCB, quedando así como último paso la instalación de todos los componentes en una caja metálica que da como resultado el dispositivo final, figura 4.3.

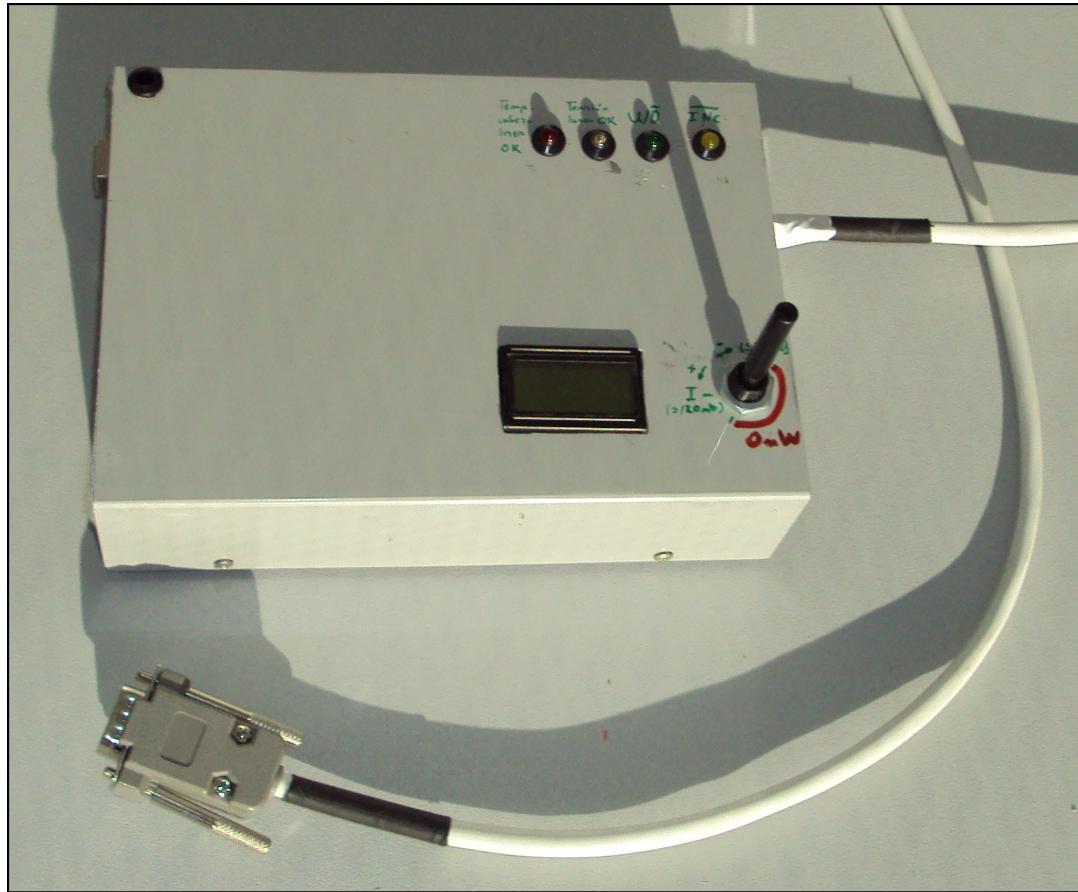


Figura 4.3: Dispositivo final montado y listo para su uso.

## 4.2. Evolución del software

El Software programado para el correcto funcionamiento del dispositivo es el resultado de haber ido modificando y ampliando unas pocas líneas escritas inicialmente con el objetivo de aprender y entender el funcionamiento del lenguaje de programación de la tarjeta Arduino.

Lo primero que se programó fué el código para manipular las posiciones del potenciómetro digital, para lo cual se procedió al desarrollo de varios programas encargados de controlar cada uno de los pines del potenciómetro digital. Una vez comprobado que estos programas compilaban y funcionaban correctamente, se procedió a su unificación y a la introducción de las modificaciones necesarias para el control más efectivo posible del potenciómetro digital.

Este primer programa fue la base sobre la que se amplió y modificó el código para adaptarlo a las necesidades que surgieron a medida que se introdujeron los diversos elementos de hardware en el diseño.

En todo momento el desarrollo del software iba ligado al desarrollo del hardware, dependiendo cada uno del otro, es decir, no se introdujeron nuevos elementos de hardware sin antes haber realizado un código que garantizara el correcto funcionamiento del último



## 4.2. EVOLUCIÓN DEL SOFTWARE



componente instalado, por lo que tampoco se avanzó en la programación del software, sin saber cuál iba a ser el elemento físico a instalar.

En el anexo I, se pueden observar algunas de las versiones iniciales del software, que debido al gran número de modificaciones al que han sido sometidas, poco tienen que ver con la versión final del programa.



# Capítulo 5

## Datos experimentales

En este capítulo se recogen todos los datos experimentales obtenidos y utilizados durante la realización del proyecto y se explica el método empleado para su obtención. Tanto para la calibración del dispositivo, como la utilización del sistema completo, dispositivo regulador de tensión y láser, fue necesaria la obtención de algunos datos experimentales. En la primera parte del capítulo se comenta la obtención de datos para la calibración de la tensión, mientras que en la segunda se explica y explicita la relación entre tensión proporcionada por el dispositivo y la intensidad obtenida por el láser. El capítulo finaliza con los errores experimentales.

### 5.1. Calibración de la tensión

En este ensayo, se lleva a cabo una lectura de la tensión que se obtiene según las distintas posiciones del potenciómetro digital. Para ello se cargó en la tarjeta Arduino el programa auxiliar del anexo H y se empleó el circuito esquematizado en la figura 5.1 mientras que su implementación real es la figura 5.2.

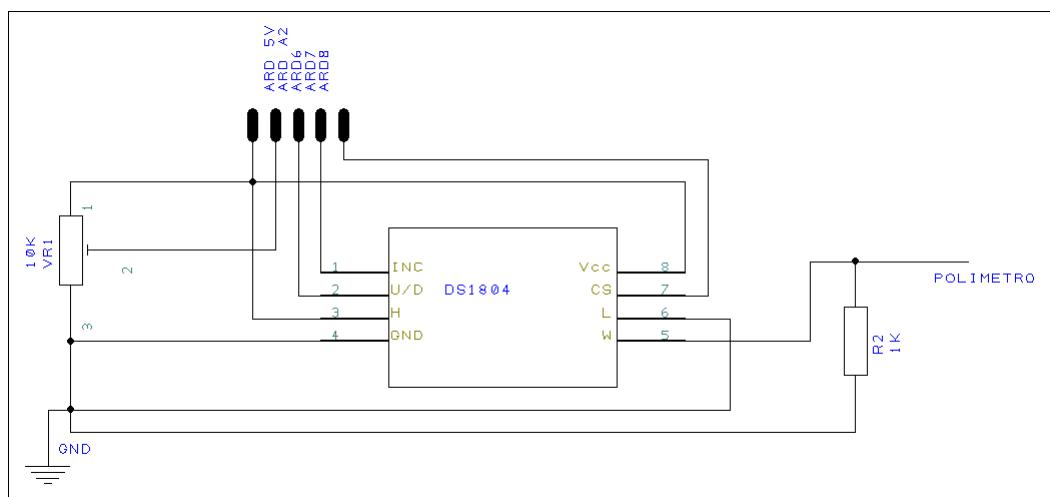


Figura 5.1: Circuito para la calibración de la posición del potenciómetro digital según el voltaje de salida.



## 5.1. CALIBRACIÓN DE LA TENSIÓN



Se han realizado medidas de tensión con un polímetro para cada una de las 99 posiciones de las que dispone el potenciómetro digital, graficándolas y aproximando estos valores por una recta, ecuación del cuadro 5.2 de la página 50, lo que permite obtener la relación de conversión entre las posiciones y el valor de tensión correspondiente a cada una.

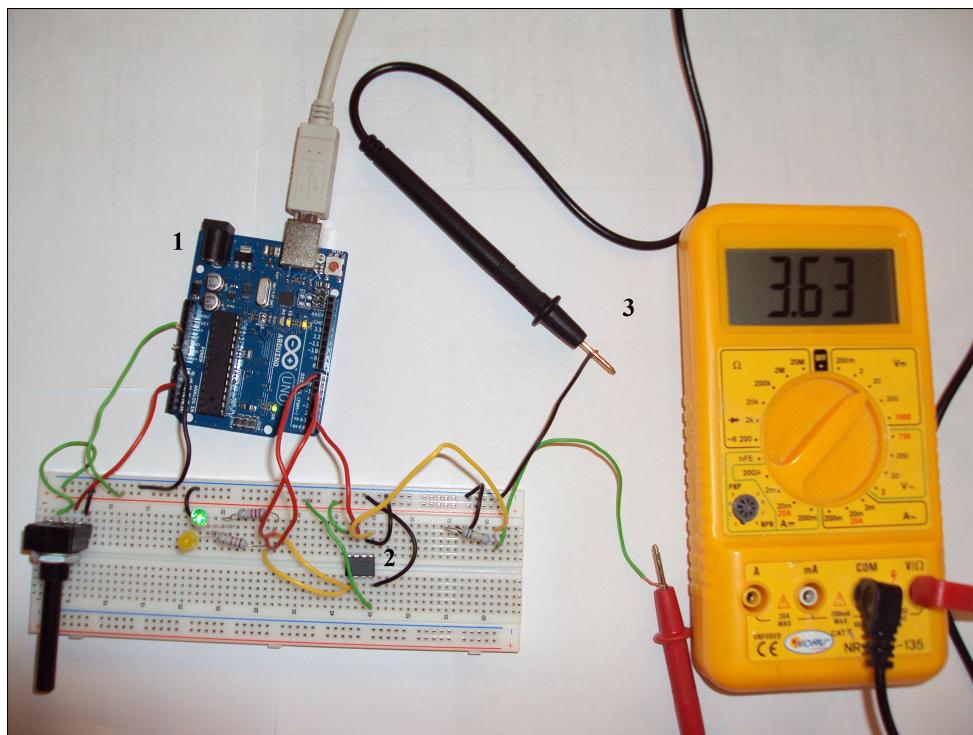


Figura 5.2: Montaje real. 1. Ardiuno, 2. Potenciómetro digital, 3. Polímetro enganchado al circuito.

Para poder observar el error sistemático cometido, se realizaron tres medidas iguales, lo que demostró que el error máximo cometido es 0,001 Voltios, o sea un error sistemático máximo inferior al 1 %.

Para realizar la calibración solo se tuvieron en cuenta los valores de posiciones en el potenciómetro superiores o iguales a 60, por razones ya comentadas en capítulos anteriores y que además se pueden observar en la tabla 5.1.

Este hecho implica que la precisión de la conversión es mayor, dado que se desprecian los valores que más pueden aumentar el error; aquellos que implican que el voltaje de salida es nulo.



Posición	Voltage (V)	Posición	Voltage (V)
60	2,48	80	3,54
61	2,52	81	3,61
62	2,57	82	3,67
63	2,62	83	3,74
64	2,66	84	3,81
65	2,71	85	3,88
66	2,76	86	3,95
67	2,81	87	4,03
68	2,86	88	4,1
69	2,91	89	4,18
70	2,97	90	4,26
71	3,02	91	4,34
72	3,07	92	4,43
73	3,13	93	4,51
74	3,18	94	4,6
75	3,24	95	4,69
76	3,3	96	4,79
77	3,36	97	4,88
78	3,42	98	4,98
79	3,48	99	4,99

Cuadro 5.1: Posición en el potenciómetro y voltaje de salida.

Los valores de esta tabla se han tomado teniendo como fuente de alimentación del circuito un ordenador a través de un cable USB y una fuente de alimentación externa de 7V. En ambos casos, los valores obtenidos fueron semejantes, lo que implica que el circuito está correctamente diseñado y trabaja con varios valores de potencia, siempre y cuando estén dentro de los límites nominales del diseño. Mediante el uso del programa QtiPlot [15], se obtiene la gráfica mostrada en la figura 5.3, que es una curva exponencial. Por ese motivo es la aproximación más exacta, que mejor se ajusta a los datos experimentales. Así, esto se puede confirmar comparando los valores de la  $R^2$  para ambas aproximaciones, cuadro 5.2.

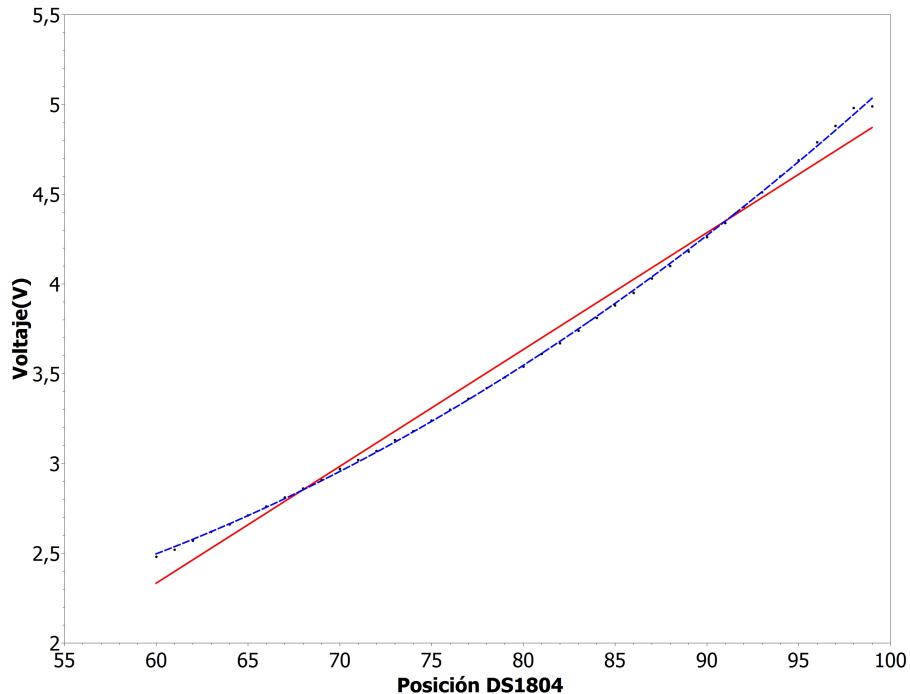


Figura 5.3: Voltaje de salida respecto la posición en el potenciómetro digital (Línea roja, aproximación lineal, línea rayada azul, exponencial).

Ecuación	$R^2$
$y = 0,065x - 1,572$	0,988
$y = 0,829 e^{(0,018x)}$	0,999

Cuadro 5.2: Ecuaciones aproximadas; lineal y exponencial.

El valor más próximo de la  $R^2$  a la unidad es el de la aproximación exponencial, lo que indica que esta aproximación será mas exacta, aunque el bajo valor del exponente (0,018) sugiere que una aproximación lineal tampoco es muy incorrecta, como confirma el error de la aproximación lineal de 0,014 V.

## 5.2. Medición de la intensidad lumínica del láser

Para este ensayo se ha utilizado una cabeza sensora y un medidor de intensidad calibrado a la longitud de onda del láser, lo que proporciona un valor de intensidad ya calibrado.

En el montaje experimental empleado, figura 5.4, el haz procedente del láser, antes de llegar a la cabeza sensora, pasa por un expansor para poder aumentar de manera notable la superficie que cubre, a costa de perder intensidad de luz.

Para realizar el ensayo se conectó el modificador de tensión objeto de este proyecto



## 5.2. MEDICIÓN DE LA INTENSIDAD LUMÍNICA DEL LÁSER



a la unidad encargada de suministrar potencia a la cabeza láser. Una vez que está en funcionamiento, y a través del expansor, el haz láser llega a la cabeza sensora. Esta última está conectada al medidor de intensidad y muestra en una pantalla la intensidad medida.

En la figura 5.4 se puede observar el sistema completo.

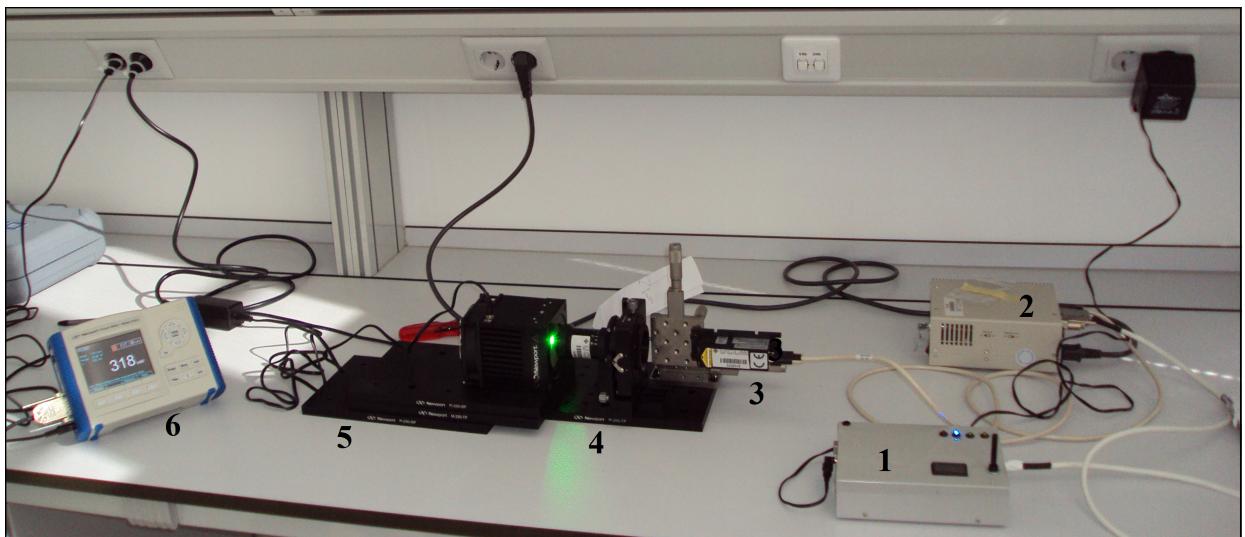


Figura 5.4: Sistema para la medida de la intensidad. 1. Controlador de tensión, 2. Alimentación del laser, 3. Láser, 4. Expansor, 5. Cabeza sensora, 6. Medidor de intensidad.

En el ensayo, a medida que se modifica la tensión suministrada al láser a través del dispositivo diseñado en este proyecto, se observa el cambio de la intensidad lumínica del láser, cumpliendo así el principal objetivo del instrumento diseñado.

Para obtener los datos presentados en la tabla 5.3 se han realizado dos barridos del voltaje comenzando por el máximo y bajando hasta el valor mínimo de 2,5 V. Para cada valor del voltaje se tomaron cinco medidas de la intensidad, como muestran las tablas del anexo J. Como a una tensión de 2,44 V el láser no emite ninguna luz, en este trabajo se consideró la intensidad medida a este valor, 186 y 204 mW según la tanda de medidas, como la intensidad debida a la luz ambiental y se ha restado de los valores medidos para obtener finalmente la tabla 5.3, con los valores de la intensidad ya promediados y el valor de la intensidad ambiente ya restado.

Primera medida		Segunda medida	
Tensión(V)	Intensidad lumínica(mW)	Tensión(V)	Intensidad lumínica(mW)
2,44	0	2,44	0
2,65	10	2,66	8
3,01	4	2,88	14
3,12	0	3,1	10
3,26	12	3,37	0
3,46	12	3,56	6



3,68	12	3,88	6
3,82	14	4,15	54
3,96	48	4,31	72
4,08	82	4,58	82
4,29	84	4,69	96
4,59	108	4,78	100
4,73	122	4,93	112
4,93	142		

Cuadro 5.3: Tensión frente a intensidad luminosa de las dos medidas.

Observando la tabla 5.3 se deduce que no se hay una intensidad lumínica que supere el fondo hasta alcanzados aproximadamente los 3,9 V de tensión en el controlador, lo que contradice las hojas de características del láser, que detallaban un valor de 2,44 V para que el láser comenzara a emitir luz. Este comportamiento puede observarse más claramente en la gráfica de la figura 5.5 de la intensidad del láser frente al valor de tensión del controlador.

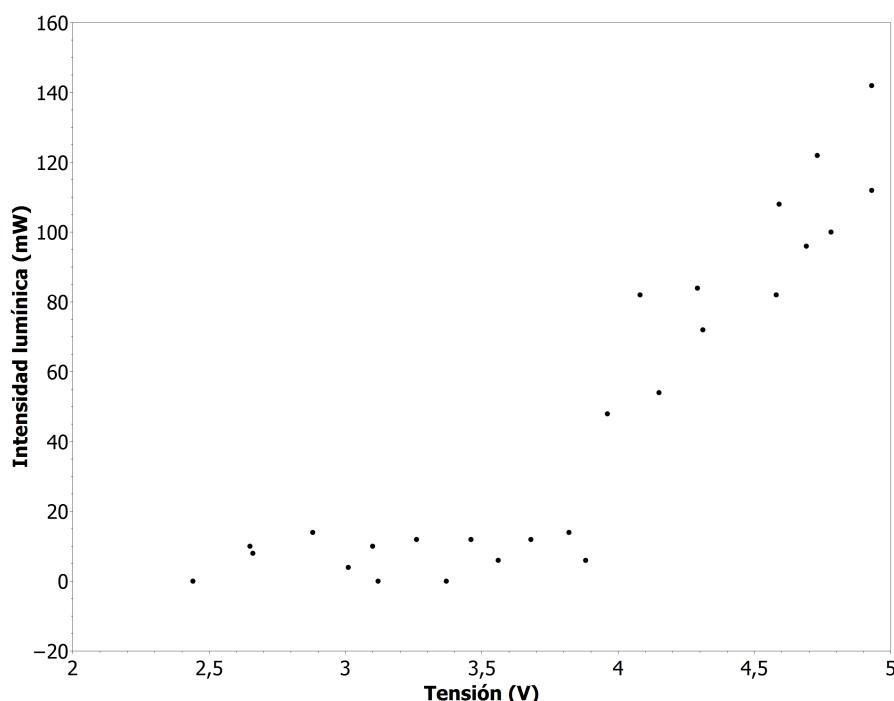


Figura 5.5: Intensidad lumínica del láser frente a voltaje del controlador.

Así, el láser no empieza prácticamente a emitir luz hasta los 4V de tensión, valor a partir del cual la intensidad lumínica aumenta casi linealmente con el voltaje suministrado. Por ello, el ajuste de los datos experimentales comenzó con los valores de voltaje de 3,9



V, figura 5.6. Al igual que en el caso anterior se han intentado realizar dos ajustes, uno lineal y otro exponencial. Como la tabla 5.4 muestra, el mejor ajuste corresponde a la función exponencial, si bien por un margen estrecho. De todas formas, es esta la función implementada en el proyecto, lo que implica un error en la medida de la intensidad del láser de 17 mW.

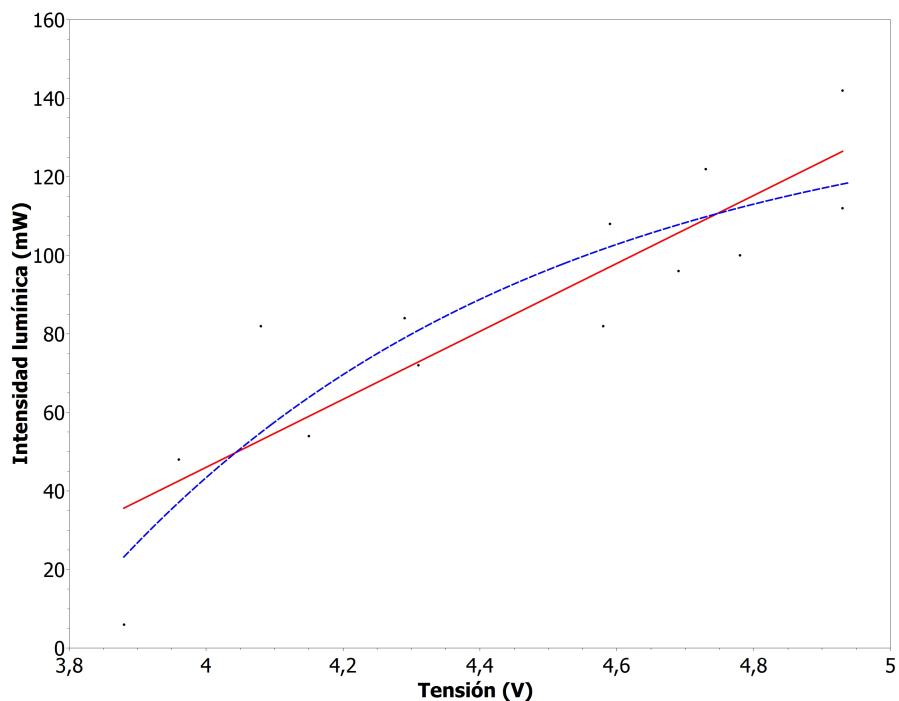


Figura 5.6: Intensidad lumínica del láser respecto al voltaje de salida del controlador (Línea roja, aproximación lineal, línea rayada azul, exponencial).

Función	$R^2$
$y = 0,789x - 300$	0,789
$y = 140 - 51391 e^{(-x/0,638)}$	0,812

Cuadro 5.4: Ecuaciones aproximadas de la intensidad lumínica; lineal y exponencial.

Además, durante la realización del ensayo se observó también el funcionamiento de los indicadores luminosos, determinando su correcto funcionamiento y por otra parte, como en todos los ensayos experimentales, en los realizados durante este proyecto también existen errores de medición, cuya magnitud final se puede calcular del siguiente modo:

- Voltaje suministrado por el controlador:

En este proyecto se han considerado dos fuentes de error: Primero, la propia tolerancia del voltímetro, cifrada en  $\pm 0,001$  V en sus especificaciones, y el error producido



## 5.2. MEDICIÓN DE LA INTENSIDAD LUMÍNICA DEL LÁSER



por el ajuste de la recta de las posiciones del potenciómetro al voltaje de salida, de valor  $\pm 0,014$  V. Por lo tanto el error cometido en el voltaje es de  $\pm 0,02$  V.

- Calibración de la intensidad lumínica:

Las fuentes de error consideradas son las mismas que en el caso anterior, pero con valores de  $\pm 2$  mW y  $\pm 17$ mW para el error de medición y el de la calibración de la intensidad respecto el voltaje del controlador, respectivamente. El error total en la intensidad es entonces, de 19 mW.

# Capítulo 6

## Conclusiones

Al finalizar este proyecto, se dispone de una aplicación que permite controlar de una manera muy intuitiva y sencilla la intensidad del láser Excelsior - 532 - 150.

Para poder cumplir con los objetivos el proyecto se ha realizado siguiendo el siguiente esquema de trabajo.

- Estudio del laser, así como de las distintas partes del mismo y modos de funcionamiento.
- Diseñar y describir las especificaciones funcionales del sistema, es decir, concretar las funciones a implementar en el proyecto.
- Desarrollo de la parte principal del proyecto (variación de la tensión suministrada de manera estable), con su correspondiente realización de pruebas, asegurando que funciona correctamente y éste realmente puede modificar la tensión suministrada al laser.
- Desarrollo de otras funciones secundarias, también con sus correspondientes pruebas y asegurando su correcto funcionamiento.

Las funciones secundarias desarrolladas son.

- Indicadores Led del correcto funcionamiento del laser (temperatura y potencia).
  - Indicadores Led del modo de funcionamiento del laser.
  - Desarrollo del hardware de lectura de la corriente y potencia de funcionamiento del láser.
- Elaboración final de todos los circuitos y componentes desarrollados y comprobación de que todo funciona correctamente.
  - Pruebas de la intensidad lumínica del laser a distintos niveles de potencia suministrada e implementación de esta información en el software desarrollado durante la realización del proyecto.



---

Como en cualquier proyecto o aplicación comercial, en esta aplicación hay elementos que no han podido ser introducidos en la primera versión. Algunos de los elementos que podrán introducirse en trabajos posteriores son:

1. Introducción de un teclado que permita introducir la tensión a suministrar al láser o bien la potencia con la que el láser tiene que funcionar.
2. Poner un botón de emergencia que en caso de un mal funcionamiento o riesgo de producir algún tipo de accidente, se pueda detener de manera automática el laser desde la unidad desarrollada.
3. Conexión y desarrollo de software de los siguientes pines del puerto RS232 del láser:
  - PIN2. Entrada. Permite encender el laser inmediatamente si 5V están disponibles en la cabeza del laser.
  - PIN10. Salida. Servicio de alarma. Avisa de que la cabeza laser está llegando al final de su vida de funcionamiento.
  - El resto de pines (pines 9, 13, 14 y 15) que no han sido desarrollados en el proyecto no se indican como posible mejora, debido a que en el manual del láser se indica que deben de estar abiertos.
4. Desarrollo a nivel de software de los pines ya instalados 4 y 5.

Para valorar el impacto que este trabajo puede tener, resaltar que mediante pequeñas modificaciones se puede conseguir que el sistema de control desarrollado durante el proyecto, pueda también servir para el control de otros instrumentos cuyos modos de funcionamiento dependan de la tensión con la que son alimentados.

Como conclusión personal he de resaltar que la realización de este proyecto me ha ayudado a adquirir nuevos conocimientos y a tener una nueva visión sobre el funcionamiento de los láseres y las utilidades que éstos pueden proporcionar. Del mismo modo, este proyecto me sirvió para darme cuenta de que, a pesar de adentrarme en terrenos desconocidos en muchas partes de la realización del mismo, con paciencia y constancia se puede conseguir un trabajo muy satisfactorio.

# Bibliografía

- [1] Boylestad, Robert L. y Nashelsky, Louis. *Electrónica: Teoría de circuitos y Dispositivos Electronicos. Capítulo tres: Transistores de unión bipolar..* Pearson education, México, 2009.
- [2] Francisco Ruiz Vassallo. *Enciclopedia Básica de Electrónica. Circuitos básicos: Electrónica analógica, electrónica digital y Etapas y Conexiones. Capítulo 6: Polarización de transistores.* Ceac
- [3] High temperature exploding wires (F. D. Bennet). *Progress in high temperature physics and chemistry (1968).*
- [4] A. G. Rousskikh. *Expansion of the plasma corona from a wire exploded in vacuum* Physics of plasmas, Vol. 17, 033505 (2010).
- [5] Marfil Reguero, Daniel, 2007. *Diseño e implementación de un cirucito eléctrico inteligente para la emisión de sonido para control de plagas.*
- [6] Manual del usuario. *EXCELSIOR, Diode-Pumped, Visible CW Laser.*
- [7] [http://personales.unican.es/perezvr/pdf/CH7ST\\_Web.pdf](http://personales.unican.es/perezvr/pdf/CH7ST_Web.pdf) [Modulación de pulsos]
- [8] <http://arduino.cc> [Página web oficial de las tarjetas Arduino]
- [9] <http://www.neoteo.com/designspark-pcb-facil-gratis> [Manual para el programa DesignSpark]
- [10] [http://dfists.ua.es/jpomares/arduino/page\\_06.htm](http://dfists.ua.es/jpomares/arduino/page_06.htm) [Manual de programación Arduino]
- [11] <http://www.sorayapaniagua.com/2011/03/14/arduino-la-revolucion-silenciosa-del-hardware-libre/> [Definición del lenguaje empleado para la programación.]
- [12] <http://www.analfatecnicos.net/archivos/15.SistemasModulacionWikipedia.pdf> [Sistemas de modulación]
- [13] <http://arduino.cc/es/Reference/LiquidCrystal> [Software para la pantalla LCD]



- 
- [14] <http://www.ucontrol.com.ar/forosmf/tutoriales-guias-y-cursos-en-ucontrol/diseno-de-circuitos-impresos-con-kicad/?PHPSESSID=omt93d1qgd6l0fi8f6qo4fb5i4> [*Manual de uso del KiCAD*]
  - [15] <http://alternativeto.net/software/qtiplot/> [*Programa QtiPlot*]
  - [16] <http://todoelectronica.com/%C3%82%C2%BFpor-arduino-ventajas-sobre-otros-microcontroladores-p-13507.html> [*Ventajas de la tarjeta Arduino sobre otros microcontroladores*]
  - [17] <http://www.arduino.projects.com/node/10> [*Control de la tarjeta Arduino a través de un USB*]
  - [18] [http://www.apetega.org/xor-outono08/documentacion/Linguaxe\\_Arduino.pdf](http://www.apetega.org/xor-outono08/documentacion/Linguaxe_Arduino.pdf) [*Lenguaje Arduino*]
  - [19] <http://www.sorayapanagua.com/2011/03/14/arduino-la-revolucion-silenciosa-del-hardware-libre/> [*Origen del lenguaje Arduino*]
  - [20] <http://freedomdefined.org/Licenses> [*Licencias de hardware*]

## **Anexo A**

### **Potenciómetro digital DS1804**



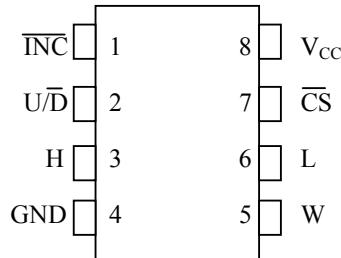
# DS1804 NV Trimmer Potentiometer

[www.maxim-ic.com](http://www.maxim-ic.com)

## FEATURES

- Single 100-position taper potentiometer
- Nonvolatile (NV) on-demand wiper storage
- Operates from 3V or 5V supplies
- Up/down, increment-controlled interface
- Available in 8-pin (300mil) DIP, 8-pin (150mil) SO, 8-pin (118mil) µSOP, and flip-chip packages
- Operating Temperature:
  - Industrial: -40°C to +85°C
- Standard Resistance Values:
  - DS1804-010 10kΩ
  - DS1804-050 50kΩ
  - DS1804-100 100kΩ

## PIN ASSIGNMENT



8-Pin DIP (300mil)

8-Pin SO (150mil)

8-Pin µSOP (118mil)

8-Pin Flip-Chip Package

See Mech. Drawings Section

## PIN DESCRIPTION

H	- High-Terminal of Potentiometer
L	- Low-Terminal of Potentiometer
W	- Wiper of Potentiometer
V <sub>CC</sub>	- 3V or 5V Power Supply
CS	- Chip Select
U/D	- Up/Down Control
INC	- Increment/Decrement Wiper Control
GND	- Ground

## DESCRIPTION

The DS1804 NV trimmer potentiometer is a nonvolatile digital potentiometer that has 100 positions. The device provides an ideal method for low-cost trimming applications using a CPU or manual control input with minimal external circuitry. Wiper position of the DS1804 can be stored in EEPROM memory on demand. The device's wiper position is manipulated by a three-terminal port that provides an increment/decrement counter controlled interface. This port consists of the control inputs CS, INC, and U/D. The DS1804 is available in three resistor grades, which include a 10kΩ, 50kΩ, and 100kΩ. The device is provided in an industrial temperature grade. Additionally, the DS1804 will operate from 3V or 5V supplies and is ideal for portable application requirements. Three packaging options are available and include the 8-pin (300mil) DIP, 8-pin (150mil) SO, 8-pin (118mil) µSOP, and the flip-chip package.

## OPERATION

The DS1804 is a single nonvolatile potentiometer. The device has a total of 100 tap-points including the L- and H- terminals. A total of 99 resistive segments exist between the L- and H- terminals. These tap-points are accessible to the W-terminal, whose position is controlled via a three-terminal control port. A block diagram of the DS1804 is shown in Figure 1.

The three-terminal port of the DS1804 provides an increment/decrement interface which is activated via a chip-select input. This interface consists of the input signals  $\overline{CS}$ ,  $\overline{INC}$ , and  $U/D$ . These input signals control a 7-bit up/down counter. The output of the 7-bit up/down counter controls a 1 of 100 decoder to select wiper position. Additionally, this interface provides for a wiper storage operation using the  $\overline{CS}$  and  $\overline{INC}$  input control pins. The timing diagram for the three-terminal interface control is shown in Figure 2.

## PIN DESCRIPTIONS

**V<sub>CC</sub>** - Power Supply. The DS1804 will support supply voltages ranging from +2.7V to +5.5V.

**GND** - Ground.

**H** - High-Terminal Potentiometer. This is the high terminal of the potentiometer. It is not required that this terminal be connected to a potential greater than the L-terminal. Voltage applied to the H-terminal cannot exceed the power-supply voltage, V<sub>CC</sub>, or go below ground.

**L** - Low-Terminal Potentiometer. This is the low terminal of the potentiometer. It is not required that this terminal be connected to a potential less than the H-terminal. Voltage applied to the L-terminal cannot exceed the power-supply voltage, V<sub>CC</sub>, or go below ground.

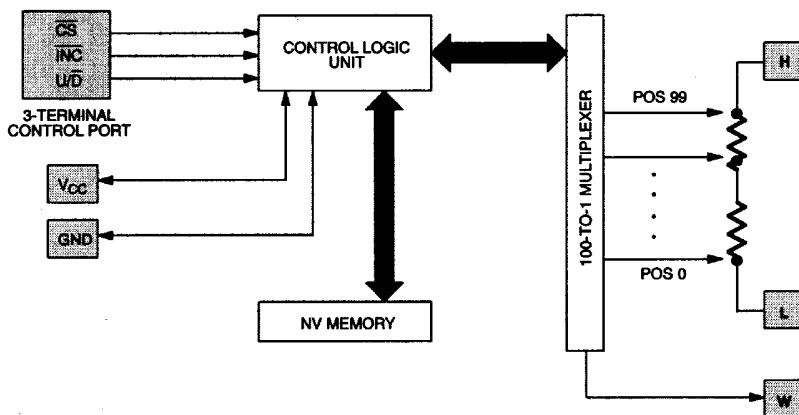
**W** - Wiper of the Potentiometer. This pin is the wiper of the potentiometer. Its position on the resistor array is controlled by the three-terminal control port. Voltage applied to the wiper cannot exceed the power-supply voltage, V<sub>CC</sub>, or go below ground.

**$\overline{CS}$**  - Chip Select. The  $\overline{CS}$  input is used to activate the control port of the DS1804. This input is active low. When in a high-state, activity on the  $\overline{INC}$  and  $U/D$  port pins will not affect or change wiper position.

**$\overline{INC}$**  - Wiper Movement Control. This input provides for wiper position changes when the  $\overline{CS}$  pin is low. Wiper position changes of the W-terminal will occur one position per high-to-low transition of this input signal. Position changes will not occur if the  $\overline{CS}$  pin is in a high state.

**$U/D$**  - Up/Down Control. This input sets the direction of wiper movement. When in a high state and  $\overline{CS}$  is low, any high-to-low transition on  $\overline{INC}$  will cause a one position movement of the wiper towards the H-terminal. When in a low state and  $\overline{CS}$  is low, any high-to-low transitions on  $\overline{INC}$  will cause the position of the wiper to move towards the L-terminal.

## DS1804 BLOCK DIAGRAM Figure 1



### POWER-UP/POWER-DOWN CONDITIONS

On power-up, the DS1804 will load the value of EEPROM memory into the wiper position register (or 1 of 100 decoder). The value of this register can then be set to another wiper position if desired, by using the three-terminal control port. On power-up, wiper position will be loaded within a maximum time period of 500µs once the power-supply is stable. Additionally, the three-terminal interface port will be active after 50ms.

On power-down, the wiper position register data will be lost. On the next device power-up, the value of EEPROM memory will be loaded into the wiper position register.

On shipment from the factory, Dallas Semiconductor does not guarantee a specified EEPROM memory value. This value should be set by the customer as needed.

### NONVOLATILE WIPER STORAGE

Wiper position of the DS1804 can be stored using the **INC** and **CS** inputs. Storage of the wiper position takes place whenever the **CS** input transitions from low-to-high while the **INC** is high. Once this condition has occurred the value of the current wiper position will be written to EEPROM memory.

The DS1804 is specified to accept 50,000 writes to EEPROM before a wear-out condition. After wear-out the DS1804 will still function and wiper position can be changed during powered conditions using the three-terminal control port. However, on power-up the wiper-position will be indeterminate.

### ONE-TIME PROGRAMMABILITY (OTP)

The DS1804 can be easily used as an OTP device. The user of the DS1804 can trim the desired value of the wiper position and set this position for storage as described above. Any activity through the three-terminal port can then be prevented by connecting the **CS** input pin to **V<sub>CC</sub>**. Also, an OTP application does not adversely affect the number of times EEPROM is written, since EEPROM will only be loaded and *not written* during a power-up or power-down condition.

On power-up, the DS1804 will load the current value of EEPROM memory into the wiper position register.

**ABSOLUTE MAXIMUM RATINGS\***

Voltage on Any Pin Relative to Ground	-1.0V to +7.0V
Operating Temperature Range	-40°C to +85°C
Storage Temperature Range	-55°C to +125°C
Soldering Temperature	See IPC/JEDEC J-STD-020A Specification

\* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

## 1) Insert A (Burn-in Disclaimer)

The Dallas Semiconductor DS1804 is built to the highest quality standards and manufactured for long term reliability. All Dallas Semiconductor devices are made using the same quality materials and manufacturing methods. However, the flip-chip package version of the DS1804 is not exposed to environmental stresses, such as burn-in, that some industrial applications require. For specific reliability information on this product, please contact the factory in Dallas at 972-371-4448.

**RECOMMENDED DC OPERATING CONDITIONS**

(-40°C to +85°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V <sub>CC</sub>	+2.7		5.5	V	1
Input Logic 1	V <sub>IH</sub>	2.0		V <sub>CC</sub> + 0.5	V	1,2
Input Logic 0	V <sub>IL</sub>	-0.5		+0.8 +0.6	V	1,15
Resistor Inputs	L,H,W	GND - 0.5		V <sub>CC</sub> + 0.5	V	1,3

**DC ELECTRICAL CHARACTERISTICS**(-40°C to +85°C; V<sub>CC</sub> = 2.7V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Current	I <sub>CC</sub>			400	µA	4
Input Leakage	I <sub>LI</sub>	-1		+1	µA	
Wiper Resistance	R <sub>W</sub>		400	1000	Ω	
Wiper Current	I <sub>W</sub>			1	mA	
Standby Current: 3 Volts 5 Volts	I <sub>STBY</sub>		10 20	48	µA µA	5
Wiper Load Time	t <sub>WLT</sub>		500		µs	6
Power-Up Time	t <sub>PU</sub>		50		ms	14

**ANALOG RESISTOR CHARACTERISTICS** (-40°C to +85°C; V<sub>CC</sub>= 2.7V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
End-to-End Resistor Tolerance		-20		+20	%	8
Absolute Linearity			±0.6		LSB	9
Relative Linearity			±0.25		LSB	10
-3dB Cutoff Frequency	f <sub>cutoff</sub>				MHz	11
Temperature Coefficient			750		ppm/°C	

**CAPACITANCE** (25°C; V<sub>CC</sub>= 2.7V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Capacitance	C <sub>IN</sub>			5	pF	12
Output Capacitance	C <sub>OUT</sub>			7	pF	12

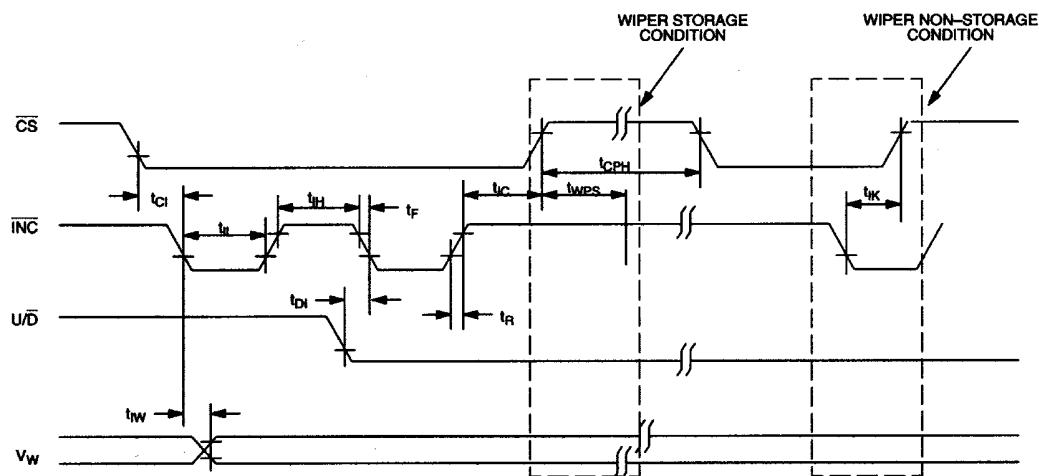
**AC ELECTRICAL CHARACTERISTICS** (-40°C to +85°C; V<sub>CC</sub>= 2.7V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
CS to INC Setup	t <sub>CI</sub>	50			ns	13
U/D to INC Setup	t <sub>DI</sub>	100			ns	13
INC Low Period	t <sub>IL</sub>	50			ns	13
INC High Period	t <sub>IH</sub>	100			ns	13
INC inactive to CS Inactive	t <sub>IC</sub>	500			ns	13
CS Deselect Time	t <sub>CPH</sub>	100			ns	13
Wiper Change to INC Low	t <sub>IW</sub>			200	ns	13
INC Rise and Fall Times	t <sub>R</sub> , t <sub>F</sub>			500	μs	13
INC Low to CS Inactive	t <sub>IK</sub>	50			ns	16
Wiper Storage Time	t <sub>WST</sub>			10	ms	13, 17

**NOTES:**

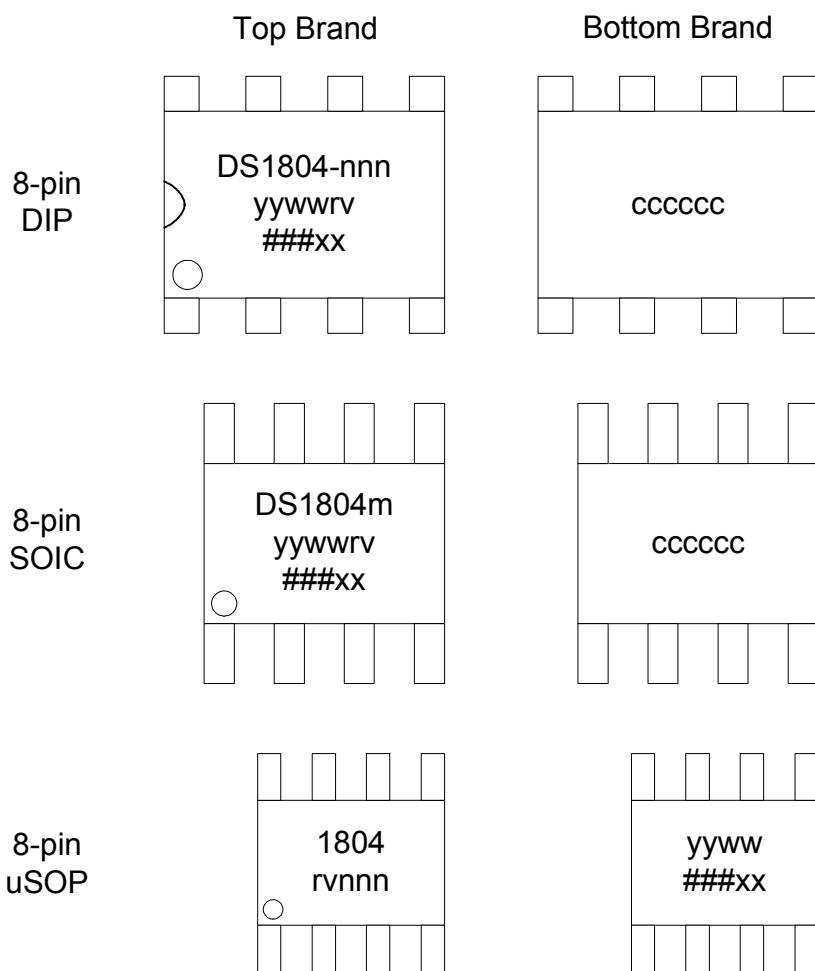
- 1) All voltages are referenced to ground.
- 2) Valid for  $V_{CC} = 5V$  only.
- 3) Resistor input voltages cannot go below ground or exceed  $V_{CC}$  by the amounts as shown in the table.
- 4) Maximum current specifications are based on the clock rate of  $\overline{INC}$  input. This specification represents the current required when changing the wiper position.
- 5) Standby current levels apply when all inputs are driven to appropriate supply levels.  $\overline{CS}$ ,  $\overline{INC}$ ,  $U/D = V_{CC}$ .
- 6) Wiper load time is specified as the time required for the DS1804 to load the wiper position with the contents of nonvolatile memory once  $V_{CC}$  has reached a stable operating voltage equal to or greater than 2.7V.
- 7) The DS1804 is available in three resistor values. These include the DS1804-010, 10k $\Omega$ ; the DS1804-050, 50k $\Omega$ ; and the DS1804-100, 100k $\Omega$ .
- 8) Valid at 25°C only.
- 9) Absolute linearity is used to compare measured wiper voltage versus expected wiper voltage as determined by wiper position. The DS1804 is specified to provide an absolute linearity of  $\pm 0.60$  LSB.
- 10) Relative linearity is used to determine the change in voltage between successive tap positions. The DS1804 is specified to provide a relative linearity specification of  $\pm 0.25$  LSB.
- 11) -3dB cutoff frequency characteristics for the DS1804 depend on potentiometer total resistance. DS1804-010, 1MHz; DS1804-050, 200 kHz; and DS1804-100, 100 kHz.
- 12) Capacitance values apply at 25°C.
- 13) See Figure 2.
- 14) Power-up time is specified as the time required before the three-terminal control becomes active after a stable power supply level has been reached.
- 15) At  $V_{CC} = 2.7V$ ,  $V_{IL} = 0.8V$ .
- 16) The  $\overline{INC}$  low to  $\overline{CS}$  inactive is specified to be 50ns minimum. This is the transition condition which allows the DS1804 three-terminal port to become inactive without writing the EEPROM memory of the part.
- 17) Wiper Storage Time,  $t_{WST}$ , is the time required for the DS1804 to write EEPROM memory for storage of a new wiper position. The maximum time required to accomplish this task is specified at 10ms.

## THREE-TERMINAL INTERFACE TIMING DIAGRAM Figure 2



## DS1804 ORDERING INFORMATION

ORDERING NUMBER	PACKAGE	OPERATING TEMPERATURE	VERSION
DS1804-010	8L DIP (300MIL)	-40°C TO +85°C	10kΩ
DS1804-050	8L DIP (300MIL)	-40°C TO +85°C	50kΩ
DS1804-100	8L DIP (300MIL)	-40°C TO +85°C	100kΩ
DS1804Z-010	8L SOIC (150MIL)	-40°C TO +85°C	10kΩ
DS1804Z-050	8L SOIC (150MIL)	-40°C TO +85°C	50kΩ
DS1804Z-100	8L SOIC (150MIL)	-40°C TO +85°C	100kΩ
DS1804U-010	8L µSOP (118MIL)	-40°C TO +85°C	10kΩ
DS1804U-050	8L µSOP (118MIL)	-40°C TO +85°C	50kΩ
DS1804U-100	8L µSOP (118MIL)	-40°C TO +85°C	100kΩ
DS1804X-010	8L FCP (118MIL)	-40°C TO +85°C	10kΩ
DS1804X-050	8L FCP (118MIL)	-40°C TO +85°C	50kΩ
DS1804X-100	8L FCP (118MIL)	-40°C TO +85°C	100kΩ

**BRANDING INFORMATION Figure 3**

DS1804 Brand Codes			
Code	Description	Values	
nnn	Pot Size (DIP/uSOP)	010	10k ohm
		050	50k ohm
		100	100k ohm
m	Pot Size (SOIC)	X	10k ohm
		L	50k ohm
		C	100k ohm
yyww	Date Code. It identifies the year and work week the device was assembled	eg. 0035 is the 35th work week of 2000	
rv	Die Revision	Generally this is a letter followed by a number (eg. B2)	
####xx	Lot Code		
cccccc	Country of Assembly		



## **Anexo B**

### **Transistor BC546**



## BC546/547/548/549/550

### Switching and Applications

- High Voltage: BC546,  $V_{CEO}=65V$
- Low Noise: BC549, BC550
- Complement to BC556 ... BC560



### NPN Epitaxial Silicon Transistor

#### Absolute Maximum Ratings $T_a=25^\circ C$ unless otherwise noted

Symbol	Parameter	Value	Units
$V_{CBO}$	Collector-Base Voltage : BC546	80	V
	: BC547/550	50	V
	: BC548/549	30	V
$V_{CEO}$	Collector-Emitter Voltage : BC546	65	V
	: BC547/550	45	V
	: BC548/549	30	V
$V_{EBO}$	Emitter-Base Voltage : BC546/547	6	V
	: BC548/549/550	5	V
$I_C$	Collector Current (DC)	100	mA
$P_C$	Collector Power Dissipation	500	mW
$T_J$	Junction Temperature	150	°C
$T_{STG}$	Storage Temperature	-65 ~ 150	°C

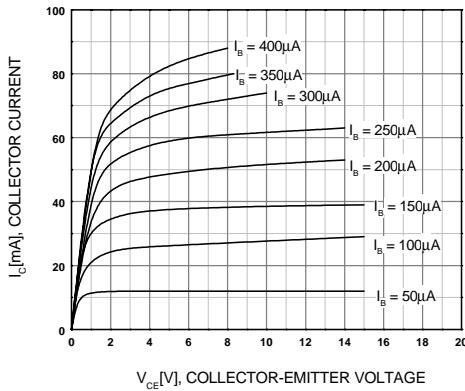
#### Electrical Characteristics $T_a=25^\circ C$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
$I_{CBO}$	Collector Cut-off Current	$V_{CB}=30V, I_E=0$			15	nA
$h_{FE}$	DC Current Gain	$V_{CE}=5V, I_C=2mA$	110		800	
$V_{CE}$ (sat)	Collector-Emitter Saturation Voltage	$I_C=10mA, I_B=0.5mA$ $I_C=100mA, I_B=5mA$		90 200	250 600	mV mV
$V_{BE}$ (sat)	Base-Emitter Saturation Voltage	$I_C=10mA, I_B=0.5mA$ $I_C=100mA, I_B=5mA$		700 900		mV mV
$V_{BE}$ (on)	Base-Emitter On Voltage	$V_{CE}=5V, I_C=2mA$ $V_{CE}=5V, I_C=10mA$	580	660	700 720	mV mV
$f_T$	Current Gain Bandwidth Product	$V_{CE}=5V, I_C=10mA, f=100MHz$		300		MHz
$C_{ob}$	Output Capacitance	$V_{CB}=10V, I_E=0, f=1MHz$		3.5	6	pF
$C_{ib}$	Input Capacitance	$V_{EB}=0.5V, I_C=0, f=1MHz$		9		pF
NF	Noise Figure : BC546/547/548	$V_{CE}=5V, I_C=200\mu A$		2	10	dB
	: BC549/550	$f=1KHz, R_G=2K\Omega$		1.2	4	dB
	: BC549	$V_{CE}=5V, I_C=200\mu A$		1.4	4	dB
	: BC550	$R_G=2K\Omega, f=30~15000MHz$		1.4	3	dB

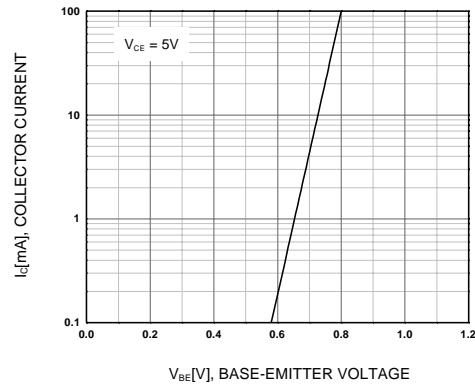
### $h_{FE}$ Classification

Classification	A	B	C
$h_{FE}$	110 ~ 220	200 ~ 450	420 ~ 800

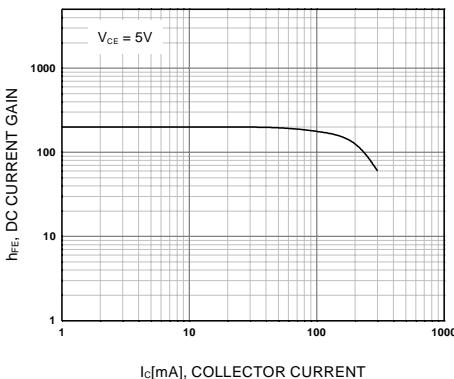
## Typical Characteristics



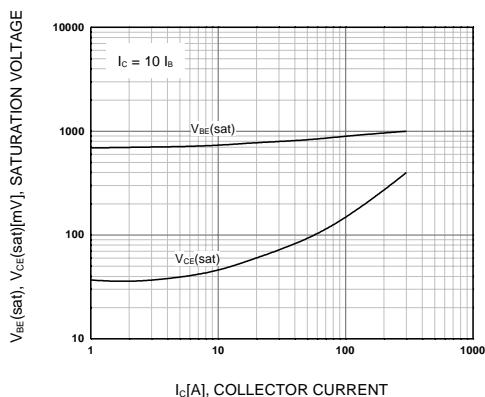
**Figure 1. Static Characteristic**



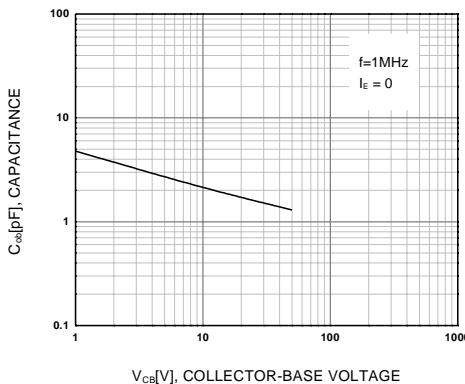
**Figure 2. Transfer Characteristic**



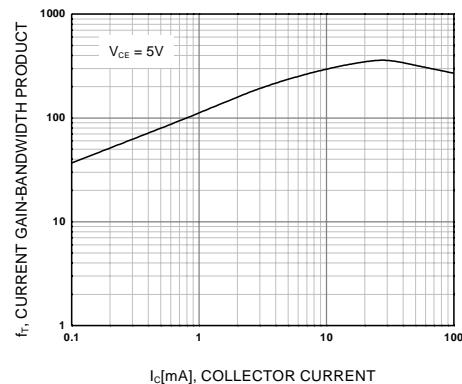
**Figure 3. DC current Gain**



**Figure 4. Base-Emitter Saturation Voltage  
Collector-Emitter Saturation Voltage**



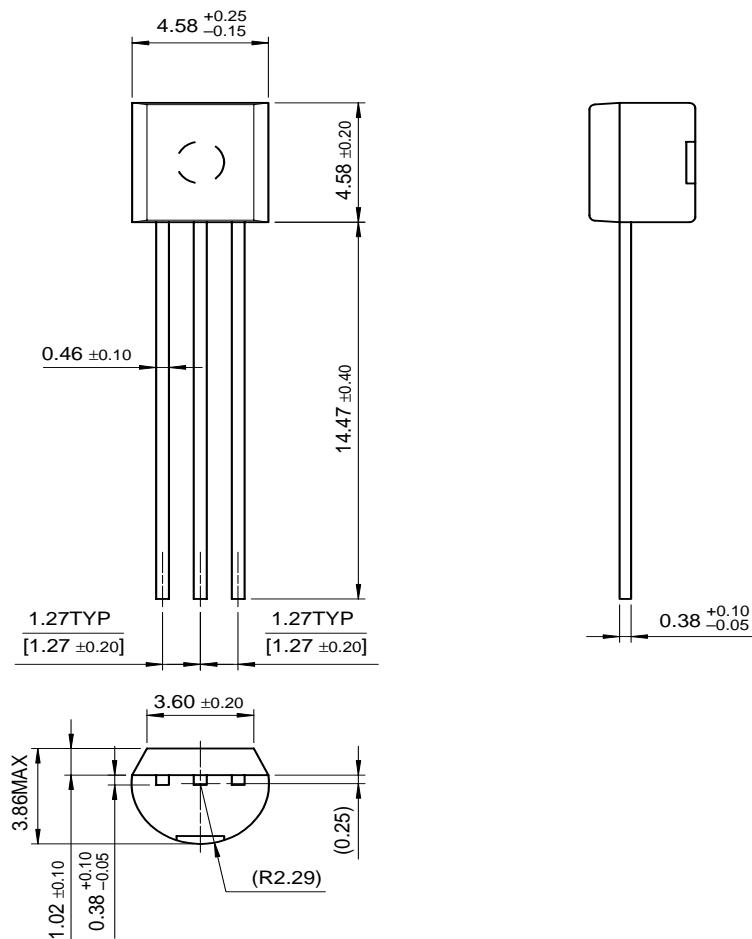
**Figure 5. Output Capacitance**



**Figure 6. Current Gain Bandwidth Product**

## Package Dimensions

TO-92



Dimensions in Millimeters

## TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACE <sup>TM</sup>	FACT <sup>TM</sup>	ImpliedDisconnect <sup>TM</sup>	PACMAN <sup>TM</sup>	SPM <sup>TM</sup>
ActiveArray <sup>TM</sup>	FACT Quiet series <sup>TM</sup>	ISOPLANAR <sup>TM</sup>	POP <sup>TM</sup>	Stealth <sup>TM</sup>
Bottomless <sup>TM</sup>	FAST <sup>®</sup>	LittleFET <sup>TM</sup>	Power247 <sup>TM</sup>	SuperSOT <sup>TM-3</sup>
CoolFET <sup>TM</sup>	FASTR <sup>TM</sup>	MicroFET <sup>TM</sup>	PowerTrench <sup>®</sup>	SuperSOT <sup>TM-6</sup>
CROSSVOLT <sup>TM</sup>	FRFET <sup>TM</sup>	MicroPak <sup>TM</sup>	QFET <sup>TM</sup>	SuperSOT <sup>TM-8</sup>
DOME <sup>TM</sup>	GlobalOptoisolator <sup>TM</sup>	MICROWIRE <sup>TM</sup>	QS <sup>TM</sup>	SyncFET <sup>TM</sup>
EcoSPARK <sup>TM</sup>	GTO <sup>TM</sup>	MSX <sup>TM</sup>	QT Optoelectronics <sup>TM</sup>	TinyLogic <sup>TM</sup>
E <sup>2</sup> CMOST <sup>TM</sup>	HiSeC <sup>TM</sup>	MSXPro <sup>TM</sup>	Quiet Series <sup>TM</sup>	TruTranslation <sup>TM</sup>
EnSigna <sup>TM</sup>	I <sup>2</sup> C <sup>TM</sup>	OCX <sup>TM</sup>	RapidConfigure <sup>TM</sup>	UHC <sup>TM</sup>
Across the board. Around the world. <sup>TM</sup>		OCXPro <sup>TM</sup>	RapidConnect <sup>TM</sup>	UltraFET <sup>®</sup>
The Power Franchise <sup>TM</sup>		OPTOLOGIC <sup>®</sup>	SILENT SWITCHER <sup>®</sup>	VCX <sup>TM</sup>
Programmable Active Droop <sup>TM</sup>		OPTOPLANAR <sup>TM</sup>	SMART START <sup>TM</sup>	

## DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

## LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

## PRODUCT STATUS DEFINITIONS

### Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.



## **Anexo C**

### **Pantalla digital. LCD**



classic mono LCDs

# SPECIFICATION CHARACTER TYPE DOT MATRIX LCD MODULE

ITEM NUMBER: FDCC0802C-FLYYBH-91LE  
ESTABLISHED DATE: 1999.06  
INITIAL ISSUED DATE: 2005.05  
DATASHEET VERSION: 2008 VERSION

*This item supplied to Farnell by MIDAS COMPONENTS*



MIDAS COMPONENTS LTD

Address: Electra House, 32 Southtown Road, Great Yarmouth, Norfolk, England, NR31 ODU  
Email: sales@midascomponents.co.uk  
Website: www.midascomponents.co.uk  
Tel: +44 (0)1493 602602 Fax: +44 (0)1493 665111



ISSUED BY: 魏惠东 CHECKED BY: 李金凤 APPROVED BY: 王海良

COPYRIGHT © 2000 ~ 2009 FORDATA ELECTRONIC CO., LTD. ALL RIGHTS RESERVED

## BOOKBINDING AREA

 <b>FORDATA ELECTRONIC CO.,LTD</b> PROFESSIONAL LCD SUPPLIER FROM CHINA	<b>STANDARD DOC.</b>	<b>DATASHEET STATEMENT</b>
--	--------------------------	----------------------------

1. The following icons are absolutely designed by FORDATA independently in 2007-SEP. They are not in common use in the LCD industry yet but just used for marking out FORDATA products' characteristics quickly and simply without any special meaning. FORDATA reserves the composing right and copyright. No one else is allowed to adopt these icons without FORDATA's approval.
2. The ISO9001 logo used in this document is authorized by SGS ([www.sgs.com](http://www.sgs.com)). FORDATA had already successfully passed the strict and professional ISO9001:2000 Quality Management System Certification and got the certificate (No.: CN07/00404)
3. The technologies/techniques/crafts which denoted by the following icons are not exclusively owned by FORDATA, but also shared by FORDATA's LCD strategic cooperators, however all these technologies/techniques/crafts have been finally confirmed by FORDATA's professional engineers and QC department.
4. As the difference in test standard and test conditions, also FORDATA's insufficient familiarity with the actual LCD using environment, all the referred information in this DATASHEET (including the icons) only have two functions:
  - 4.1: providing quick reference when you are judging whether or not the product meets your requirements.
  - 4.2: listing out definitely the tolerance.
5. The sequence of the icons is random and doesn't indicate the importance grade.
6. Icons explanation



FORDATA's 2006 version logo. FORDATA is an integrated manufacturer of flat panel display (FPD). FORDATA supplies TN, HTN, STN, FSTN monochrome LCD panel; COB, COG, TAB LCD module; and all kinds of LED backlight.

### classic mono LCDs



#### FAST RESPONSE TIME

This icon on the cover indicates the product is with high response speed; Otherwise not.



#### PROTECTION CIRCUIT

This icon on the cover indicates the product is with protection circuit; Otherwise not.



#### HIGH CONTRAST

This icon on the cover indicates the product is with high contrast; Otherwise not.



#### LONG LIFE VERSION

This icon on the cover indicates the product is long life version (over 9K hours guaranteed); Otherwise not.



#### WIDE VIEWING SCOPE

This icon on the cover indicates the product is with wide viewing scope; Otherwise not.



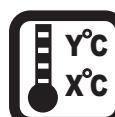
#### Anti UV VERSION

This icon on the cover indicates the product is against UV line. Otherwise not.



#### RoHS COMPLIANCE

This icon on the cover indicates the product meets ROHS requirements; Otherwise not.



#### OPERATION TEMPERATURE RANGE

This icon on the cover indicates the operating temperature range (X-Y).



#### 3TIMEs 100% QC EXAMINATION

This icon on the cover indicates the product has passed FORDATA's thrice 100% QC. Otherwise not.



#### TWICE SELECTION OF LED MATERIALS

This icon on the cover indicates the LED had passed FORDATA's twice strict selection which promises the product's identical color and brightness; Otherwise not.



#### 3.0V

This icon on the cover indicates the product can work at 3.0V exactly; otherwise not.

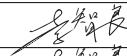
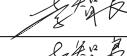
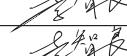
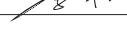


#### N SERIES TECHNOLOGY (2008 developed)

FORDATA adopts new structure, new craft, new technology and new materials inside both LCD module and LCD panel to improve the "RainBow"

## BOOKBINDING AREA

	<b>FORDATA ELECTRONIC CO.,LTD</b> PROFESSIONAL LCD SUPPLIER FROM CHINA	<b>STANDARD DOC.</b>	<b>REVISION RECORD</b>	<b>PAGE</b>	<b>1/20</b>
--	---	--------------------------	------------------------	-------------	-------------

<b>NO.</b>	<b>DATE</b>	<b>DESCRIPTION</b>	<b>ITEM</b>	<b>PAGE</b>	<b>APPROVED</b>
1	2005.05	INITIAL ISSUED	ALL	ALL	LU BOO
2	2007.04	Added further information of LED backlight	4	4/20	
3	2008.01	Adopt logos on the cover for fast reference	-	Cover	
4	2008.10	Deleted "N = No Ic" from CODE2	-	Code System	
5	2008.10	Added CODE "B" for DFSTN version in CODE7	-	Code System	

**BOOKBINDING AREA**

 <b>FORDATA ELECTRONIC CO., LTD</b> PROFESSIONAL LCD SUPPLIER FROM CHINA	<b>STANDARD</b>	<b>CODE SYSTEM</b>	<b>PAGE</b>	<b>1/1</b>
	<b>DOC.</b>	STANDARD COB		

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
FD	C	C	08	01	A	F	L	Y	Y	B	W	5	2	L	E

No.	REMARKS	DESCRIPTION		
1	COMPANY ABBRAVIATED	FD = FORDATA		
2	IC packing	C = Chip On Board      G = Chip On Glass      T = TAB		
3	LCM type	C = Character      G = Graphic		
4	Chyaracter	08, 10, 12, 16, 20, 24, 40, = Character number Per line		
	Graphic	80, 100, 120, 122, 128, 160 .... = Row Dots Quantity		
5	Character	01, 02, 04, = Character Lines		
	Graphic	32, 64, 80, 128, 160 .... = Column Dots Quantity		
6	Serial Number	A~Z		
7	Polarizer type	R = Positive Reflective      F = Positive Transflective M = Positive Transmissive      N = Negative Transmissive E = Negative, Transflective B = Negative, Dual optical compensation (for FSTN type only)		
8	Backlight type	N = No Backlight L = Array Type LED Backlight E = EL backlight without Invertor C = CCFL backlight without Invertor		
		S = Edge Type LED Backlight F = EL backlight with Invertor T = CCFL backlight with Invertor		
9	Backlight color	N = No Backlight      Y = Yellow-Green      W = White R = Red      A = Amber      C = Blue-Green B = Blue      G = Green		
10	LCD panel type	T = TN      H = HTN      Y = Yellow-Green STN G = Gray STN      B = Blue STN      F = FSTN		
11	Viewing angle	B = Bottom 6:00      T = Top 12:00      R = Right 3:00      L = Left 9:00		
12	Operation temperature range	S = 0°C ~ 50°C (Single Supply Voltage)      D = 0°C ~ 50°C (Dual Supply Voltage) W = -20°C ~ 70°C (Single Supply Voltage)      H = -20°C ~ 70°C (Dual Supply Voltage) T = -30°C ~ 80°C (Single Supply Voltage)      E = -30°C ~ 80°C (Dual Supply Voltage)		
13	Driving Voltage	1 : Vlcm = 3.0V, No / EL / CCFL Backlight or Vlcm = 3.0V, Vled = LED voltage, (Via AK) 2 : Vlcm = 3.6V, Vled = 5.0V (Not via AK) 3 : Vlcm = 3.6V, Vled = LED voltage, (Not via AK) 4 : Vlcm = 5.0V, Vled = LED voltage, (Not via AK) 5 : Vlcm = 5.0V, Vled = 5.0V (Not via AK) 6 : Vlcm = 5.0V, No / EL / CCFL Backlight or Vlcm = 5.0V, Vled = LED voltage, (Via AK) 7 : Vlcm = 3.6V, No / EL / CCFL Backlight or Vlcm = 3.6V, Vled = LED voltage, (Via AK) 8 : Vlcm = 3.0V, Vled = 5.0V 9 : Vlcm = 3.0V, Vled = LED voltage, (Not via AK)		
14	Backlight Connect Method	0 = PIN1 LED-, PIN2 LED+ 1 = PIN15(17/19) LED+, PIN16(18/20) LED- 2 = PIN15(17/19) LED-, PIN16(18/20) LED+ 3 = PIN15(17/19) LED+, PIN16(18/20) NC 4 = PIN15(17/19) NC, PIN16(18/20) LED+ 5 = PINA LED+, PINK LED- 6 = No / EL / CCFL Backlight		
15	IC Manufacturer	X = SAMSUNG      L = SUNPLUS      S = SITRONIX T = TOSHIBA      E = EPSON      H = HOLTEK Q = ASLIC      N = CIMTEK      P = PRINCETON		
16	Font Set	R = English - Russia      E = English - Japanese U = English - Europe      H = English - Hebrew K = English - Europe      N = NO FONT SET		

 <b>FORDATA ELECTRONIC CO., LTD</b> PROFESSIONAL LCD SUPPLIER FROM CHINA	STANDARD DOC.	CONTENTS	PAGE	3/20
---	------------------	----------	------	------

1. GENERAL SPECIFICATIONS -----	Page 4
2. MECHANICAL SPECIFICATIONS -----	Page 4
3. ABSOLUTE MAXIMUM RATINGS -----	Page 4
4. ELECTRONICAL CHARACTERISTIC -----	Page 4
5. OPTICAL CHARACTERISTICS -----	Page 5
6. ELECTRICAL SPECIFICATIONS-----	Page 6
7. EXTERNAL DIMENSION-----	Page 8
8. PIN ASSIGNMENT -----	Page 9
9. POWER SUPPLY -----	Page 9
10. REFLECTOR OF SCREEN AND DDRAM ADDRESS -----	Page 9
11. INSTRUCTION TABLE -----	Page 10
12. INSTRUCTION DESCRIPTION-----	Page 11
13. CGRAM -----	Page 15
14. DDRAM -----	Page 15
15. INITIALIZATION -----	Page 16
16. INTERFACE TO MPU -----	Page 18
17. FONT MAP -----	Page 19
18. PACKING DETAIL -----	Page 20

### BOOKBINDING AREA

 <b>FORDATA ELECTRONIC CO., LTD</b> PROFESSIONAL LCD SUPPLIER FROM CHINA	<b>PRODUCT SPEC.</b>	<b>MODE NO.</b>	<b>PAGE</b>	<b>4/20</b>
		FDCC0802C-FLYYBH-91LE		

### 1. GENERAL SPECIFICATIONS

ITEM	NOMINAL DIMENSIONS / AVAILABLE OPTIONS
DISPLAY FORMAT	8 Characters by 2 Lines
LCD PANEL OPTIONS	STN (Yellow-green color)
POLARIZER OPTIONS	Positive, Transflective
BACKLIGHT OPTIONS	Array type LED backlight (Yellow-green color)
VIEWING ANGLE OPTIONS	6:00 ( Bottom )
TEMPERATURE RANGE OPTIONS	Wide temp. range ( -20°C ~ 70°C )
CONTROLLER IC	SUNPLUS
DISPLAY DUTY	1/16
DRIVING BIAS	1/5

### 2. MECHANICAL SPECIFICATIONS

<b>OVERALL SIZE</b>	LED backlight version : 40.0 x 35.4 x max 13.0				<b>mm</b>
<b>VIEWING AREA</b>	30.4W x 13.9H	mm	<b>HOLE-HOLE</b>	36.0W x 30.0H	mm
<b>CHARACTER SIZE</b>	2.95W x 4.75H	mm	<b>CHARACTER PITCH</b>	0.40W x 0.40H	mm
<b>DOT SIZE</b>	0.55W x 0.55H	mm	<b>DOT PITCH</b>	0.05W x 0.05H	mm

### 3. ABSOLUTE MAXIMUM RATINGS

ITEM	SYMBOL	CONDITION	MIN	MAX	UNIT
POWER SUPPLY ( LOGIC )	Vdd	25°C	-0.3	7.0	V
POWER SUPPLY (LCD)	V0	25°C	Vdd -13.5	Vdd +0.3	V
INPUT VOLTAGE	Vin	25°C	-0.3	Vdd +0.3	V
OPERATING TEMPERATURE	Vopr	—	-20	70	°C
STORAGE TEMPERATURE	Vstg	—	-30	80	°C

### 4. ELECTRONICAL CHARACTERISTIC\*

ITEM	SYMBOL	CONDITION	STANDARD			<b>UNIT</b>
			MIN	TYP	MAX	
Input voltage	Vdd	+3V	2.7	3.0	4.5	V
Supply current	Idd	Vdd=3V	—	1.3	—	mA
Recommended LCD driving voltage for normal temp. Version module	Vdd - V0	-20°C	4.45	—	4.85	V
		0°C	4.35	—	4.70	
		25°C	4.30	4.50	4.70	
		50°C	4.20	—	4.60	
		70°C	3.90	—	4.55	
LED forward voltage	Vf	25°C	3.8	—	4.4	V
LED forward current	If	25°C	—	50	—	mA
LED reverse Current	Ir	25°C	—	—	50	μA
LED Peak wave length	λp	25°C If = 50mA	568	—	575	nm
LED illuminance (Without LCD)	Lv	25°C If = 50mA	—	70	—	cd/m <sup>2</sup>
LED life time	—	25°C If = 50mA	9K**	—	—	Hours

\* The above data are for reference only.

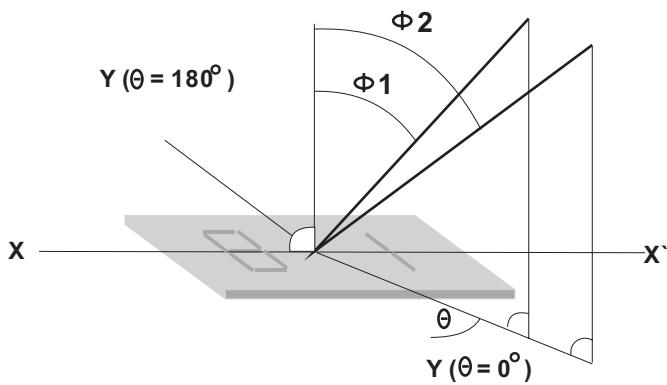
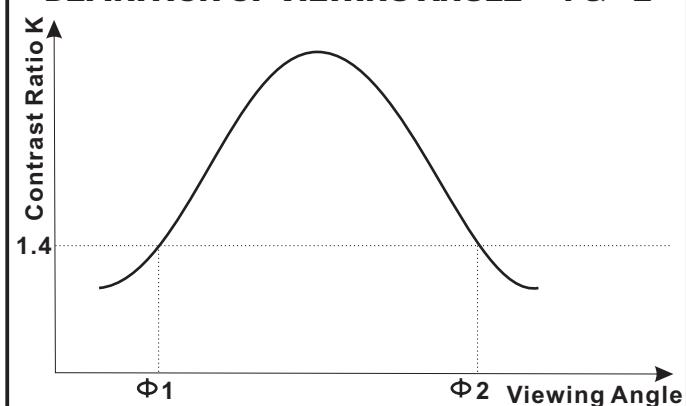
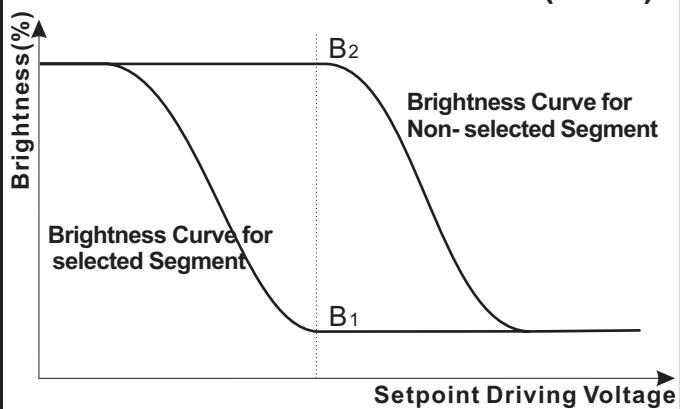
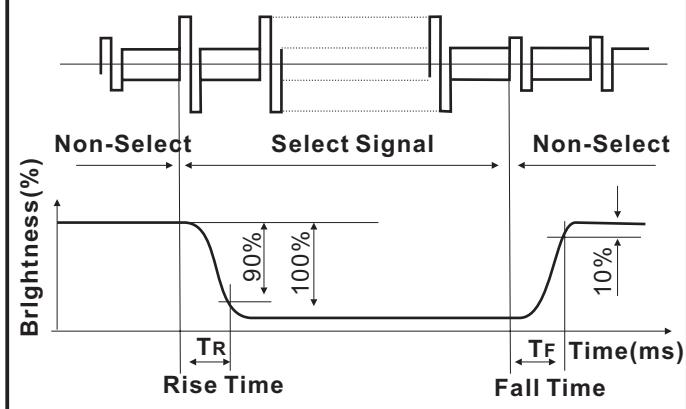
\*\* The warranty period of FORDATA LCD module is 1YEAR counted from the date shown on the label of products.


**5. OPTICAL CHARACTERISTIC**
**FOR TN TYPE LCD MODULE ( $T_A=25^{\circ}\text{C}$ ,  $V_{dd}=3.0\text{V} \pm 0.25\text{V}$ )**

ITEM	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
VIEWING ANGLE	$\phi_2 - \phi_1$	K=4	30	—	—	deg
	$\theta$		25	—	—	
CONTRAST RATIO	K	—	—	2	—	—
RESPONSE TIME(RISE)	$T_R$	—	—	120	150	ms
RESPONSE TIME(FALL)	$T_F$	—	—	120	150	ms

**FOR STN TYPE LCD MODULE ( $T_A=25^{\circ}\text{C}$ ,  $V_{dd}=3.0\text{V} \pm 0.25\text{V}$ )**

ITEM	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
VIEWING ANGLE	$\phi_2 - \phi_1$	K=4	40	—	—	deg
	$\theta$		60	—	—	
CONTRAST RATIO	K	—	—	6	—	—
RESPONSE TIME(RISE)	$T_R$	—	—	150	250	ms
RESPONSE TIME(FALL)	$T_F$	—	—	150	250	ms

**DEFINITION OF ANGLE  $\phi$  &  $\theta$** 

**DEFINITION OF VIEWING ANGLE  $\phi_1$  &  $\phi_2$** 

**DEFINITION OF CONTRAST RATIO K( $=B_2/B_1$ )**

**DEFINITION OF OPTICAL RESPONSE TIME  $T_R$  &  $T_F$** 


**BOOKBINDING AREA**

	<b>FORDATA ELECTRONIC CO., LTD</b> PROFESSIONAL LCD SUPPLIER FROM CHINA	PRODUCT SPEC.	MODE NO. FDCC0802C-FLYYBH-91LE	PAGE	6/20
--	--	------------------	-----------------------------------	------	------

## 6. ELECTRICAL SPECIFICATIONS

### 6.1.1 DC CHARACTERISTICS ( VDD = 2.7V to 4.5V, TA = 25 °C )

CHARACTERISTICS	SYMBOL	LIMIT			UNIT	TEST CONDITION
		MIN.	TYP.	MAX.		
INPUT HIGH VOLTAGE	V <sub>IH1</sub>	0.7Vdd	---	Vdd	V	Pins ( E. RS. R/W. DB0 - DB7 )
INPUT LOW VOLTAGE	V <sub>IL1</sub>	-0.3	---	0.55	V	
INPUT HIGH CURRENT	I <sub>IH</sub>	-1.0	---	1.0	μA	Pins ( RS. R/W. DB0 - DB7 )
INPUT LOW CURRENT	I <sub>IL</sub>	-5.0	-15	-30	μA	Vdd = 3.0V
OUTPUT HIGH VOLTAGE ( TTL )	V <sub>OH1</sub>	0.75Vdd	---	---	V	I <sub>OH</sub> = - 0.1mA Pins: DB0 - DB7
OUTPUT LOW VOLTAGE ( TTL )	V <sub>OL1</sub>	---	---	0.2Vdd	V	I <sub>OL</sub> = 0.1mA Pins: DB0 - DB7

### 6.1.2 AC CHARACTERISTICS ( VDD = 2.7V to 4.5V, TA = 25 °C )

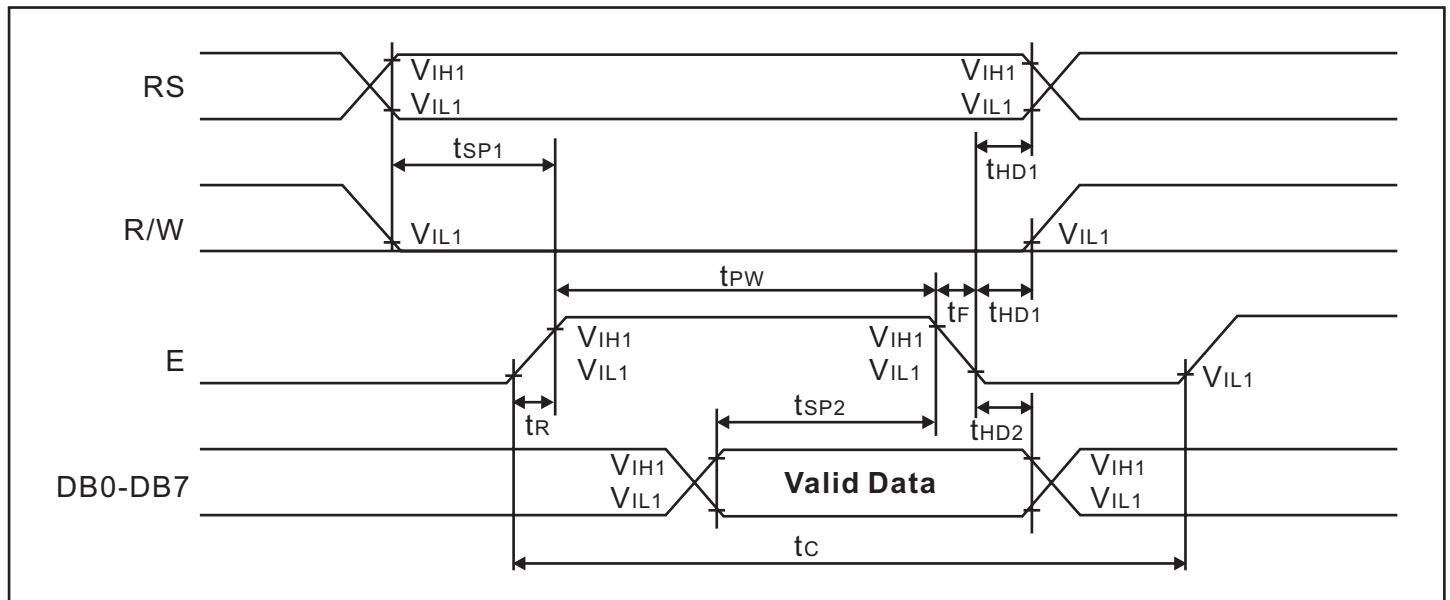
#### Write mode

CHARACTERISTICS	SYMBOL	LIMIT			UNIT	TEST CONDITION
		MIN.	TYP.	MAX.		
ENABLE CYCLE TIME	t <sub>c</sub>	1000	---	---	ns	Pin E
ENABLE PULSE WIDTH	t <sub>PW</sub>	450	---	---	ns	Pin E
ENABLE RISE/ FALL TIME	t <sub>R</sub> , t <sub>F</sub>	---	---	25	ns	Pin E
ADDRESS SETUP TIME	t <sub>SP1</sub>	60	---	---	ns	Pins RS, R/W, E
ADDRESS HOLD TIME	t <sub>HD1</sub>	20	---	---	ns	Pins RS, R/W, E
DATA SETUP TIME	t <sub>SP2</sub>	195	---	---	ns	Pins: DB0 - DB7
DATA HOLD TIME	t <sub>HD2</sub>	10	---	---	ns	Pins: DB0 - DB7

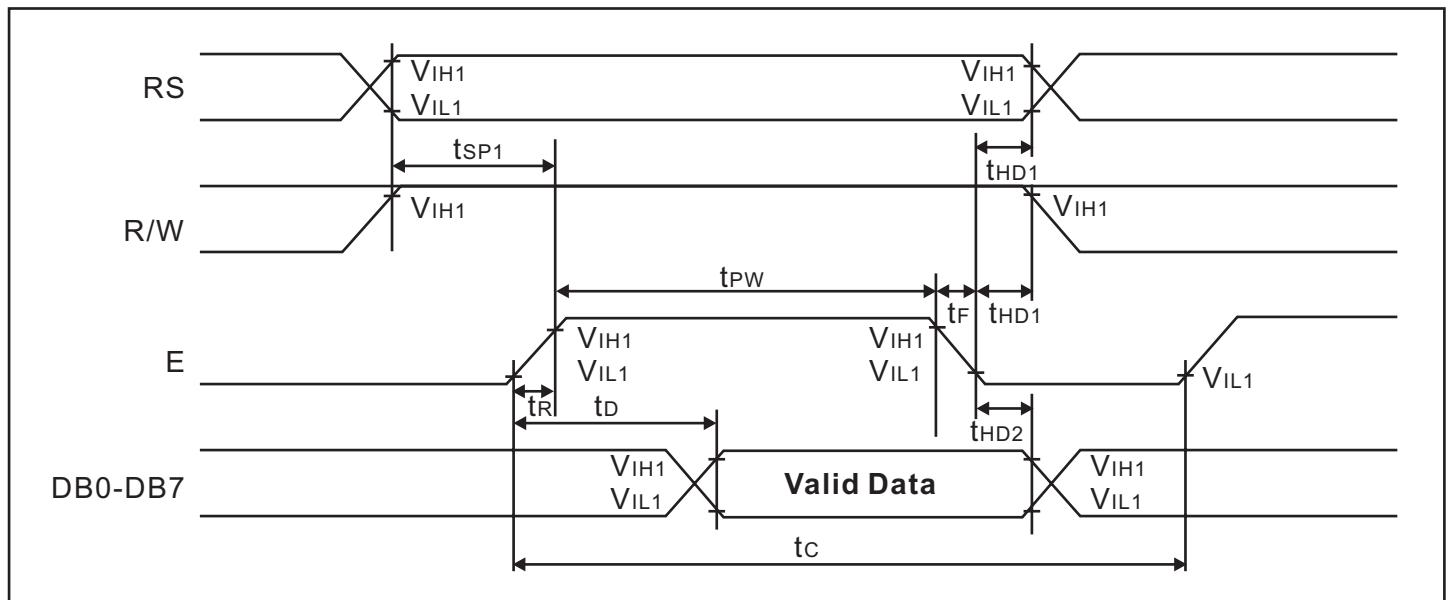
#### Read mode

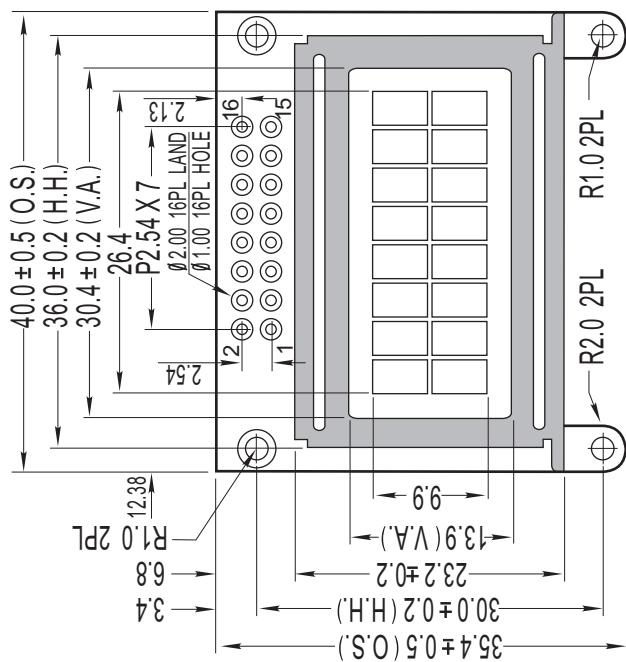
CHARACTERISTICS	SYMBOL	LIMIT			UNIT	TEST CONDITION
		MIN.	TYP.	MAX.		
ENABLE CYCLE TIME	t <sub>c</sub>	1000	---	---	ns	Pin E
ENABLE PULSE WIDTH	t <sub>PW</sub>	450	---	---	ns	Pin E
ENABLE RISE/ FALL TIME	t <sub>R</sub> , t <sub>F</sub>	---	---	25	ns	Pin E
ADDRESS SETUP TIME	t <sub>SP1</sub>	60	---	---	ns	Pins RS, R/W, E
ADDRESS HOLD TIME	t <sub>HD1</sub>	20	---	---	ns	Pins RS, R/W, E
DATA OUTPUT DELAY TIME	t <sub>D</sub>	---	---	360	ns	Pins: DB0 - DB7
DATA HOLD TIME	t <sub>HD2</sub>	5	---	---	ns	Pins: DB0 - DB7

### 6.3.1 WRITE MODE TIMING DIAGRAM

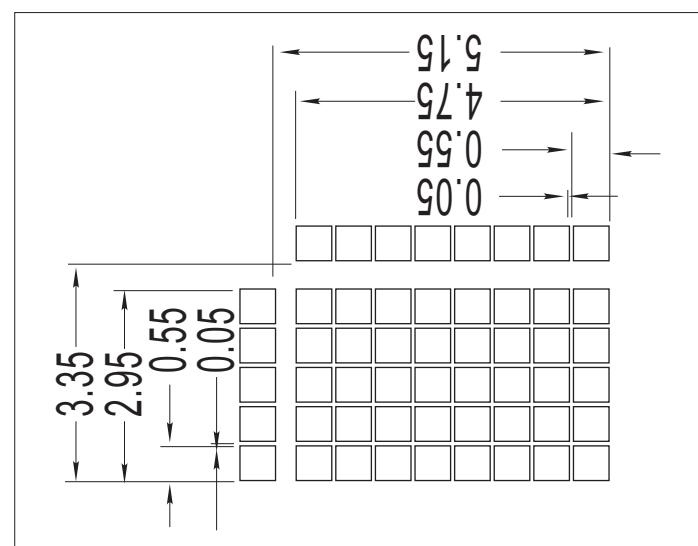
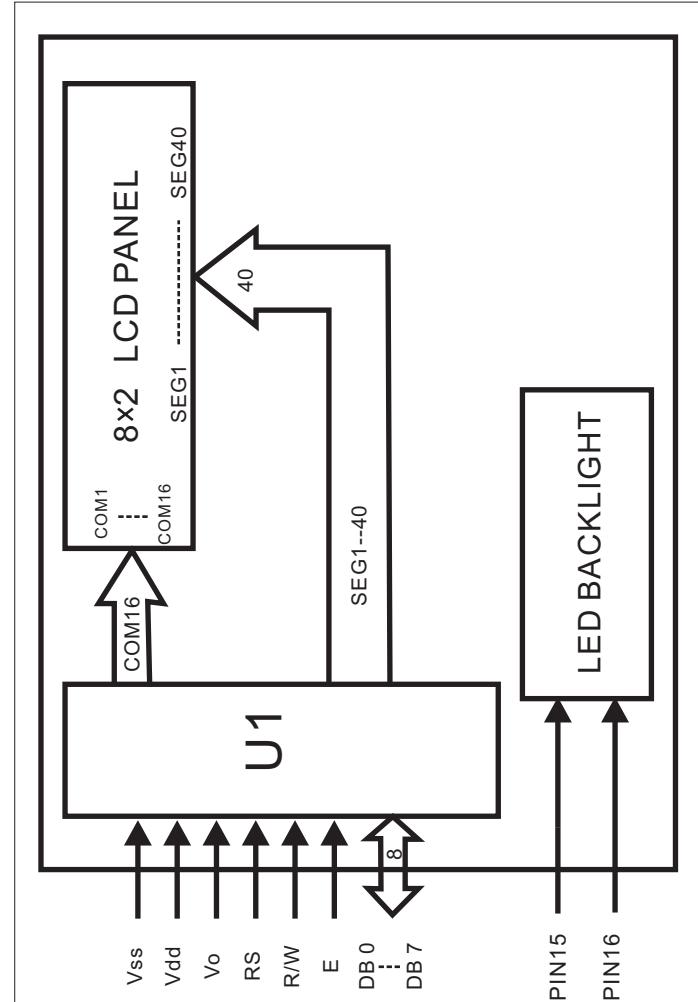


### 6.3.2 READ MODE TIMING DIAGRAM




**7. EXTERNAL DIMENSIONS**


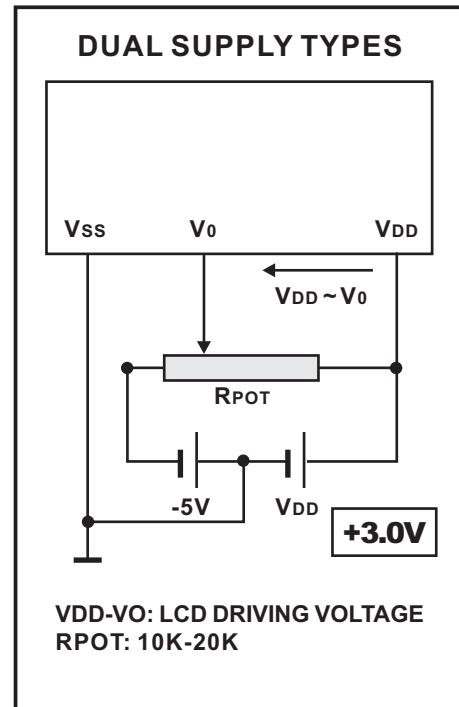
ITEM	T1	T2	UNIT
LED backlight	9.0	13.0	mm



<b>FORDATA ELECTRONIC CO.,LTD</b> PROFESSIONAL LCD SUPPLIER FROM CHINA	<b>PRODUCT SPEC.</b>	<b>MODE NO.</b> <b>FDCC0802C-FLYYBH-91LE</b>	<b>PAGE</b>	<b>9/20</b>
---	----------------------	---	-------------	-------------

**8.PIN ASSIGNMENT**

PIN	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	Power supply for LCM (+3.0V)
3	V0	Contrast Adjust
4	RS	Register Select Signal
5	R/W	Data Read / Write
6	E	Enable Signal
7-14	DB0 - DB7	Data bus line
15	LED+	Power supply for BKL (+4.2V)
16	LED-	Power supply for BKL (0V)

**9.POWER SUPPLY****10. REFLECTOR OF SCREEN AND DDRAM ADDRESS**

Display position	1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8		
DDRAM address	00	01	02	03	04	05	06	07	08	09
Display position										
DDRAM address	0A	0B	0C	0D	0E	0F	10	11	12	13
Display position										
DDRAM address	14	15	16	17	18	19	1A	1B	1C	1D
Display position										
DDRAM address	1E	1F	20	21	22	23	24	25	26	27
Display position	2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8		
DDRAM address	40	41	42	43	44	45	46	47	48	49
Display position										
DDRAM address	4A	4B	4C	4D	4E	4F	50	51	52	53
Display position										
DDRAM address	54	55	56	57	58	59	5A	5B	5C	5D
Display position										
DDRAM address	5E	5F	60	61	62	63	64	65	66	67

1-1 means first character of line 1 on screen

**BOOKBINDING AREA**

	<b>FORDATA ELECTRONIC CO.,LTD</b>	<b>PRODUCT SPEC.</b>	<b>MODE NO.</b>	<b>PAGE</b>	<b>10/20</b>
<b>PROFESSIONAL LCD SUPPLIER FROM CHINA</b>					<b>FDCC0802C-FLYYBH-91LE</b>

**11. INSTRUCTION TABLE**

Instruction	Instruction Code											Description	Execution Time(fosc=270kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Clear Display	0	0	0	0	0	0	0	0	0	1		Write 20H to DDRAM set DDRAM address to 00H from AC	1.52ms
Return Home	0	0	0	0	0	0	0	0	1	-		Set DDRAM address to 00H from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.52ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH		Assign cursor moving direction and enable the shift of entire display	38μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B		Set display(D) cursor(C) and blinking of cursor(B) on/off	38μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-		Set cursor moving and display shift control bit, and the direction, without changing DDRAM data	38μs
Function Set	0	0	0	0	1	DL	N	F	-	-		Set interface data length(DL:8bit/4bit), number of display line (N:2line/1line) and,display font type F:5X11dots / 5X8dots	38μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0		Set CGRAM address in address counter	38μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0		Set DDRAM address in address counter	38μs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0		Whether during internal operation or not can be known by reading BF The contents of address counter can also be read	0 μs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0		Write data into internal RAM (DDRAM/CGRAM)	38μs
Read data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0		Read data from internal RAM (DDRAM/CGRAM)	38μs

	<b>FORDATA ELECTRONIC CO.,LTD</b>	<b>PRODUCT SPEC.</b>	<b>MODE NO.</b>	<b>PAGE</b>	<b>11/20</b>
	PROFESSIONAL LCD SUPPLIER FROM CHINA		FDCC0802C-FLYYBH-91LE		

**12. INSTRUCTION DESCRIPTION****A. Clear Display**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Clear all the display data by writing 20H (space code) to all DDRAM address, and set DDRAM address to 00H into AC (address counter).

Return cursor to the original status, namely, bring the cursor to the left edge on the first line of the display.

Make the entry mode increment ( I/D = HIGH )

**B. Return Home**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	-

Set DDRAM address to 00H into the address counter.

Return cursor to its original site and return display to its original status, if shifted.

Contents of DDRAM does not change.

**C. Entry Mode Set**

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	SH

Set the moving direction of cursor and display.

**I/D:Increment /decrement of DDRAM address(cursor or blink)**

I/D=High,cursor/blink moves to right and DDRAM address is increased by 1.

I/D=low,cursor/blink moves to left and DDRAM address is decreased by 1.

\*CGRAM operates the same way as DDRAM, when reading from or writing to CGRAM.

**SH:Shift of entire display**

When DDRAM read (CGRAM read/write) operation or SH=Low, shifting of entire display is not performed.if SH=High, and DDRAM write operation,shift of entire display is performed according to I/D value(I/D=High,shift left, I/D=Low, shift right).

## BOOKBINDING AREA

	<b>FORDATA ELECTRONIC CO.,LTD</b>	<b>PRODUCT SPEC.</b>	<b>MODE NO.</b>	<b>PAGE</b>
	PROFESSIONAL LCD SUPPLIER FROM CHINA		FDCC0802C-FLYYBH-91LE	12/20

### D. Display ON/OFF Control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

#### D:Display ON/OFF control bit

When D=High, entire display is turned on.

When D=Low, display is turned off, but display data remains in DDRAM.

#### C:Cursor ON/OFF control bit

When C=High, cursor is turned on.

When C=Low, cursor is disappeared in current display ,but I/D register preserves its data.

#### B:Cursor Blink ON/OFF control bit

When B=High, cursor blink is on, which performs alternately between all the High data and display characters at the cursor position.

When B=Low ,blink is off.

### E. Cursor or Display Shift

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	-	-

Shifting of right/left cursor position or display without writing or reading of display data.

This instruction is used to correct or search display data.

During 2-line mode display, cursor moves to the 2<sup>nd</sup> line after the 40<sup>th</sup> digit of the 1<sup>st</sup> line.

Note that display shift is performed simultaneously in all the lines.

When displayed data is shifted repeatedly, each line is shifted individually.

When display shift is performed, the contents of the address counter are not changed.

S/C	R/L	Operation
0	0	Shift cursor to the left, AC is decreased by 1
0	1	Shift cursor to the right, AC is increased by 1
1	0	Shift all the display to the left,cursor moves according to the display
1	1	Shift all the display to the right,cursor moves according to the display

## F. Function set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	-	-

### DL:Interface data length control bit

When DL=High, it means 8-bit bus mode with MPU.

When DL=Low, it means 4-bit bus mode with MPU.

When 4-bit bus mode, it needs to transfer 4-bit data twice.

### N:Display line number control bit

When N=Low, 1-line display mode is set.

When N=High, 2-line display mode is set.

### F:Display font type control bit

When F=Low, 5x8 dots format display mode is set.

When F=High, 5x11 dots format display mode.

## G. Set CGRAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Set CGRAM address to AC.

This instruction makes CGRAM data available from MPU.

## H. Set DDRAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Set DDRAM address to AC.

This instruction makes DDRAM data available from MPU.

When 1-line display mode (N=Low), DDRAM address is from 00H to 4FH

In 2-line display mode(N=High),DDRAM address in the 1<sup>st</sup> line is from 00H to 27H and DDRAM address in the 2<sup>nd</sup> line is from 40H to 67H

	<b>FORDATA ELECTRONIC CO.,LTD</b>	<b>PRODUCT SPEC.</b>	<b>MODE NO.</b>	<b>PAGE</b>
	PROFESSIONAL LCD SUPPLIER FROM CHINA		FDCC0802C-FLYYBH-91LE	14/20

### I. Read Busy Flag & Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This instruction shows whether IC is in internal operation or not .

If BF is High, internal operation is in progress and shall wait until BF is to be Low, which by then the next instruction can be performed. In this instruction you and also read the value of the address counter.

### J. Write data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write binary 8-bit data to DDRAM/CGRAM.

The selection of RAM from DDRAM, and CGRAM, is set by the previous address set instruction(DDRAM address set,CGRAM address set).

RAM set instruction can also determine the AC direction to RAM.

After write operation, the address is automatically increased /decreased by 1, according the entry mode.

### K. Read data from RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

Read binary 8-bit data from DDRAM/CGRAM.

The selection of RAM is set by the previous address set instruction. If the address set instruction of RAM is not performed before this instruction, the data that has been read first is invalid, as the direction of AC is not yet determined. If RAM data is read several times without RAM address instructions set before read operation, the correct RAM data can be obtained from the second. But the first data would be incorrect, as there is no time margin to transfer RAM data.

	<b>FORDATA ELECTRONIC CO.,LTD</b>	<b>PRODUCT SPEC.</b>	<b>MODE NO.</b>	<b>PAGE</b>	<b>15/20</b>
PROFESSIONAL LCD SUPPLIER FROM CHINA			FDCC0802C-FLYYBH-91LE		

In case of DDRAM read operation,cursor shift instruction plays the same role as DDRAM address set instruction,it also transfers RAM data to output data register.

After read operation,address counter is automatically increased/decreased by 1 according to the entry mode.

After CGRAM read operation,display shift may not be executed correctly.

Note:In case of RAM write operation,AC is increased/decreased by 1 as in read operation.

At this time,AC indicates the next address position, but only the previous data can be read by the read instruction.

### 13. RELATIONSHIP BETWEEN CHARACTER CODE AND CGRAM

Character code	CGRAM Address	CGRAM Data	Pattern number
D7 D6 D5 D4 D3 D2 D1 D0	A5 A4 A3 A2 A1 A0	P7 P6 P5 P4 P3 P2 P1 P0	
0 0 0 0 x 0 0 0	0 0 0 0 0 0 0	x x x 0 1 1 1 0	pattern 1
	0 0 1	x x x 1 0 0 0 1	
	0 1 0	x x x 1 0 0 0 1	
	0 1 1	x x x 1 1 1 1 1	
	1 0 0	x x x 1 0 0 0 1	
	1 0 1	x x x 1 0 0 0 1	
	1 1 0	x x x 1 0 0 0 1	
	1 1 1	x x x 0 0 0 0 0	
0 0 0 0 x 1 1 1	0 0 0 0 0 0 0	x x x 1 0 0 0 1	pattern8
	0 0 1	x x x 1 0 0 0 1	
	0 1 0	x x x 1 0 0 0 1	
	0 1 1	x x x 1 1 1 1 1	
	1 0 0	x x x 1 0 0 0 1	
	1 0 1	x x x 1 0 0 0 1	
	1 1 0	x x x 1 0 0 0 1	
	1 1 1	x x x 0 0 0 0 0	

### 14. DISPLAY DATA RAM(DDRAM)

DDRAM stores display data of maximum 80x8 bits(80 characters).

DDRAM address is set in the address counter(AC) as a hexadecimal number

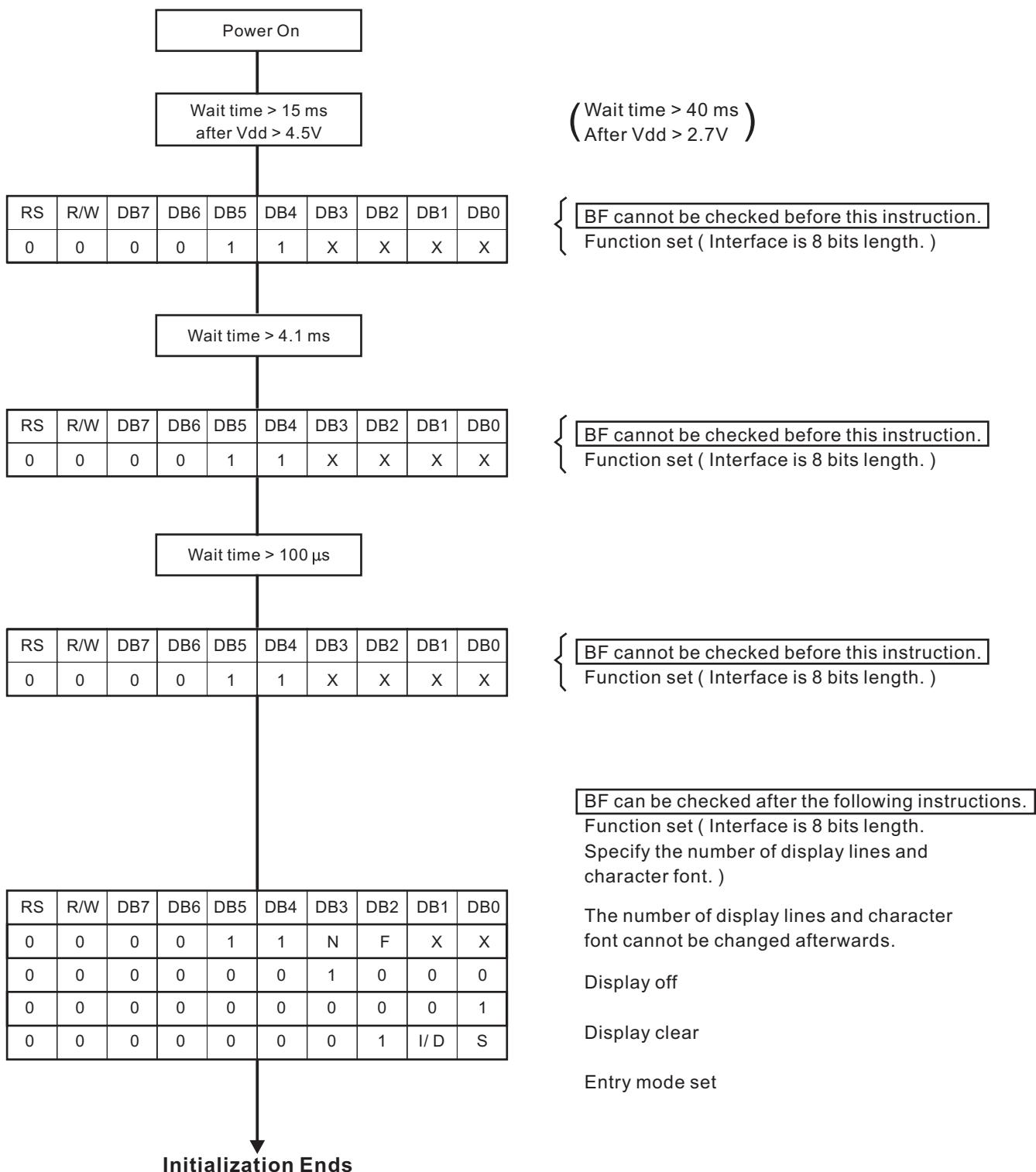
MSB

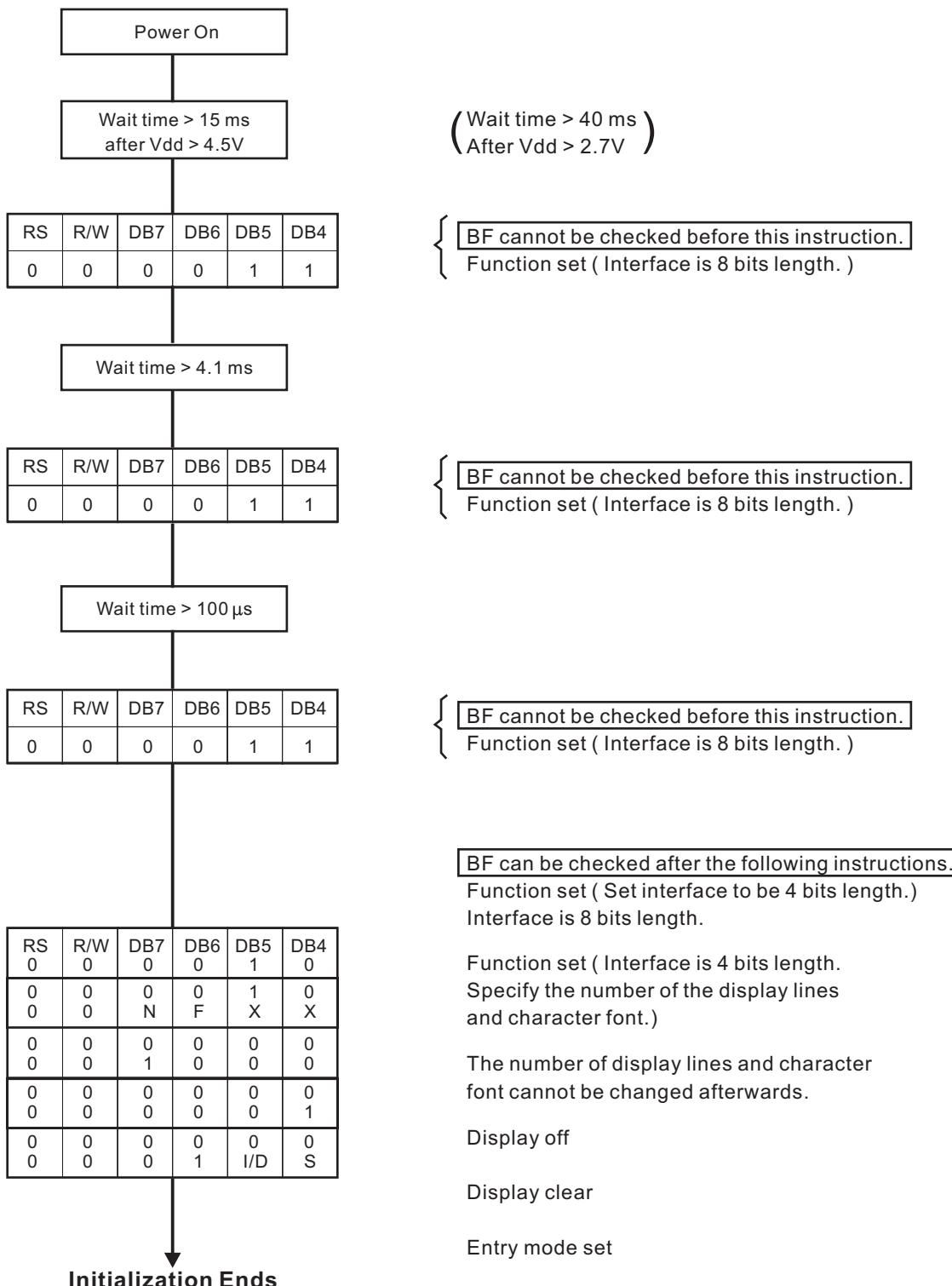
LSB

AC6	AC5	AC4	AC3	AC2	AC1	AC0
-----	-----	-----	-----	-----	-----	-----

## 15. INITIALIZATION

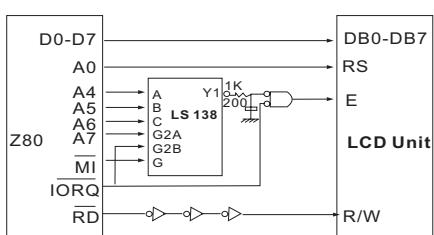
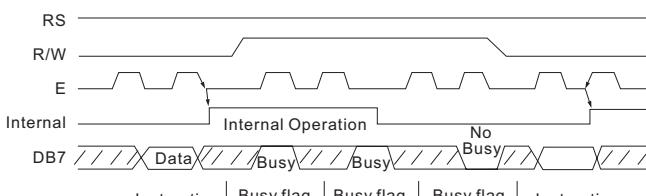
### **15.1 8-bit interface mode (Condition: fosc = 270KHZ)**



**15.2 4-bit interface mode (Condition: fosc = 270KHZ)**


## 16. INTERFACE TO MPU

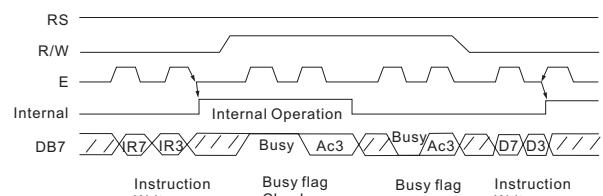
### INTERFACE TO 8-BIT MPU



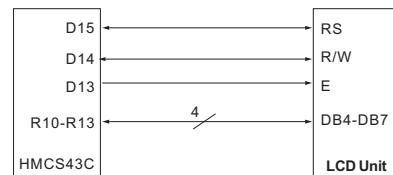
If Interface Data Is 8-bits Long

Data transfer is made through all 8 bus lines from DB0 to DB7

### INTERFACE TO 4-BIT MPU



Note: IR7,IR3: Instruction 7th bit,3rd bit  
Ac3: Address Counter 3rd bit



If Interface Data Is 4-bit Long

Data transfer is accomplished through 4 bus lines from DB4 to DB7.(while the rest of 4 bus lines from DB0 to DB3 are not used.)

Data transfer is completed when 4-bits of data is transferred twice.(upper 4-bits of data, then lower 4-bits of data.)

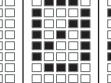
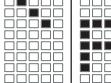
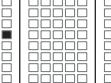
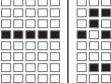
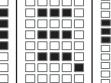
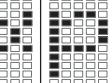
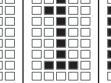
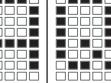
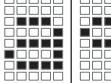
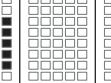
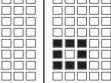
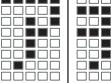
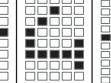
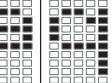
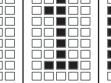
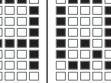
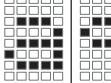
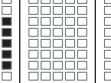
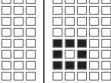
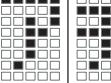
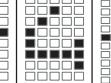
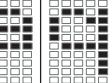
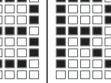
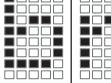
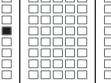
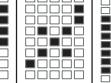
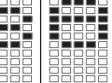
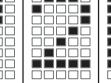
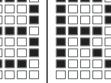
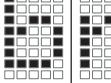
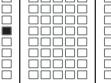
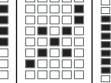
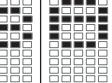
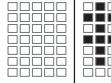
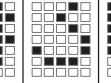
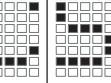
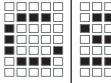
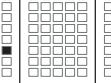
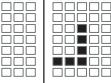
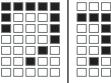
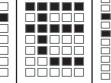
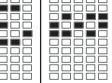
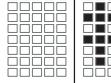
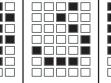
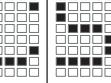
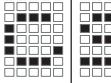
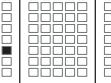
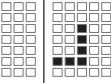
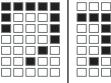
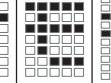
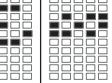
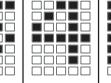
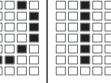
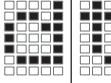
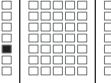
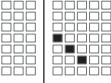
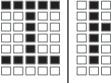
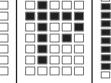
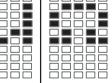
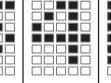
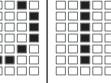
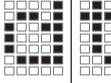
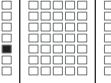
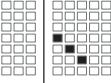
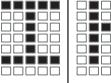
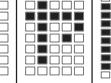
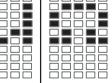
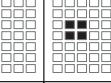
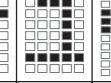
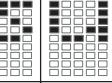
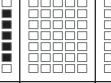
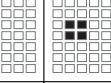
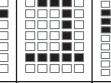
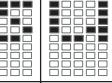
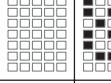
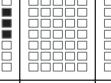
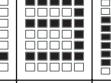
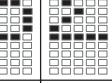
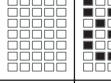
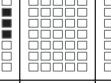
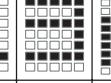
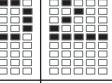
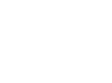
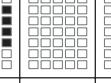
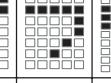
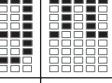
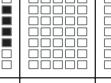
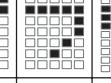
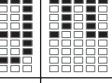
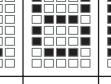
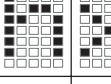
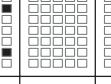
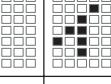
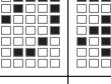
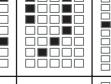
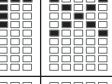
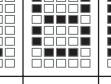
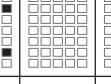
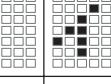
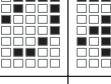
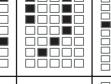
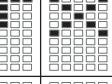
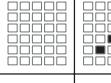
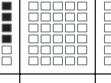
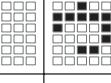
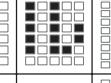
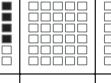
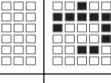
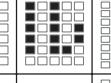
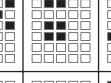
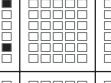
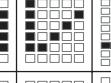
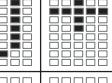
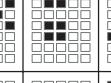
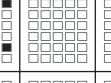
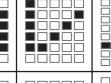
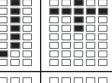
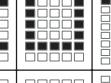
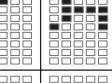
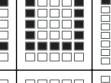
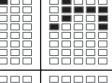
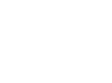
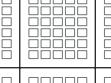
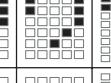
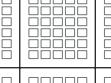
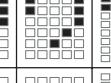
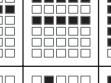
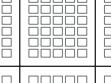
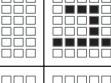
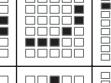
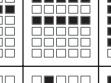
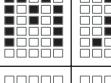
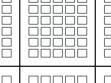
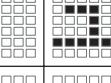
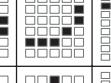
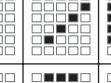
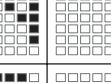
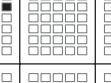
## Features

1. Interface to an 8-bit or 4-bit MPU is available.
2. 192 types of alphanumeric, symbols and special characters can be displayed with the built - in character generator (ROM).
3. Other preferred characters can be displayed by character generator (RAM).
4. Various instructions may be programmed.
  - Clear display
  - Cursor at home
  - On/Off cursor
  - Blink character
  - Shift display
  - Shift cursor
  - Read/Write display data .etc.
5. Compact and light weight design which can easily be integrated into end products.
6. Single power supply +5V drive (except for extended temperature type).
7. Low power consumption.

## BOOKBINDING AREA

	<b>FORDATA ELECTRONIC CO., LTD</b>	<b>PRODUCT SPEC.</b>	<b>MODE NO.</b>	<b>PAGE</b>	<b>19/20</b>
PROFESSIONAL LCD SUPPLIER FROM CHINA	FDCC0802C-FLYYBH-91LE				

**17. STANDARD FONT MAP**

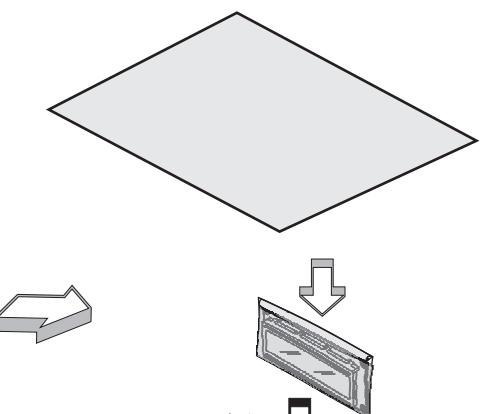
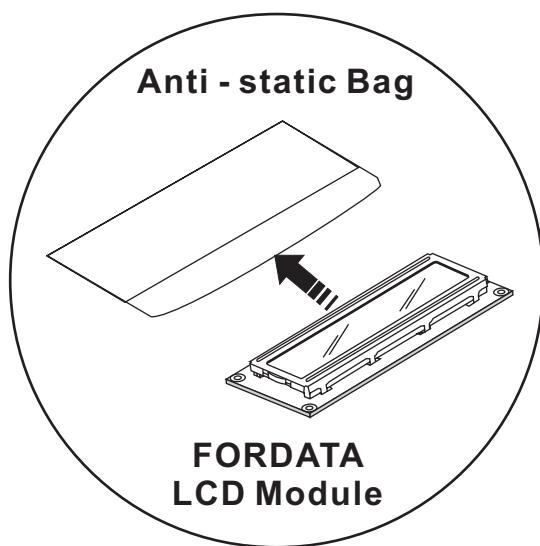
Upper 4bit Lower 4bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHHL	HHHH						
LLLL	CG RAM (1)																				
LLLH	(2)																				
LLHL	(3)																				
LLHH	(4)																				
LHLL	(5)																				
LHLH	(6)																				
LHHL	(7)																				
LHHH	(8)																				
HLLL	(1)																				
HLLH	(2)																				
HLHL	(3)																				
HLHH	(4)																				
HHLL	(5)																				
HHLH	(6)																<img alt="font map for (6) LLLL" data				

## 18. PACKING DETAIL

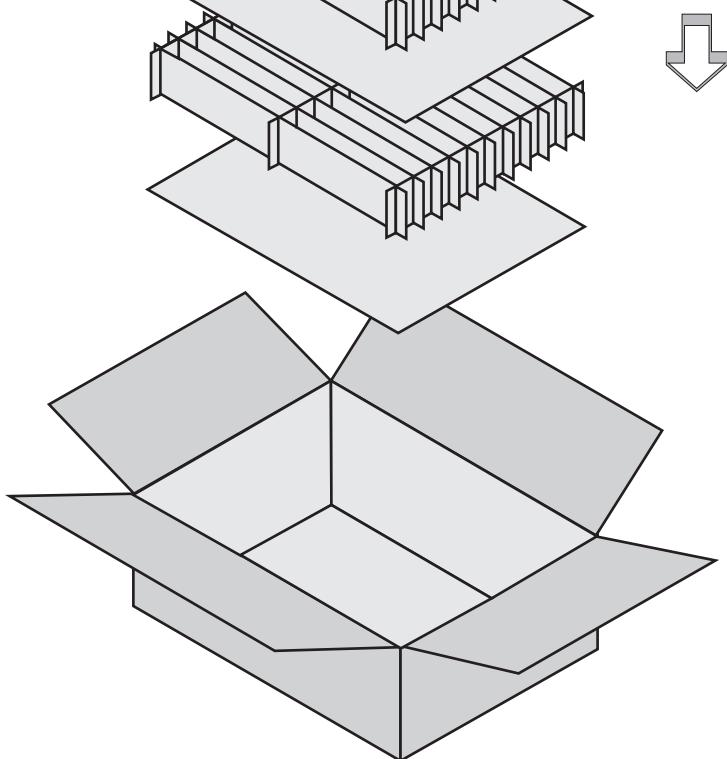
WITH LED BKL
45 PCS/BOX
10 BOXES/CARTON
450 PCS/CARTON
18.00 KGS/CTN(G.W.)
0.07 M <sup>3</sup> /CARTON

WITHOUT LED BKL
90 PCS/BOX
10 BOXES/CARTON
900 PCS/CARTON
16.00 KGS/CTN(G.W.)
0.07 M <sup>3</sup> /CARTON

NOTE
1. The weight is estimated for reference only.
2. Packing detail may be changed without notice.



**BOX** 



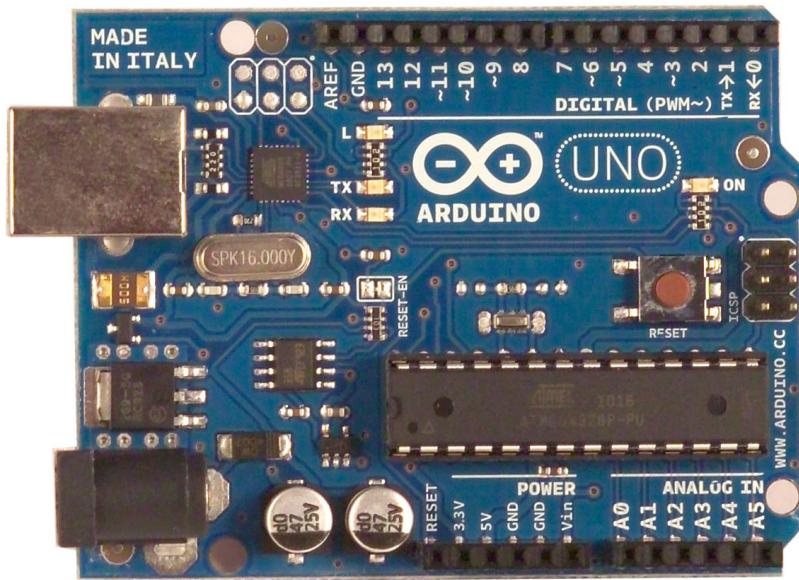
**CARTON** 



## **Anexo D**

### **Arduino Uno**

# Arduino UNO



## Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

## Index

Technical  
Specifications

Page 2

How to use Arduino  
Programming Environment, Basic Tutorials

Page 6

Terms &  
Conditions

Page 7

Environmental Policies  
half sqm of green via Impatto Zero®

Page 7



**radiospares**

**RADIONICS**



# Technical Specification

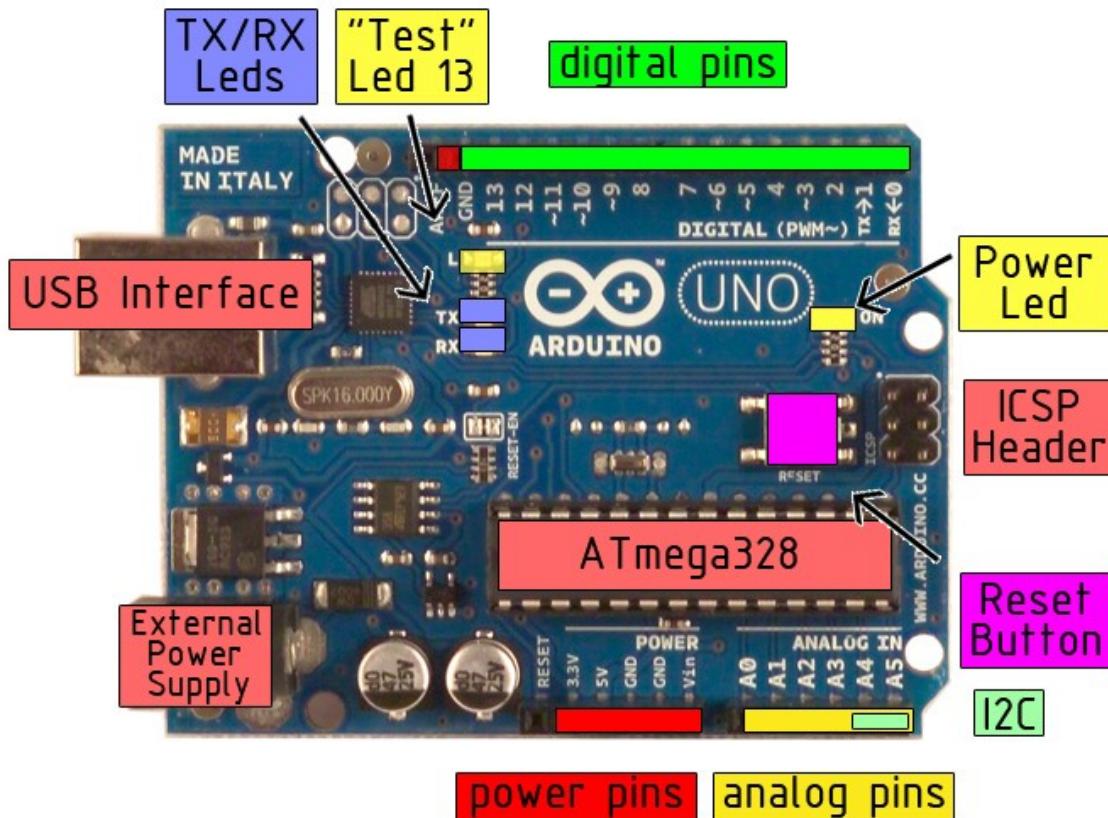


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

## Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

## the board



**radiospares**

**RADIONICS**



## Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



**radiospares**

**RADIONICS**



The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I<sup>2</sup>C: 4 (SDA) and 5 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an \*.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I<sup>2</sup>C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I<sup>2</sup>C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

## Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



**radiospares**

**RADIONICS**



## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



**radiospares**

**RADIONICS**



# How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

## Linux Install

## Windows Install

## Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

## Blink led

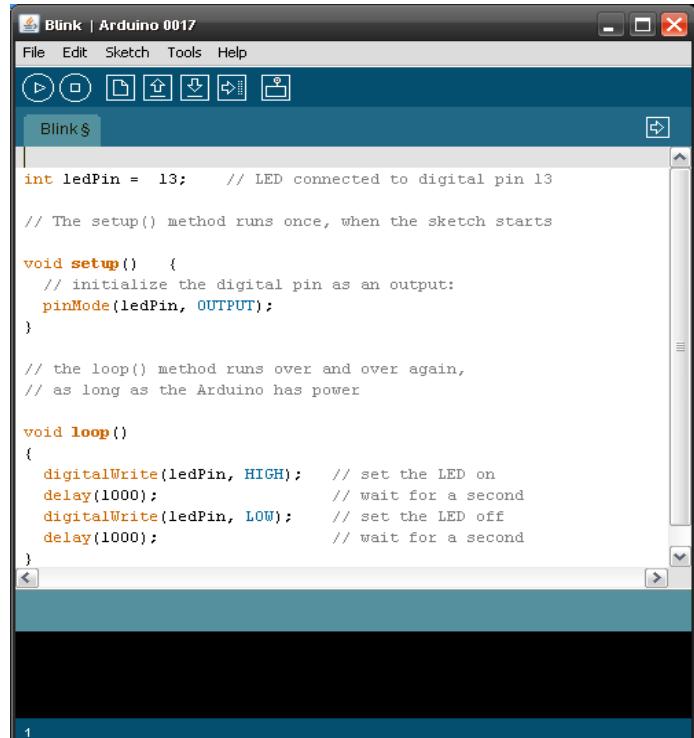
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>  
Arduino-0017>Examples>  
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to  
**Tools>SerialPort**  
and select the right serial port, the one arduino is attached to.



The screenshot shows the Arduino IDE interface with the title bar 'Blink | Arduino 0017'. The code editor contains the 'Blink' sketch:

```
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```

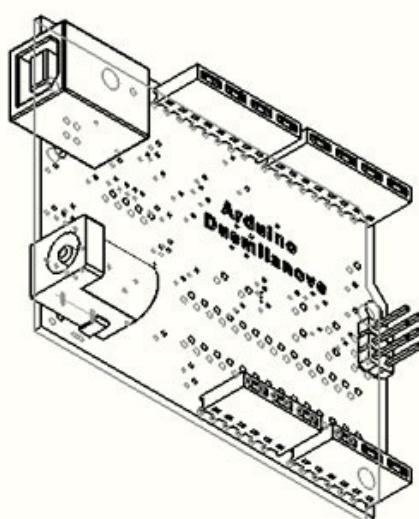
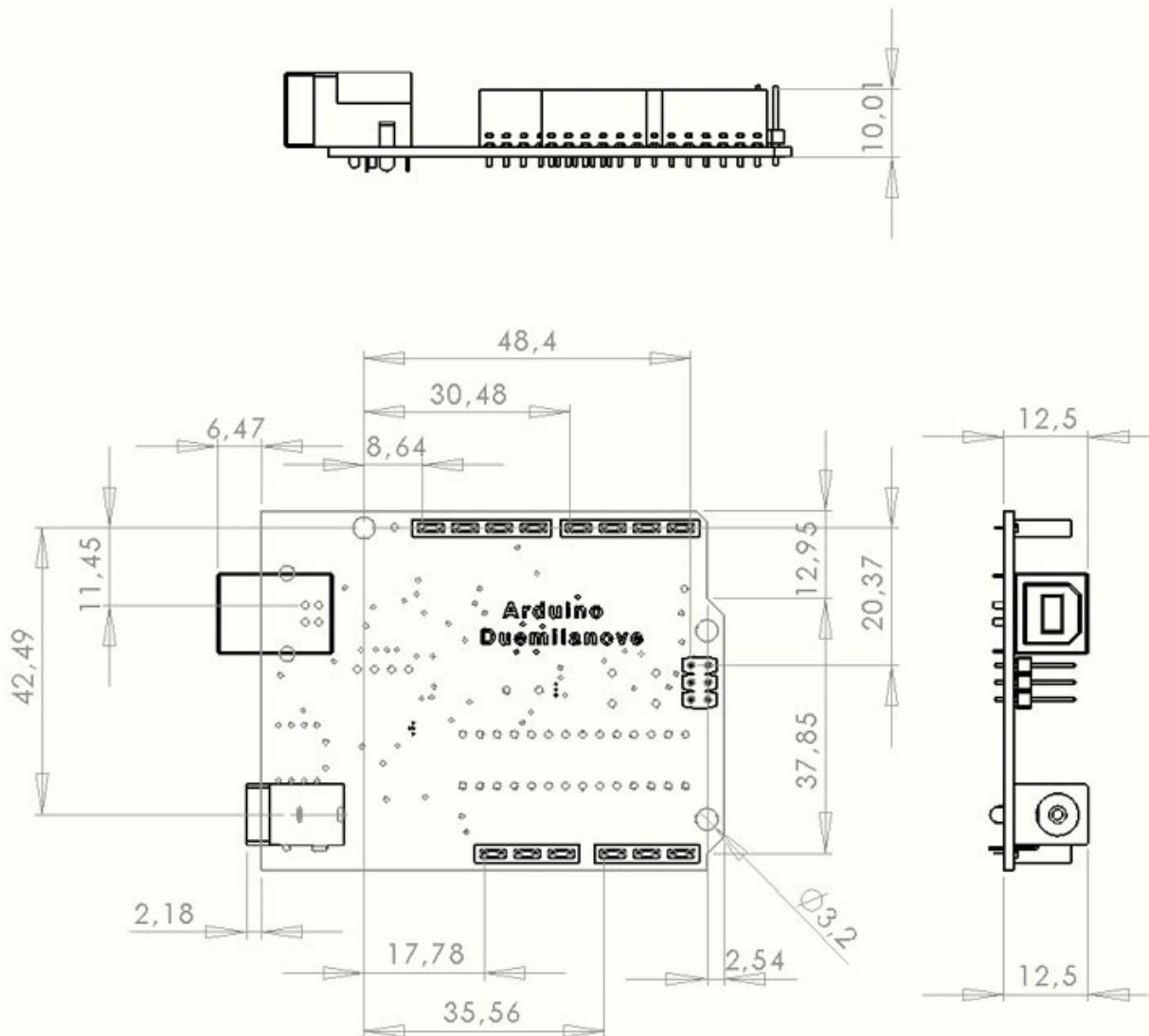


**radiospares**

**RADIONICS**



## Dimensioned Drawing



**radiospares**

**RADIONICS**



# Terms & Conditions



## 1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

## 2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

## 3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

## 4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



## Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



**radiospares**

**RADIONICS**





## **Anexo E**

### **Lenguaje Arduino**

# GUIA RÁPIDA LENGUAJE ARDUINO

## ESTRUCTURA

### Estructura principal

- void **setup()** (estructura de configuración)
- void **loop()** (estructura del bucle principal)

### Estructuras de control

- **if()**
- **if()...else**
- **for()**
- **switch()...case**
- **while()**
- **do()... while**
- **break**
- **continue**
- **return**
- **goto**

### Sintaxis

- ; (punto y coma)
- {} (corchetes)
- // (comentario línea única)
- /\* \*/ (comentario multilínea)

### Operadores aritméticos

- = (asignación)
- + (suma)
- - (resta)
- \* (multiplicación)
- / (división)
- % (módulo)

### Operadores de comparación

- == (igual que)
- != (no igual que)
- < (menor que)
- > (mayor que)

- <= (menor o igual que)
- >= (mayor o igual que)

### Operadores booleanos

- && (and)
- || (or)
- ! (not)

### Operadores compuestos

- ++ (incremento)
- -- (decremento)
- += (suma compuesta)
- -= (resta compuesta)
- \*= (multiplicación compuesta)
- /= (división compuesta)

### Constantes

- HIGH | LOW
- INPUT | OUTPUT
- true | false

### Tipo de datos

- **boolean** (booleano)
- **char** (carácter)
- **byte** (byte)
- **int** (entero)
- **unsigned int** (entero sin signo)
- **long** (entero largo)
- **unsigned long** (entero largo sin signo)
- **float** (coma flotante)
- **double** (coma flotante doble)
- **string** (cadena de texto)
- **array** (array)
- **void** (nada)

## FUNCIONES

### Funciones I/O digitales

- **pinMode(pin, mode)**
- **digitalWrite(pin, value)**
- **int digitalRead(pin)**

- **constrain(x, a, b)**
- **map(value, fromLow, fromHigh, toLow, toHigh)**
- **pow(base, exponente)**
- **sq(x)**
- **sqrt(x)**
- **sin(rad)**
- **cos(rad)**
- **tan(rad)**

### Funciones I/O analógicas

- **int analogRead(pin)**
- **analogWrite(pin, value)**

### Funciones números aleatorios

- **randomSeed(semilla)**
- **long random(max)**
- **long random(min, max)**

### Funciones I/O avanzadas

- **shiftOut(dataPin, clockPin, bitOrder, value)**
- **unsigned long pulseIn(pin, value)**

### Comunicaciones serie

- **Serial.begin(baudios)**
- **int Serial.available()**
- **int Serial.read()**
- **Serial.flush()**
- **Serial.print(datos)**
- **Serial.println(datos)**

### Funciones de tiempo

- **unsigned long millis()**

- **delay(milisegundos)**

- **delayMicroseconds(microsegundos)**

### Funciones matemáticas

- **min(x, y)**
- **max(x, y)**
- **abs(x)**

## Anexo F

# Software principal desarrollado

```
/*El siguiente programa, trata de establecer un control externo sobre un láser,  
tanto en la potencia del mismo como los testigos de puesta en marcha.  
Este programa puede ser ampliado para la lectura tanto de la corriente como  
de la potencia con la cual está funcionando el láser  
*/  
  
/*En este primer bloque, se declararan las variables necesarias para la  
posterior programación, así como los pines de entrada y salida de la tarjeta  
Arduino Uno. También se carga la biblioteca tanto de matemáticas (para las  
posibles operaciones) como la de la pantalla LCD (para la manipulación de la  
misma)  
*/  
  
//Estas variables, son las entradas y salidas de la tarjeta Arduino  
int pinPot = A2; //Salida del potenciómetro.  
int CS = 8; /*Interrumpor del DS1804. Para que este en posición ON, tiene  
que estar en estado bajo(LOW)*/  
int UD = 6; /*Dirección del DS1804. En estado bajo(LOW)se establece una  
dirección descendente y en estado alto(HIGH)la dirección establecida es  
ascendente*/  
int INC = 7; /*Posiciones a cambiar en el DS1804. Para cambiar una posición,  
es necesario que CS este ON. Un pulso bajo, otro alto y otro bajo, equivalen  
a una posición.*/  
  
/*Estas variables, son utilizadas durante el programa, para poder almacenar  
lecturas de los potenciómetros (manual y digital), realizar conversiones y  
operaciones necesarias.*/  
float valorPot; //Primera lectura del potenciómetro manual.  
float valorPot2; //Segunda lectura del potenciómetro manual.
```



```
float difPot; /* Incremento entre la primera lectura y la segunda del potenciómetro manual.*/
float valorDig; //Primera lectura del potenciómetro digital.
int valorDigEnt; //Valor de valorDig en entero.
float valorDig2; //Segunda lectura del potenciómetro digital
int valorDig2Ent; //Valor de valorDig2 en entero
float difDig; /*Incremento entre la primera lectura y la segunda, del potenciómetro digital*/
int difDigEnt; //Valor de difDig en entero
float voltInic; //Variable para almacenar el voltaje al inicio de la sesión.
float DS1804; /*Posición en la cual se encuentra el DS1804, en la segunda lectura, para poder hacer el cambio a voltaje*/
int DS1804Ent; //Valor de DS1804 en entero
int contador = 0; /*Marcará el límite para las posiciones del potenciómetro digital.*/
float a; /*Constante en la transformación lineal del potenciómetro manual al digital*/
float voltaje; //Valor final de la tensión(V)
float intlum; //Valor de la tensión lumínica(mW)

//A continuación se cargan las bibliotecas necesarias.
#include "math.h" //Incluir la biblioteca de matemáticas
#include <LiquidCrystal.h> //Incluir la biblioteca de LCD.

//Iniciación de la librería de la LCD con los pins: 12, 11, 5, 4, 3, 2
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

/*El siguiente bloque solo se inicia una vez cada vez que se enchufa la tarjeta arduino. En este bloque el potenciómetro digital, se establece en su posición correspondiente a la posición del potenciómetro manual. Se hacen los cambios y las transformaciones necesarias para que la posición mínima inicial del potenciómetro digital, sea la posición 60, ya que es a partir de esta posición cuando suministra unos 2,5 voltios, que es la tensión mínima que nos interesa.
*/
void setup()
{
    /*El primer paso es establecer los pines de la tarjeta arduino, como de entrada ó de salida, según corresponda.*/
}
```



```
Serial.begin (9600);
pinMode(pinPot, INPUT); // Entrada
pinMode(CS, OUTPUT); //Salida
pinMode(UD, OUTPUT); //Salida
pinMode(INC, OUTPUT); //Salida

/*Primero el potenciómetro digital se establece en la posición cero (para esto, bajamos 100 posiciones en el mismo para asegurarnos que empieza en la posición cero), para posteriormente establecerlo a la posición equivalente del potenciómetro manual.

*/
digitalWrite(CS,LOW);
digitalWrite(UD,LOW);
for(contador = 0; contador <= 100; contador++)
{
    digitalWrite(INC,LOW);
    delay(10);
    digitalWrite(INC,HIGH);
    delay(10);
    digitalWrite(INC,LOW);
}
contador = 0;

/*Establecimiento del potenciómetro digital en la posición equivalente a la del potenciómetro manual.*/
/*a = (99-60)/1023; Cálculo e la constante a, necesaria para pasar las 1023 posiciones del potenciómetro rotatorio a las 40 posiciones que quedan del DS1804.*/
a = 0.038124; /*Constante para la transformación lineal de las posiciones del potenciómetro manual al digital */
valorPot = analogRead(pinPot); /*Iguala la variable valorPot a la lectura hecha por el arduino de la entrada pinPot.*/
valorDig = (valorPot * a) + 60; /*Transformación lineal. Del manual al digital. Se le suma 60 para que la posición mínima en la que pueda empezar el potenciómetro digital sea la posición 60*/
valorDigEnt = (int)valorDig; /*Pasa el valor del número de posiciones de la primera medida del potenciómetro digital a un número entero*/

digitalWrite(UD,HIGH);
```



```
for(contador = 0; contador <= valorDigEnt; contador++)
{
    digitalWrite(INC,LOW);
    delay(10);
    digitalWrite(INC,HIGH);
    delay(10);
    digitalWrite(INC,LOW);
}
digitalWrite(UD, LOW);
digitalWrite(CS, HIGH);

/*Para mostrar la posición inicial en la pantalla del ordenador, instaurar
las dos siguientes lineas
Serial.println("Posicion inicial:");
Serial.println(valorDigEnt);
*/
voltInic= 0.829 * exp(0.018*valorDigEnt);
//Configuración de la pantalla LCD
lcd.begin(8, 2); //Ocho columnas y dos filas.
lcd.setCursor(0, 0); /*Establece el cursor de la LCD en la columna 0 y
en la fila 0.*/
lcd.print("Voltaje inicial:"); //Muestra en la LCD: "Voltaje inicial:"
for (int positionCounter = 0; positionCounter < 16; positionCounter++) {
    lcd.scrollDisplayLeft(); //Desplaza el texto hacia la izquierda
    delay(300);
}
for (int positionCounter = 0; positionCounter < 16; positionCounter++) /*32
posiciones, 16 que ocupa el texto mas 16 que se desplazaron a la izquierda*/
{
    lcd.scrollDisplayRight(); //Desplaza el texto hacia la derecha
    delay(300);
}
lcd.setCursor(0, 1);/*Establece el cursor de la LCD en la columna 0 y en
la fila 1.*/
lcd.print(voltInic); /*Muestra la variable valorDigEnt, que corresponde al
voltaje correspondiente a la posición inicial del potenciómetro digital.*/
```



```
lcd.setCursor(7, 1);
lcd.print("V");
}

/*En el siguiente bloque, se estableceran las instrucciones necesarias para
poder hacer cambiar el valor del potenciómetro digital, con cambios realizados
en el potenciómetro manual. Así como mostrar en la LCD dichos cambios y sus
correspondientes valores de voltaje. La definición de las distintas variables
viene en la primera parte del programa.
*/
void loop()
{
    /*A continuación, se realiza una primera lectura de la posición del
potenciómetro manual y se transforma a su posición digital correspondiente.*/
    valorPot = analogRead(pinPot);
    valorDig = (valorPot * a); /*Transformación lineal para poder transformar las
1023 posiciones del potenciómetro manual, a las 39 restantes del digital (Se
dice 39 por que el potenciómetro digital tiene 99 posiciones, pero como ya
se indicó con anterioridad, interesa que la posición mínima sea la 60)*/
    valorDigEnt = (int)valorDig;
    delay(1000);
    //A continuación se realiza la segunda lectura del potenciómetro manual
    valorPot2 = analogRead(pinPot);
    valorDig2 = (valorPot2 * a); //Transformación lineal de la nueva posición.
    valorDig2Ent =(int)valorDig2;

    /*Las siguientes operaciones, nos indican cuanto a cambiado el potenciómetro
manual, para poder comunicarselo al potenciómetro digital y que este cambie
las posiciones correspondientes.*/
    difPot = valorPot2 - valorPot; /*Diferencia entre la primera lectura y la
segunda.*/
    difDig = (difPot * a); /*Transformación lineal del incremento entre la primera
y la segunda medida*/
    difDigEnt = (int)difDig;

    /*se establecen 3 segundos útiles para poder cambiar el potenciómetro digital
a una nueva posición, para ello se ha establecido la siguiente cuenta atras,
durante la cual se puede cambiar el potenciómetro manual. */
    if(abs(difPot) > 2 )
```



```
{  
    delay(1000);  
    lcd.setCursor(0, 0);  
    lcd.print("T. mod.:");  
    lcd.setCursor(0, 1);  
    lcd.print("2 seg");  
    delay(1000);  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("T. mod.:");  
    lcd.setCursor(0, 1);  
    lcd.print("1 seg");  
    delay(1000);  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("T. mod.:");  
    lcd.setCursor(0, 1);  
    lcd.print("0 seg");  
    delay(1000);  
}  
  
//Se vuelve a realizar de nuevo la lectura de la nueva posición de valorPot2  
valorPot2 = analogRead(pinPot);  
valorDig2 = (valorPot2 * a); //Transformación lineal de la nueva posición.  
valorDig2Ent =(int)valorDig2;  
  
/*Las siguientes operaciones, indican cuanto a cambiado el potenciómetro  
manual, para poder comunicarselo al potenciómetro digital y que este cambie  
las posiciones correspondientes*/  
difPot = valorPot2 - valorPot; /*Diferencia entre la primera lectura y la  
segunda.*/  
difDig = (difPot * a); /*Transformación lineal del incremento entre la  
primera y la segunda medida*/  
difDigEnt = (int)difDig;  
/*A continuación se establecen las operaciones necesarias para poder cambiar  
el potenciómetro digital a medida que cambiamos el potenciómetro manual. */  
if( abs(difPot) > 2 ) /*Debido a que el potenciómetro manual, puede oscilar  
una posición sin que se le toque, se establece que el número de posiciones  
que este varíe tienen que ser dos para que se pueda establecer algún cambio  
en los datos recogidos por el programa.*/
```



---

```
{  
    digitalWrite(CS, LOW);  
  
    /*A continuación establecemos el potenciómetro digital según si el número  
    de posiciones aumenta o disminuye*/  
    if (difPot < -2) /*Con esto se indica que el número de posiciones esta  
    disminuyendo.*/  
    {  
        digitalWrite(UD,LOW);  
    }  
    else if (difPot > 2) /*Aquí se indica que el número de posiciones está  
    aumentando.*/  
    {  
        digitalWrite(UD,HIGH);  
    }  
  
    /*Una vez establecida la dirección del potenciómetro digital, con el  
    siguiente bucle for, se cambia el número de posiciones indicadas en la  
    variable difDigEnt*/  
    for (contador = 0; contador <= abs(difDigEnt); contador++)  
    {  
        /*como ya se indicó anteriormente, para cambiar una posición hace falta  
        una bajada-subida-bajada.*/  
        digitalWrite(INC,LOW);  
        delay(10);  
        digitalWrite(INC,HIGH);  
        delay(10);  
        digitalWrite(INC,LOW);  
  
    }  
    digitalWrite(UD,LOW);  
    digitalWrite(CS, HIGH);  
  
    /*A continuación, se transforma la segunda lectura a su valor  
    correspondiente en voltios.*/  
    a = 0.038124;  
    DS1804 = (valorPot2 * a) + 60; /*Transformación lineal de la segunda  
    lectura. Del manual al digital. Se le suma 60 por que la posición mínima en la
```



---

```
que se empezó era la posición 60.*/
DS1804Ent = (int)DS1804;
/*La siguiente transformación de las posiciones a tensión, se realiza
mediante una aproximación exponencial.*/
voltaje = 0.829 * exp(0.018*DS1804);
/*En el caso de haber optado por realizar una aproximación lineal (es
menos exacta), la transformación hubiese sido la siguiente:
voltaje = (abs(DS1804Ent) * 0.065) - 1.572;*/

/*Para mostrar en la pantalla del ordenador el número de posiciones cambiadas
y la posición mas reciente del potenciómetro digital, instaurar las dos siguientes
lineas
Serial.print("El potenciómetro cambio: ");
Serial.print(difDigEnt);
Serial.println("posiciones");
Serial.print ("Última posición introducida en el potenciómetro: ");
Serial.println(valorDig2Ent);
*/
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("cambios:");
lcd.setCursor(0, 1);
lcd.print(difDigEnt); /*Muestra en la LCD el número de posiciones cambiadas
por el potenciómetro digital.*/

delay(3000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("voltaje:");
lcd.setCursor(0, 1);
lcd.print(voltaje); /*Muestra en pantalla el voltaje correspondiente a la
última posición marcada.*/
lcd.setCursor(7, 1);
lcd.print("V");
delay(3000);

/*Transformación de la tensión seleccionada a su correspondiente valor de
```



---

```
intensidad lumínica (mW).*/
```

```
    if(voltaje > 3.8) /*Se pone esta limitación de voltaje, ya que hasta los  
3,8 V aproximadamente el laser no empieza a aumentar la intensidad lumínica.*/  
    {  
        intlum = 140.823 - 51391.325 * exp(-voltaje/0.638); /* Esta es la expresión  
para la aproximación exponencial decreciente que se ha llevado a cabo*/  
        lcd.clear();  
        lcd.setCursor(0, 0);  
        lcd.print("Int. Lum:");  
        lcd.setCursor(0, 1);  
        lcd.print(intlum); /*Muestra en pantalla la intensidad lumínica  
correspondiente a la última posición marcada.*/  
        lcd.setCursor(6, 1);  
        lcd.print("mW");  
    }  
    else if(voltaje <3.8)  
    {  
        lcd.begin(8, 2); //Ocho columnas y dos filas.  
        lcd.setCursor(0, 0); /*Establece el cursor de la LCD en la columna 0 y en  
la fila 0.*/  
        lcd.print("Tension baja");  
        for (int positionCounter = 0; positionCounter < 16; positionCounter++)  
        {  
            lcd.scrollDisplayLeft(); //Desplaza el texto hacia la izquierda  
            delay(300);  
        }  
    }  
  
    difPot = 0;  
}
```



## Anexo G

### Software para relación posición-tensión

```
/* Programa para poder hacer una tabla equivalente de las posiciones
de 0 a 99 con sus tensiones correspondientes*/
int CS = 8; //Interrumpor del DS1804. PIN del Arduino con LED integrado
int UD = 6; //Dirección del DS1804
int INC = 7; //Posiciones a cambiar en el DS1804
int contador= 0; /*Marcará el límite para que INC se esté quieto y no empiece
a subir o bajar*/
#include<math.h> //Incluir la biblioteca de matemáticas.

void setup()

{
    Serial.begin (9600);
    pinMode(CS, OUTPUT);      //CS Salida
    pinMode(UD, OUTPUT);      //UD Salida
    pinMode(INC, OUTPUT);     //INC Salida

    /*Lo siguiente baja el DS1804 100 posiciones, para que empiece a contar
desde cero*/
    digitalWrite(CS,LOW);
    digitalWrite(UD,LOW);
    for(contador = 0; contador <= 99; contador++)
    {
        digitalWrite(INC,LOW);
        delay(100);
        digitalWrite(INC,HIGH);
    }
}
```



---

```
    delay(100);
    digitalWrite(INC,LOW);
}

digitalWrite(CS, HIGH); //Para volver a cerrar el DS1804
contador = 0;

/*Va subiendo una a una las 99 posiciones del DS1804, dejando 3 segundos
entre posición y posición para que de tiempo a medir el voltaje manualmente.*/
digitalWrite(CS, LOW);
digitalWrite(UD, HIGH);
for(contador = 0; contador <= 99;contador++)
{
    digitalWrite(INC, LOW);
    delay(100);
    digitalWrite(INC, HIGH);
    delay(100);
    digitalWrite(INC, LOW);

    Serial.println("Posicion ascendente: ");
    Serial.println(contador);
    delay(5000);
}
contador = 99;
digitalWrite(CS, HIGH);
/*Va bajando una a una las 99 posiciones del DS1804, dejando 3 segundos
entre posición y posición para que de tiempo a medir el voltaje manualmente.*/
digitalWrite(CS, LOW);
digitalWrite(UD, LOW);
for(contador = 99; contador >= 0; contador--)
{
    digitalWrite(INC, LOW);
    delay(100);
    digitalWrite(INC, HIGH);
    delay(100);
    digitalWrite(INC, LOW);

    Serial.println("Posicion descendente: ");
    Serial.println(contador);
```



---

```
    delay(5000);
}
}

void loop()
{
}

}
```



# Anexo H

## Software inicial

### H.1. Lectura de potenciómetro rotatorio y conversión al digital.

```
/*Leer el potenciómetro manual y pasar las 1023 posiciones de este a las 99
posiciones que tiene el potenciómetro digital. Y ademas que el número de
posiciones sea un número entero*/
int pinPot = A2; /*Conecta la salida del potenciómetro manual al pin2
del Arduino*/
float valorPot; //Primera lectura del potenciómetro manual.
float valorPot2; //Segunda lectura del potenciómetro manual.
float difPot; /*Incremento entre la primera lectura y la segunda, del
potenciómetro manual*/
float valorDig; //Primera lectura del potenciómetro digital.
int valorDigEnt; //Valor de valorDig en entero.
float valorDig2; //Segunda lectura del potenciómetro digital
int valorDig2Ent; //Valor de valorDig2 en entero
float difDig; /*Incremento entre la primera lectura y la segunda, del
potenciómetro digital*/
int difDigEnt; //Valor de difDig en entero
int CS = 2; //Interrumpor del DS1804
int UD = 4; //Dirección del DS1804
int INC = 7; //Posiciones a cambiar en el DS1804
int contador = 0; /*Marcará el límite para que INC se esté quieto y no
empiece a subir o bajar*/
#include<math.h> //Incluir la biblioteca de matemáticas
void setup()
{
```



```
Serial.begin (9600);
pinMode(pinPot, INPUT); // Establece pinMode como entrada
pinMode(CS, OUTPUT); //CS Salida
pinMode(UD, OUTPUT); //UD Salida
pinMode(INC, OUTPUT); //INC Salida
}
void loop()
{
    valorPot = analogRead(pinPot); /* le da a la variable valorPot, la lectura
del pin A2;*/
    valorDig = valorPot*99/1023; /*Pasa las 1023 posiciones del potenciómetro
manual, a las 99 del digital*/
    valorDigEnt = (int)valorDig; /*Pasa el valor del número de posiciones de la
primera medida del potenciómetro digital a un número entero*/
    delay(3000);
    valorPot2 = analogRead(pinPot); /*le da a la variable valorPot2, la nueva
lectura del pin A2*/
    valorDig2 = valorPot2*99/1023; /*Pasa las 1023 posiciones del potenciómetro
manual, a las 99 del digital*/
    valorDig2Ent =(int)valorDig2; /*Pasa el valor del número de posiciones de la
segunda medida del potenciómetro digital a un número entero*/

    difPot = valorPot2 - valorPot; /*Diferencia entre la primera lectura y
la segunda*/
    difDig = difPot*99/1023; /*Pasa las 1023 posiciones del potenciómetro manual,
a las 99 del digital*/
    difDigEnt = (int)difDig; /*Pasa el valor de la diferencia de la primera medida
y la segunda a un número entero*/

if(abs(difPot) > 2) /*Para que solo aparezcan cambios cuando el potenciómetro
manual cambie mas de 2 posiciones, ya que puede variar una posición el solo,
así nos quitamos problemas con estos cambios que no podemos controlar*/
{
    digitalWrite(CS, HIGH); // Establece ON el interruptor del DS1804

    if (difPot < -2) //Si la diferencia de posiciones cambiadas es menor que -2
    {
        digitalWrite(UD,LOW); // Establece la posicion del DS1804 para abajo
```



```
}

else if (difPot > 2) /*Si la diferencia de posiciones cambiadas es mayor que 2*/
{
    digitalWrite(UD,HIGH); //Establece la posición del DS1804 para arriba
}

for (contador = 0; contador = difDigEnt; contador++)
{
    digitalWrite(INC,LOW);
    digitalWrite(INC,HIGH);
    delay(500);
    digitalWrite(INC,LOW);
    Serial.println("contador = " );
    Serial.println(contador);
}

Serial.print("El potenciómetro cambio: ");
Serial.print(difDigEnt);
Serial.println("posiciones"); /*Escribe en pantalla el valor de la variable valorDig*/
Serial.print ("Ultima posicion introducida en el potenciómetro: ");
Serial.println(valorDig2Ent);
//delay(5000); //Espera 5 segundos y con la siguiente orden se apaga.
//digitalWrite(CS, LOW);

difPot = 0;
}
}
```

## H.2. Pontenciómetros y pantalla LCD

/\*Lectura del potenciómetro manual y paso de las posiciones de este a las del potenciómetro digital(DS1804).

Establecimiento del Ds1804 en la posición 60, ya que es donde da el voltaje mínimo que nos interesa.

Conversión de las posiciones del DS1804 en voltaje mediante una aproximación lineal.

Mostrar datos en pantalla LCD.

\*/



```
int pinPot = A2; /*Conecta la salida del potenciómetro manual al pin2 del
Arduino */

float valorPot; //Primera lectura del potenciómetro manual.
float valorPot2; //Segunda lectura del potenciómetro manual.
float difPot; /*Incremento entre la primera lectura y la segunda, del
potenciómetro manual*/
float valorDig; //Primera lectura del potenciómetro digital.
int valorDigEnt; //Valor de valorDig en entero.
float valorDig2; //Segunda lectura del potenciómetro digital
int valorDig2Ent; //Valor de valorDig2 en entero
float difDig; /*Incremento entre la primera lectura y la segunda, del
potenciómetro digital*/
int difDigEnt; //Valor de difDig en entero
int CS = 8; //Interrumpor del DS1804. PIN del Arduino con LED integrado
int UD = 6; //Dirección del DS1804
int INC = 7; //Posiciones a cambiar en el DS1804
float DS1804; /*Posición en la cual se encuentra el DS1804, en la segunda
lectura, para poder hacer el cambio a voltaje*/
int DS1804Ent;

int contador = 0; /*Marca el límite para que INC se este quieto y no empiece
a subir o bajar*/
float a; /*Constante en la transformación lineal del potenciómetro manual
al digital*/
float voltaje;

#include "math.h" //Incluir la biblioteca de matemáticas
#include <LiquidCrystal.h> //Incluir la biblioteca de LCD.
// iniciar la librería de la LCD con los pins: 12, 11, 5, 4, 3, 2
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
    Serial.begin (9600);
    pinMode(pinPot, INPUT); // Establece pinMode como entrada
    pinMode(CS, OUTPUT); //CS Salida
    pinMode(UD, OUTPUT); //UD Salida
    pinMode(INC, OUTPUT); //INC Salida
```



/\*Lo siguiente baja el DS1804 100 posiciones, y luego lo fija al valor de la posición del potenciómetro rotatorio, para así cada vez que se encienda, el DS1804, tome el valor del potenciómetro rotatorio\*/

```
digitalWrite(CS,LOW);
digitalWrite(UD,LOW);
for(contador = 0; contador <= 100; contador++)
{
    digitalWrite(INC,LOW);
    delay(100);
    digitalWrite(INC,HIGH);
    delay(100);
    digitalWrite(INC,LOW);
}
contador = 0;
```

/\*La transformación siguiente, se hace de forma que el DS1804, comience en la posición 60 como minimo, ya que es a partir de esta posición cuando se alcanza la tensión mínima que interesa\*/

```
//a = (99-60)/1023; Explicación el la pag 40 del diario
a = 0.038124; /*Constante para la transformación lineal de las
posiciones del potenciómetro manual al digital*/
valorPot = analogRead(pinPot); /* le da a la variable valorPot, la
lectura del pin A2;*/
valorDig = (valorPot * a) + 60; /*transformación lineal. Del manual
al digital.*/
valorDigEnt = (int)valorDig; /*Pasa el valor del número de posiciones de
la primera medida del potenciómetro digital a un número entero*/

// 
digitalWrite(UD,HIGH);
for(contador = 0; contador <= valorDigEnt; contador++)
{
    digitalWrite(INC,LOW);
    delay(100);
    digitalWrite(INC,HIGH);
    delay(100);
    digitalWrite(INC,LOW);
```



```
}

digitalWrite(CS, HIGH);

/*Si se quiere que salga en la pantalla del ordenador, desestablecer
como comentario las siguientes instrucciones
Serial.println("Posicion inicial:");
Serial.println(valorDigEnt);
*/


//Establecimiento de las filas y columnas de la LCD.
lcd.begin(8, 2);
//Establecer el cursor de la LCD en la fila 0, columna 0
lcd.setCursor(0, 0);
//Escribir en la LCD: hola mundo
lcd.print("Pos.Inic");
//Establecer el cursor de la LCD en la fila 1, columna 0
lcd.setCursor(0, 1);
lcd.print(valorDigEnt);

}

void loop()
{
    valorPot = analogRead(pinPot); /* le da a la variable valorPot, la lectura
    del pin A2;*/

    valorDig = (valorPot * a); /*Pasa las 1023 posiciones del potenciómetro manual,
    a las 39 restantes del digital*/
    valorDigEnt = (int)valorDig; /*Pasa el valor del número de posiciones de la
    primera medida del potenciómetro digital a un número entero*/
    delay(5000);

    valorPot2 = analogRead(pinPot); /*le da a la variable valorPot2, la nueva
    lectura del pin A2*/

    valorDig2 = (valorPot2 * a); /*Pasa las 1023 posiciones del potenciómetro
    manual, a las 39 restantes del digital*/
    valorDig2Ent =(int)valorDig2; /*Pasa el valor del número de posiciones de la
    segunda medida del potenciómetro digital a un número entero*/

    difPot = valorPot2 - valorPot; /*Diferencia entre la primera lectura y la
    segunda*/
    difDig = (difPot * a); /*Pasa las 1023 posiciones del pótenciómetro manual,
```



```
a las 39 restantes del digital*/
difDigEnt = (int)difDig; /*Pasa el valor de la diferencia de la primera medida y la segunda a un número entero*/

lcd.clear(); //Limpia la pantalla

if( abs(difPot) > 2 ) /*Para que solo aparezcan cambios cuando el potenciómetro manual cambie mas de 2 posiciones, ya que puede variar una posición el solo, así nos quitamos problemas con estos cambios que no podemos controlar*/
{
    digitalWrite(CS, LOW); // Establece ON el interruptor del DS1804

    if (difPot < -2) //Si la diferencia de posiciones cambiadas es menor que -2
    {
        digitalWrite(UD,LOW); // Establece la posición del DS1804 para abajo
    }
    else if (difPot > 2) /*Si la diferencia de posiciones cambiadas es mayor que 2*/
    {
        digitalWrite(UD,HIGH); //Establece la posición del DS1804 para arriba
    }

    for (contador = 0; contador <= abs(difDigEnt); contador++) /*Introduce el número de pulsos necesarios hasta alcanzar la posición deseada*/
    {
        /*Los tres siguientes digitalWrite, meten una posición, ya que cada posición es una bajada-subida-bajada.*/
        digitalWrite(INC,LOW);
        delay(100);
        digitalWrite(INC,HIGH);
        delay(100);
        digitalWrite(INC,LOW);

        /*Si se desea que salga en la pantalla del ordenador, desestablecer como comentario las siguientes instrucciones
        Serial.println("contador = " );
        Serial.println(contador);
        */
    }
}
```



```
lcd.setCursor(0, 0);
lcd.print("contador");
lcd.setCursor(0, 1);
lcd.print(contador);
}
digitalWrite(CS, HIGH);

a = 0.038124; /*Constante para la transformación lineal de las posiciones
del potenciómetro manual al digital*/

DS1804 = (valorPot2 * a) + 60; //transformacion lineal
DS1804Ent = (int)DS1804;
/*Transformación de las posiciones a tension. Mediante aproximacion lineal,
cuya gráfica se encuentra en la pag 42 del diario*/
voltaje = (abs(DS1804Ent) * 0.065) - 1.572;
voltaje = 0.829 * exp(0.018*DS1804);

/*Si se desea que salga en la pantalla del ordenador, desestablecer como
comentario las siguientes instrucciones
Serial.print("El potenciómetro cambio: ");
Serial.print(difDigEnt);
Serial.println("posiciones"); /*Escribe en pantalla el valor de la
variable valorDig*/
Serial.print ("Última posición introducida en el potenciómetro: ");
Serial.println(valorDig2Ent);
Serial.print ("Última posición introducida en el potenciómetro: ");
Serial.println(valorDig2Ent);
*/
lcd.clear(); //Limpia la pantalla
lcd.setCursor(0, 0);
lcd.print("cambios:");
lcd.setCursor(0, 1);
lcd.print(difDigEnt);

delay(3000);
lcd.clear(); //Limpia la pantalla
lcd.setCursor(0, 0);
```



```
lcd.print("voltaje:");
lcd.setCursor(0, 1);
lcd.print(voltaje);
//delay(5000); //Espera 5 segundo y con la siguiente orden se apaga.
//digitalWrite(CS, LOW);
diffPot = 0;
}
}
```



# Anexo I

## Tablas de intensidad lumínica

Tensión(V)	Intensidad lumínica(mW)	Tensión(V)	Intensidad lumínica(mW)
4,93	340	3,68	200
	320		200
	310		200
	340		190
	330		200
4,73	300	3,46	200
	320		200
	300		190
	310		190
	310		210
4,59	300	3,26	190
	290		200
	300		200
	280		200
	300		200
4,29	280	3,12	200
	260		160
	270		190
	270		190
	270		190
4,08	270	3,01	200
	300		180
	240		190
	250		190
	280		190
3,96	240	2,65	180
	240		200



	220		200
	220		200
	250		200
3,82	210	2,44	180
	200		190
	200		180
	190		190
	200		190

Cuadro I.1: Tensión frente a intensidad luminosa. Medida1.

Datos de la segunda medición:

Tensión(V)	Intensidad lumínica(mW)	Tensión(V)	Intensidad lumínica(mW)
4,93	330 310 310 310 320	3,56	190 240 220 200 200
4,78	310 310 300 310 290	3,37	220 180 180 210 190
4,69	320 300 280 300 300	3,10	190 200 230 230 220
4,58	290 280 220 280 260	2,88	230 230 200 200 230
4,31	270 270 260 280 300	2,66	230 210 200 210 210
4,15	270	2,44	200



	220		190
	260		210
	260		220
	280		200
3,88	230		
	210		
	180		
	230		
	200		

Cuadro I.2: Tensión frente a intensidad luminosa. Medida2.