

Trabajo Final

**Nombre de la carrera / especialidad**

## **TÍTULO DEL TRABAJO**

**MEMORIA**

Autor/es: Apellido/s, Nombre/s

Docente/s: Apellido/s, Nombre/s

Lugar y Fecha: Mar del Plata, marzo de 2024



**LOGO**

**INSTITUCIÓN**

# Contenidos

<b>Contenidos</b>	<b>2</b>
<b>Resumen</b>	<b>3</b>
Licencia	3
<b>Prefacio</b>	<b>4</b>
1.1. Origen del proyecto	4
1.2. Motivación	4
1.3. Requisitos previos	4
<b>Introducción</b>	<b>5</b>
2.1. Objetivos del proyecto	5
2.2. Alcance del proyecto	5
<b>Supuestos</b>	<b>6</b>
3.1. Supuestos del proyecto	6
<b>Requerimientos del proyecto</b>	<b>7</b>
4.1. Requerimientos funcionales	7
4.2. Requerimientos de hardware	7
4.3. Requerimientos de firmware	7
4.4. Requerimientos no funcionales	7
<b>Entregables principales del proyecto</b>	<b>8</b>
<b>Presupuesto del tiempo</b>	<b>9</b>
6.1. Desglose del trabajo en tareas	9
6.2. Diagrama de Gantt	10
<b>Principio de funcionamiento</b>	<b>11</b>
<b>Componentes principales</b>	<b>15</b>
8.1. Placa Arduino Uno	15
8.2. Display de siete segmentos	16
8.3. Parlante	16
<b>Circuitos</b>	<b>18</b>
<b>Software</b>	<b>19</b>
10.1. Función principal void loop()	20
10.2. Librería LiquidCrystal	20
<b>Presupuesto</b>	<b>22</b>
<b>Conclusiones</b>	<b>23</b>
Próximos pasos	23
<b>Agradecimientos</b>	<b>24</b>
<b>Bibliografía</b>	<b>25</b>
Referencias bibliográficas	26
<b>Anexo I</b>	<b>27</b>
Código del proyecto	27
<b>Anexo II</b>	<b>28</b>

Hoja de datos del sensor DS18B20	28
<b>Anexo III</b>	<b>29</b>
Listado de software útil	29
<b>Anexo IV</b>	<b>30</b>
Licencias	30

# Resumen

Acá va el resumen del trabajo. Debe ser lo más breve posible. No más de dos o tres párrafos, de unas cuatro o cinco oraciones cada uno. Leyendo esto debe quedar muy claro en qué consiste el trabajo realizado, por qué el trabajo es importante, por qué el trabajo muestra que el estudiante aplicó correctamente lo aprendido en la carrera y qué información va a encontrar el lector en esta Memoria.

No usar en este resumen ninguna referencia bibliográfica del tipo [1], ni tampoco notas a pie de página ni siglas que no estén aclaradas como parte de este texto, ni tipografía en **negritas**, subrayada o *cursiva*. Dicho de otra forma, el texto en este resumen debe ser escrito de forma tal que si se recorta el mismo y se lo pega en un archivo de texto plano (extensión txt) entonces este conserve su formato y sea perfectamente entendible sin ningún agregado adicional, es decir, quede autocontenido.

Por ejemplo:

El objetivo principal de este proyecto es diseñar y construir un dispositivo funcional y fácil de manejar con el fin de poder controlar determinado proceso.

En esta memoria se describe todo el proceso que ha seguido el proyecto, desde la idea inicial hasta la creación del prototipo.

Para poder construir el dispositivo se ha empleado un microcontrolador compatible con Arduino.

Los resultados han sido más que satisfactorios y se han cumplido los objetivos planteados, dejando margen para posibles mejoras si se desea en un futuro.

## Licencia

Según la definición de Wikipedia [1], el derecho de autor es un conjunto de normas jurídicas y principios que afirman los derechos morales y patrimoniales que la ley concede a los autores (los derechos de autor), por el simple hecho de la creación de una obra literaria, artística, musical, científica o didáctica, esté publicada o inédita. La legislación sobre derechos de autor en Occidente se inicia en 1710 gracias al Estatuto de la Reina Ana.

Consultar Anexo IV para más información.

# Prefacio

A continuación, se explicará el porqué de la realización de este proyecto, así como las motivaciones e intereses personales que han llevado al autor a profundizar sobre este tema. Asimismo, se definirá el objetivo y el alcance de este trabajo.

## 1.1. Origen del proyecto

Por ejemplo: La idea de realizar este proyecto surgió a partir de ...

## 1.2. Motivación

Por ejemplo: Tal como se ha mencionado anteriormente, la motivación surge de haber realizado un proyecto utilizando la tecnología ..... Pero sin haber profundizado en ello.

## 1.3. Requisitos previos

Por ejemplo: Para poder llevar a cabo el proyecto es necesario contar con los siguientes conocimientos previos, asimismo es recomendable tener un conocimiento mínimo de ..... También haber trabajado con componentes ..... Y estar familiarizado con ..... También se valora positivamente haber practicado .....

# Introducción

La idea de esta sección es presentar el tema de modo que cualquier persona que no conoce el tema pueda entender de qué se trata y por qué es importante realizar este trabajo y cuál es su impacto.

Si en el texto se hace alusión a diferentes partes del trabajo, referirse a ellas como capítulo, sección o subsección según corresponda. Por ejemplo: “En el Capítulo 1 se explica tal cosa”, o “En la Sección 1.1 se presenta lo que sea”, o “En la subsección 1.1.1 se discute otra cosa”.

Entre párrafos sucesivos dejar un espacio, como el que se observa entre este párrafo y el anterior. Pero las oraciones de un mismo párrafo van en forma consecutiva, como se observa acá.

## 2.1. Objetivos del proyecto

Por ejemplo: El proyecto pretende diseñar, construir e implementar un dispositivo X mediante la tecnología X. Este se utilizará para tal cosa. Se realizará un estudio de los componentes compatibles existentes en el mercado para determinar cuáles son más adecuados para el proyecto. Todo el conjunto de componentes estará controlado por la tecnología X a partir de ..... Además, se diseñará un gabinete donde se colocarán todos los elementos que forman parte del proyecto para así obtener un prototipo funcional.

## 2.2. Alcance del proyecto

Por ejemplo: El alcance de este proyecto llegará hasta la creación de un prototipo funcional y preparado para determinado tipo de usuarios, cuya principal función será la de ..... Y se estudiará la implementación de otras funcionalidades que sean del interés del usuario. No se contempla el diseño de una versión comercial.

# Supuestos

## 3.1. Supuestos del proyecto

Deberemos listar los supuestos, es decir, que suponemos contar con determinados materiales, herramientas o instrumental. En el caso de utilizar una lista tabulada, el formato es el siguiente:

Este es el primer elemento de la lista.  
Este es el segundo elemento de la lista.  
Notar el uso de las mayúsculas y el punto al final de cada elemento.

Si se desea poner una lista numerada, el formato es el siguiente:

1. Este es el primer elemento de la lista.
2. Este es el segundo elemento de la lista.
3. Notar el uso de las mayúsculas y el punto al final de cada elemento.

Por ejemplo: Para el desarrollo del presente proyecto se supone:

Contaremos con todos los componentes para la construcción del dispositivo: placa Arduino modelo X, componentes adicionales (resistencias, sensores, capacitores, transistores), etcétera....

Contaremos con herramientas e instrumental necesario, tanto como para analizar las señales, como para el trabajo con la placa de desarrollo: fuente de laboratorio, osciloscopio digital, analizador lógico, herramientas varias, etcétera.....

Contaremos con infraestructura necesaria para realizar las pruebas, conexión de potencia a la red eléctrica, PC de escritorio para recibir el estado de las salidas, artefactos eléctricos a controlar, etcétera .....

# Requerimientos del proyecto

## 4.1. Requerimientos funcionales

Por ejemplo:

1. REQ-001. Aceptar la codificación de distintos dispositivos mediante la utilización únicamente del protocolo RC5. [Prioridad Baja].
2. REQ-002. Tener un radio de lectura de la señal infrarroja de 10 metros en un ambiente interior sin demasiada luz solar. [Prioridad Intermedia].
3. REQ-003. Poder conectarse a cualquier dispositivo que utilice una alimentación mediante la red eléctrica. [Prioridad Alta].

## 4.2. Requerimientos de hardware

Por ejemplo:

1. REQ-004. Mostrar mediante LEDs las salidas activas. [Prioridad Baja].
2. REQ-005. Usar salidas provistas de optoacopladores y relés. [Prioridad Alta].
3. REQ-006. Usar una fuente de alimentación externa (AC/DC). [Prioridad Alta].

## 4.3. Requerimientos de firmware

Por ejemplo:

1. REQ-007. Interpretar la codificación del protocolo RC5 de 14 bits. [Prioridad Alta].
2. REQ-008. Conmutar cada salida sin alterar el resto. [Prioridad Alta].
3. REQ-009. Permitir cambiar la relación entrada/salida. [Prioridad Intermedia].
4. REQ-010. Aprender nuevos códigos según el control remoto. [Prioridad Baja].
5. REQ-011. Llevar un registro de los cambios en la salida. [Prioridad Intermedia].

## 4.4. Requerimientos no funcionales

Por ejemplo:

1. Fácil de utilizar: la utilización del sistema por parte del usuario final deberá ser sencilla y con pocos parámetros de carga.
2. Adaptable: será un sistema versátil, con posibilidad de configurar las diferentes teclas del control remoto disponible para satisfacer las necesidades y facilidades de cada usuario.
3. Plug and Play: sencillo de instalar, simplemente enchufe y use, sin necesidad de controladores de hardware complicados.



# Entregables principales del proyecto

Listar todo el material que se entregará con el dispositivo, por ejemplo:

- Manual de usuario.
- Diagrama esquemático y diseño de PCB.
- Documentación relacionada con el diseño de hardware.
- Código fuente (firmware y software).
- Documentación del código fuente (firmware y software).
- Diagrama de instalación.
- Informe final.

# Presupuesto del tiempo

## 6.1. Desglose del trabajo en tareas

Por ejemplo:

1. Diseño del hardware (60 horas).
  - a) Elección del sensor, periféricos y componentes adicionales necesarios (6 horas).
  - b) Análisis de señales RC5 con controles remotos varios (6 horas).
  - c) Diseño e implementación del circuito del prototipo (48 horas).
2. Desarrollo del firmware (54 horas).
  - a) Planteo y modularización del código en distintos modos: “Aprendizaje”, “Estado” y “Ejecución” (12 horas).
  - b) Captura y tratamiento de las señales RC5 (12 horas).
  - c) Procesamiento de las señales obtenidas y manejo del puerto de salida GPIO (12 horas).
  - d) Envío de información de estados de salidas al PC mediante USB (12 horas).
  - e) Testing del firmware (6 horas).
3. Desarrollo del software para el reporte de estado (44 horas).
  - a) Elección de la plataforma (2 horas).
  - b) Desarrollo e implementación del software (36 horas).
  - c) Testing con hardware conectado (6 horas).
4. Prueba y corrección de fallos en el montaje (12 horas).
  - a) Testing sistema completo: firmware, software y hardware (12 horas).
5. Elaboración de un informe con los detalles técnicos de desarrollo (16 horas).
  - a) Elaboración de la documentación de hardware (4 horas).
  - b) Elaboración de la documentación de firmware (4 horas).
  - c) Elaboración de la documentación de software (4 horas).
  - d) Elaboración de la documentación de pruebas del sistema (4 horas).

Tiempo estimado total en horas: 186 horas.

Fecha de inicio: 1 de marzo de 2023.

Fecha de finalización: 1 de noviembre de 2023.

Fecha límite: 7 de noviembre de 2023.

## 6.2. Diagrama de Gantt

Según la definición de Wikipedia [2] un diagrama de Gantt es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Por ejemplo, tal cual ilustra la figura 6.1:

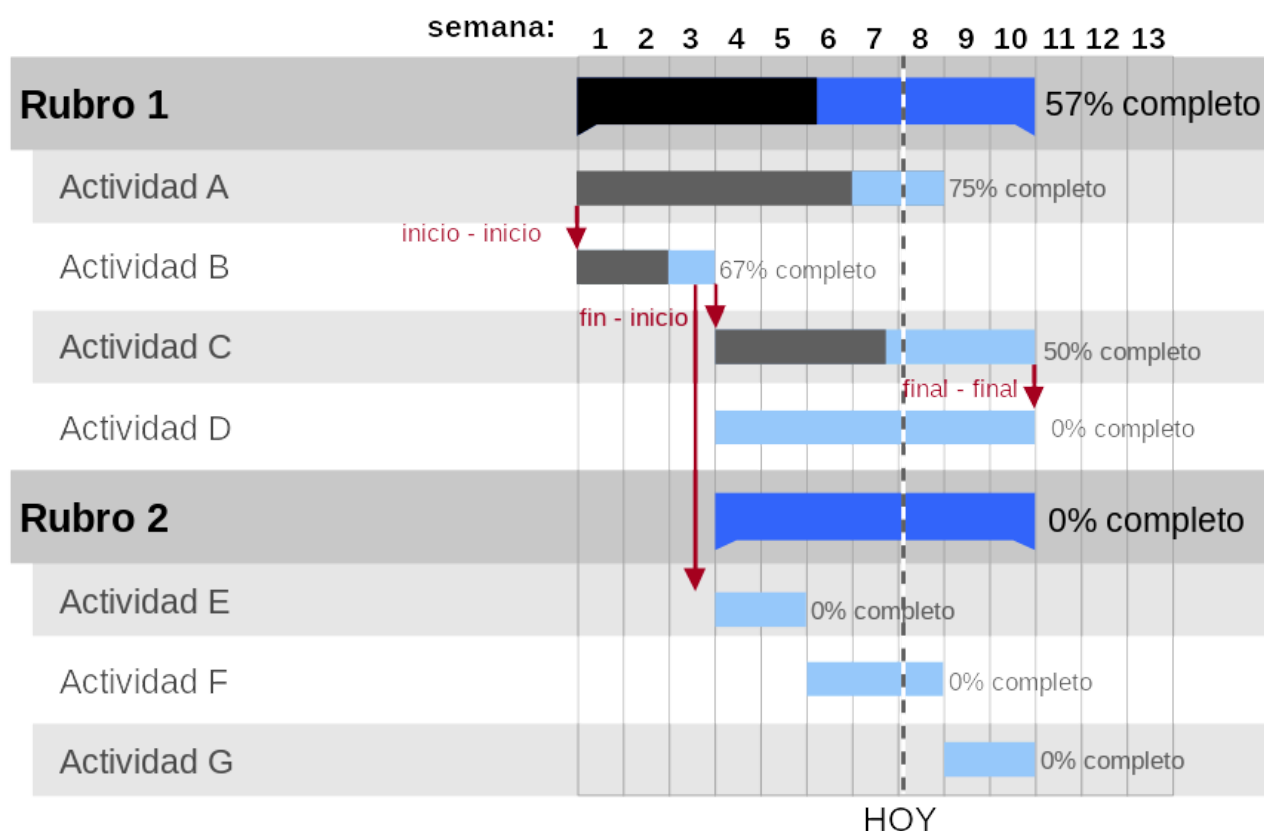


Figura 6.1: Diagrama de Gantt mostrando tres tipos de precedencias (en rojo) y porcentajes de avance.

Créditos Wikipedia<sup>1</sup>.

<sup>1</sup> De Garrybooker > Malyszkz > Mario Fèvre - traducción del archivo original en inglés en wikimedia commons.  
<http://en.wikipedia.org/wiki/File:GanttChartAnatomy.svg> (2011-12-19), CC0,  
<https://commons.wikimedia.org/w/index.php?curid=17728618>

# Principio de funcionamiento

En este capítulo se introducirá el funcionamiento del dispositivo. Se explicarán los rasgos principales para así entender el proceso que hay que seguir para hacer funcionar el producto. Más adelante se explicarán como se ha realizado cada etapa del proyecto, así como el porqué se ha trabajado de esta forma.

Es posible utilizar diferentes tipos de diagramas para ilustrar el principio de funcionamiento del sistema, por ejemplo un diagrama de bloques funcional o diagrama de bloques de procesos [3] es la representación gráfica de los diferentes procesos de un sistema y el flujo de señales donde cada proceso tiene un bloque asignado y estos se unen por flechas que representan el flujo de señales que interaccionan entre los diferentes procesos. Las entradas y salidas de los bloques se conectan entre sí con líneas de conexión o enlaces. Las líneas sencillas se pueden utilizar para conectar dos puntos lógicos del diagrama, es decir:

- Una variable de entrada y una entrada de un bloque.
- Una salida de un bloque y una entrada de otro bloque.
- Una salida de un bloque y una variable de salida

Se muestran las relaciones existentes entre los procesos y el flujo de señales de forma más realista que una representación matemática. Del mismo modo, tiene información relacionada con el comportamiento dinámico y no incluye información de la construcción física del sistema.

Por ejemplo: La función principal del dispositivo es poder realizar ..... El principio de funcionamiento del dispositivo es representado en la figura 7.1:

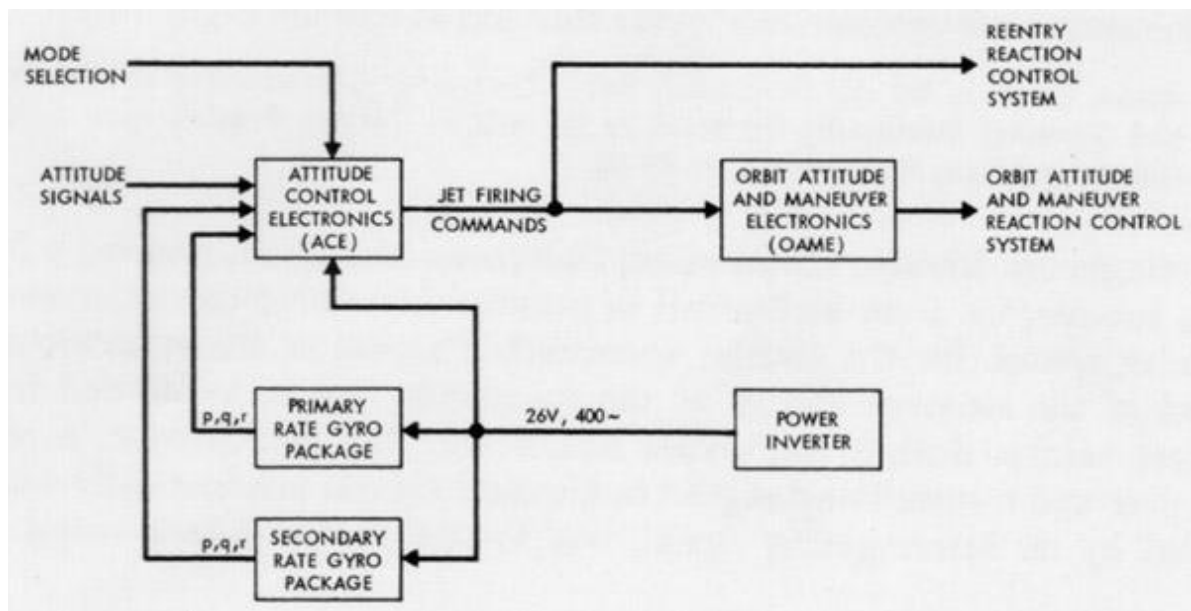


Figura 7.1: Diagrama de bloques funcional del control de altitud y sistema electrónico de maniobras de la nave espacial Gemini, junio de 1962. Créditos Wikipedia<sup>2</sup>.

<sup>2</sup> De McDonnell, "Project Gemini Familiarization Charts," - <http://history.nasa.gov/SP-4002/images/fig18.jpg>, see also Project Gemini Concept and Design January 1962 through December 1962, Dominio público, <https://commons.wikimedia.org/w/index.php?curid=4855322>

Se recomienda no utilizar **texto en negritas** en ningún párrafo, ni tampoco texto subrayado. En cambio, sí se sugiere utilizar *texto en cursiva* donde se considere apropiado.

Se sugiere que la escritura sea impersonal. Por ejemplo, no utilizar “el diseño del firmware lo hice de acuerdo con tal principio”, sino “el firmware fue diseñado utilizando tal principio”. En lo posible hablar en tiempo pasado, ya que la memoria describe un trabajo que ya fue realizado.

Se recomienda no utilizar una sección de glosario, sino colocar la descripción de las abreviaturas como parte del mismo cuerpo del texto. Por ejemplo, RTOS (Real Time Operating System, Sistema Operativo de Tiempo Real) o en caso de considerarlo apropiado mediante notas a pie de página<sup>3</sup>.

Si se desea indicar alguna página web, utilizar el siguiente formato de referencias bibliográficas, dónde las referencias se detallan en la sección de bibliografía de la memoria, utilizando el formato establecido por IEEE en [4]. Por ejemplo, “el presente trabajo se basa en la plataforma EDU-CIAA-NXP, la cual se describe en detalle en [5]”.

Al insertar figuras en la memoria se deben considerar determinadas pautas. Para empezar, usar siempre tipografía claramente legible. Luego, tener claro que es incorrecto escribir por ejemplo esto: “El diseño elegido es un cuadrado, como se ve en la siguiente figura:”



La forma correcta de utilizar una figura es la siguiente: “Se eligió utilizar un cuadrado azul para el logo, el cual se ilustra en la figura 7.2”.



Figura 7.2: Ilustración del cuadrado azul que se eligió para el diseño del logo.

El texto de las figuras debe estar siempre en español, excepto que se decida reproducir una figura original tomada de alguna referencia. En ese caso, la referencia de la cual se tomó la figura debe ser indicada en el epígrafe de la figura e incluida como una nota al pie, como se ilustra en la figura 7.3.

<sup>3</sup> Como esta nota, por ejemplo.



English word

Figura 7.3: Imagen ilustrativa de tal cosa, tomada de la página oficial del fabricante del procesador<sup>4</sup>.

La figura y el epígrafe deben conformar una unidad cuyo significado principal pueda ser comprendido por el lector sin necesidad de leer el cuerpo central de la memoria. Para eso es necesario que el epígrafe sea todo lo detallado que corresponda y si en la figura se utilizan abreviaturas, entonces aclarar su significado en el epígrafe o en la misma figura.



Figura 7.4: El lector no sabe por qué de pronto aparece esta figura.

Nunca colocar una figura en el documento antes de hacer la primera referencia a ella, como se ilustra con la figura 7.4, porque, sino el lector no comprenderá por qué de pronto aparece la figura en el documento, lo que distraerá su atención.

Para las tablas utilizar el mismo formato que para las figuras, solo que el epígrafe se debe colocar arriba de la tabla, como se ilustra en la tabla 7.1. Observar el uso de las negritas para los encabezados.

Tabla 7.1: Descripción de pines sensor de temperatura DS18B20.

PIN			NAME	FUNCTION
SO	μSOP	TO-92		
1, 2, 6, 7, 8	2, 3, 5, 6, 7	-	N.C.	No Connection.
5	4	1	GND	Ground.

En cada capítulo se debe reiniciar el número de conteos de las figuras y las tablas, por ejemplo, Figura 2.1 o Tabla 2.1, pero no se debe reiniciar el conteo en cada sección.

<sup>4</sup> ARM (2016, Jun 26). ARM Cores [en línea]. Available: <https://www.arm.com/>.

Al insertar ecuaciones en la memoria, estas se deben numerar de la siguiente forma:

$$x \geq y \qquad \qquad \qquad \text{(Ecuación 7.1)}$$

Es importante tener presente que en el caso de las ecuaciones estas pueden ser referidas por su número, como por ejemplo “tal como describe la ecuación (7.1)”, pero también es correcto utilizar los dos puntos, como por ejemplo “la expresión matemática que describe este comportamiento es la siguiente:”

$$x \geq y \qquad \qquad \qquad \text{(Ecuación 7.2)}$$

Para las ecuaciones se debe utilizar un tamaño de letra equivalente al utilizado para el texto del trabajo, en tipografía cursiva y preferentemente del tipo Times New Roman o similar. El espaciado antes y después de cada ecuación es de aproximadamente el doble que entre párrafos consecutivos del cuerpo principal del texto.

# Componentes principales

Esta sección presenta los componentes principales empleados para llevar a cabo el proyecto. En él se hará una descripción general de cada elemento y se expondrán las razones por las cuales se han escogido.

## 8.1. Placa Arduino Uno

Por ejemplo: El Arduino Uno [6] es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc. La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos.

La placa tiene 14 pines digitales, 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B. Puede ser alimentado por el cable USB o por una batería externa de 9 voltios, aunque acepta voltajes entre 7 y 20 voltios. También es similar al Arduino Nano y Leonardo.

El diseño de referencia de hardware se distribuye bajo una licencia Creative Commons Attribution Share-Alike 2.5 y está disponible en el sitio web de Arduino. Los archivos de diseño y producción para algunas versiones del hardware también están disponibles. Se puede apreciar en la figura 8.1. una imagen pictórica de la placa.

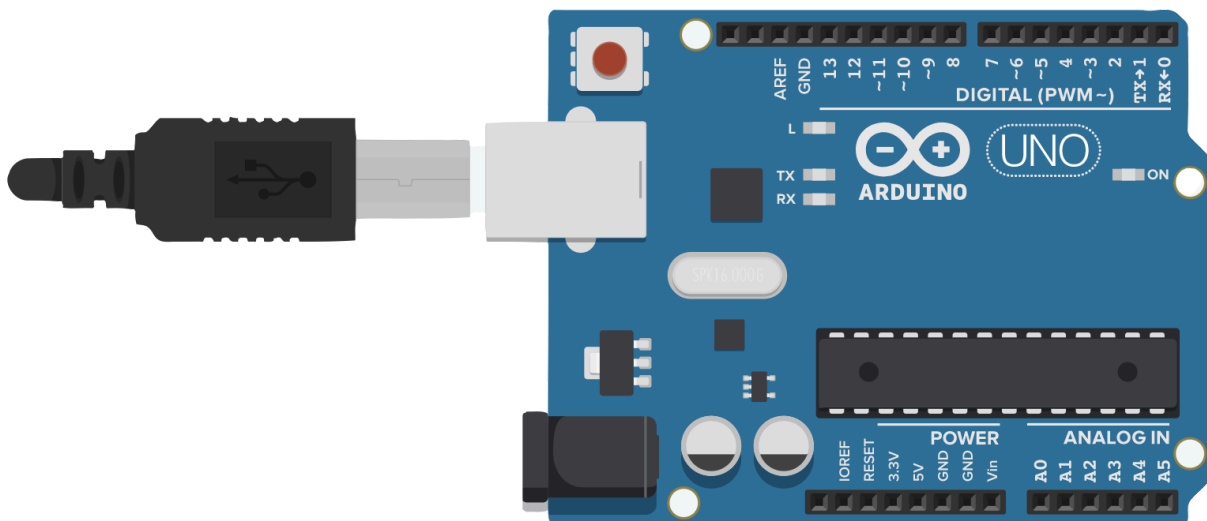


Figura 8.1: Arduino Uno. Créditos Maxbrothers2020<sup>5</sup>.

<sup>5</sup> De Maxbrothers2020 - Trabajo propio, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=101308738>.



## 8.2. Display de siete segmentos

Por ejemplo: Para la representación de caracteres del dispositivo se ha optado por emplear displays de siete segmentos, ya que son los que representan de forma más clara los caracteres numéricos.

Debido a que el dispositivo tendrá que ser utilizado en una sala amplia, se le exigirá tener unas dimensiones razonables para dicho fin. Finalmente, se han elegido displays de siete segmentos de la marca Kingbright, con unas dimensiones de 90 x 122 mm y con conexión en ánodo común.

A continuación, en la figura 8.2. se ilustra el diagrama del componente junto con las conexiones que presenta:

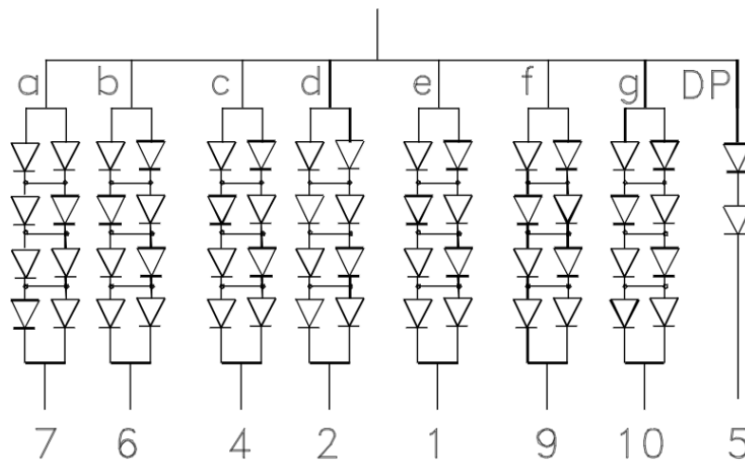


Figura 8.2: Distribución interna de cada segmento del display. Créditos: Kingbright<sup>6</sup>.

Por ejemplo: Para poder solucionar esta problemática, dado que la fuente de alimentación empleada es de 9 V, se conectará una resistencia en serie al terminal del punto decimal con el fin de adaptar el valor de la tensión al valor deseado. Aplicando la Ley de Ohm se ha obtenido el valor de la resistencia:

$$R = \frac{9V - 3,7V}{0,03mA} = 176,67\Omega \quad (\text{Ecuación 8.1})$$

## 8.3. Parlante

Por ejemplo: Hay que tener en cuenta que las salidas de una placa Arduino Uno están limitadas tanto en tensión como en intensidad, siendo esta última de un valor máximo de 40 mA aproximadamente.

Si consideramos que el parlante escogido consume una potencia de 0,5 W y tiene una impedancia de 8  $\Omega$  significaría que este consume una corriente de:

$$P = R * I^2 \quad (\text{Ecuación 8.2})$$

<sup>6</sup> <https://www.kingbrightusa.com/>

Para solventar esta problemática debemos añadir una etapa de amplificación empleando transistores BJT (Bipolar Junction Transistor, Transistor de Unión Bipolar), que permiten amplificar la señal eléctrica y de esta manera obtener la intensidad que deseamos.

Dependiendo del tipo de componente, probablemente sea necesario ilustrar su funcionamiento con un diagrama de tiempos, tal cual ilustra la figura 8.3:

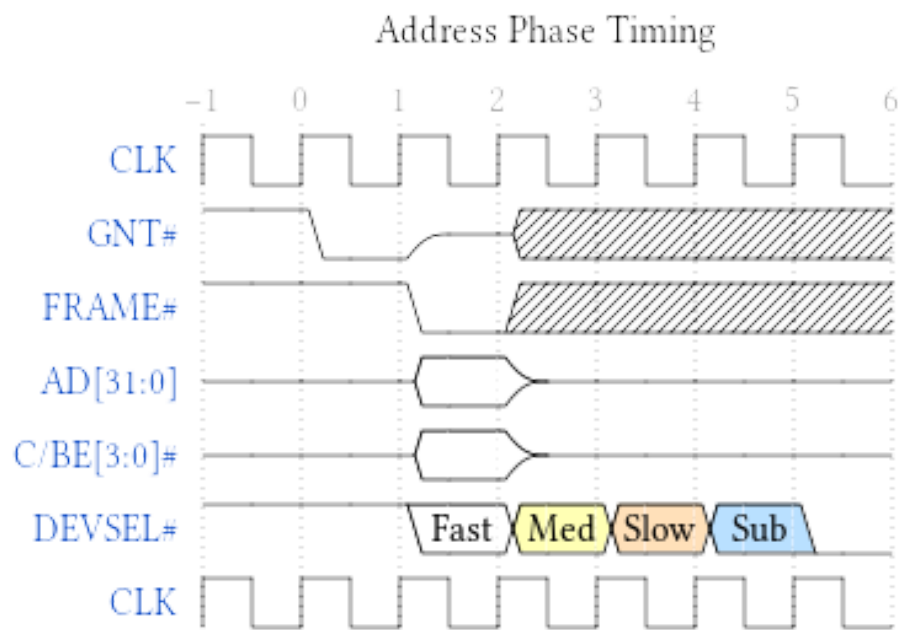


Figura 8.3: AddressPhaseTiming Diagram done in Wavedrom. Créditos: Maorm7811<sup>7</sup>.

<sup>7</sup> By Maorm7811 - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=125557078>.

# Circuitos

En esta sección se debe representar el conexionado de los componentes utilizando la simbología adecuada. Por ejemplo: El circuito resultante se muestra en la figura 9.1:

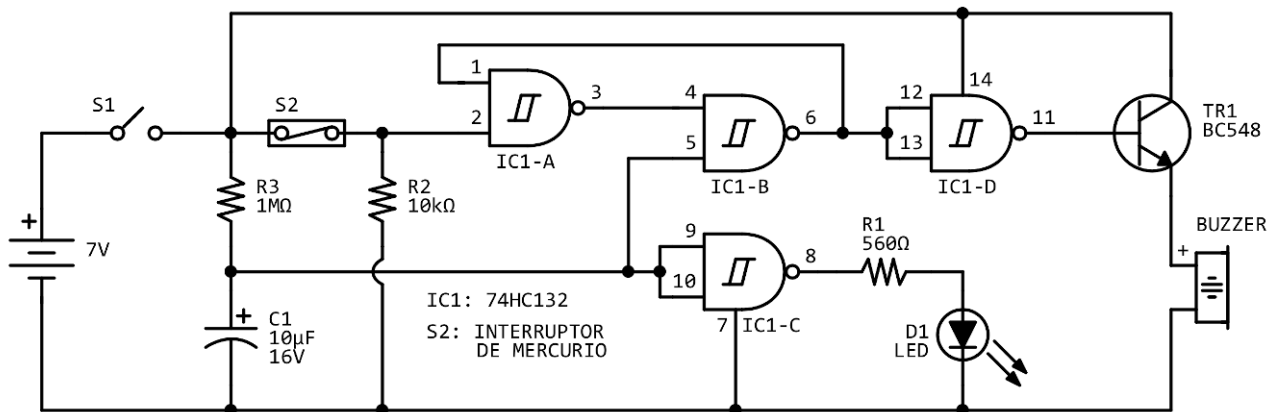


Figura 9.1: Diagrama esquemático del circuito propuesto. Créditos: producción propia.

También es posible utilizar diagramas pictóricos para representar el conexionado de los diferentes componentes, como ilustra la figura 9.2:

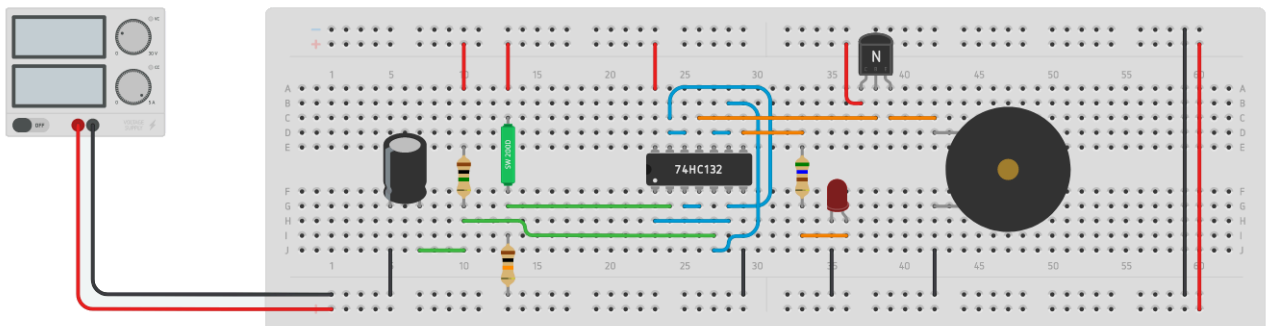


Figura 9.2: Diagrama pictórico del circuito propuesto. Créditos: producción propia.

# Software

Este capítulo describe el software necesario para controlar el sistema. En este ejemplo la arquitectura de la aplicación, figura 10.1, apunta a la claridad y, por tanto, también a la simplicidad; no pretende ser un código de producción, supongamos un microcontrolador con I<sup>2</sup>C y GPIO.

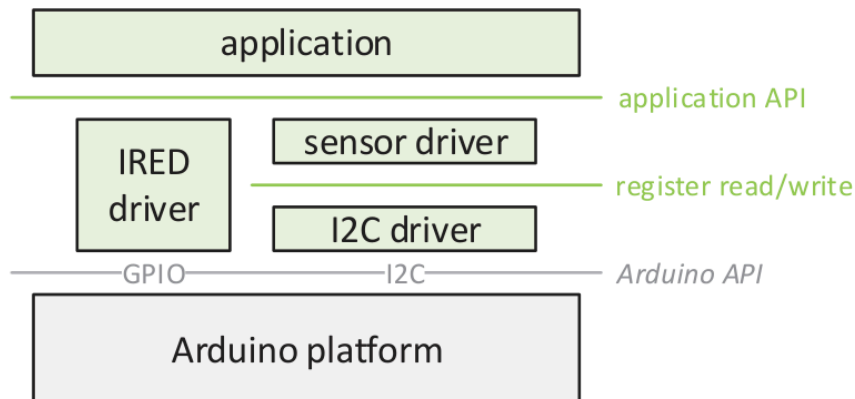


Figura 10.1: Arquitectura de software. Créditos: ams-OSRAM AG.

Se deberá documentar el proceso de diseño, implementación y prueba del algoritmo del dispositivo utilizando, por ejemplo, pseudolenguaje [7], diagramas de flujo [8] y fragmentos de código. Es importante explicar el flujo de los datos y los diferentes procesos que atraviesan.

Por ejemplo, pseudocódigo estilo Lenguaje C:

```
subproceso funcion bizzbuzz
para (i <- 1; i<=100; i++) {
    establecer print_number a verdadero;
    Si i es divisible por 3
        escribir "Bizz";
        establecer print_number a falso;
    Si i es divisible por 5
        escribir "Buzz";
        establecer print_number a falso;
    Si print_number, escribir i;
    escribir una nueva línea;
}
```

Por ejemplo, diagrama de flujo de una estructura condicional:

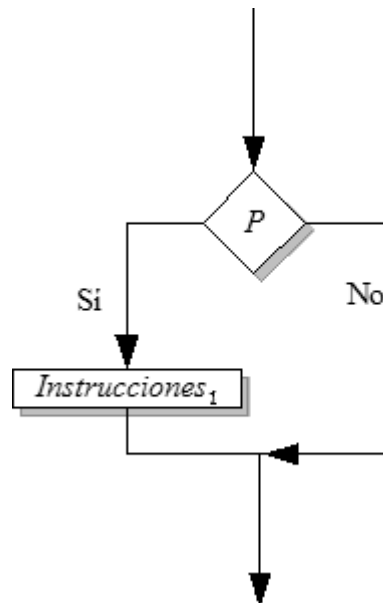


Figura 10.2: Diagrama de flujo que muestra el funcionamiento de la instrucción condicional. Créditos: Jesuja<sup>8</sup>.

## 10.1. Función principal void loop()

Es importante referenciar las instrucciones dispuestas en líneas a un número. Por ejemplo: La función de la línea n.º 32 realiza ..... según el siguiente fragmento de código codificado en C++:

```
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off
36   delay(1000); // wait for a second
37 }
```

## 10.2. Librería LiquidCrystal

En el caso de utilizar librerías de terceros, es necesario hacer referencias a ellas y al autor. Además, explicar las funciones utilizadas o que así lo requieran. Por ejemplo:

La librería LiquidCrystal de Arduino<sup>9</sup> permite la comunicación con pantallas alfanuméricas de cristal líquido (LCD).

<sup>8</sup> De Jesuja - Trabajo propio, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=2845117>.

<sup>9</sup> Reference > Libraries > Liquidcrystal <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>

Esta biblioteca permite que una placa Arduino/Genuino controle pantallas LCD (Liquid-Crystal Display, Pantalla de Cristal Líquido) basadas en el conjunto de chips Hitachi HD44780 (o uno compatible), que se encuentra en la mayoría de las pantallas LCD basadas en texto. La biblioteca funciona en modo de 4 u 8 bits (es decir, usando 4 u 8 líneas de datos, además de las líneas de control rs, enable y, opcionalmente, rw).

Para usar esta biblioteca deberá incluir la siguiente línea:

```
#include <LiquidCrystal.h>
```

# Presupuesto

Importante realizar un registro de los costos involucrados en la creación del dispositivo. En caso de discriminar algún componente, realizar la justificación. Los precios deberán poseer una fuente o referencia del mercado local, e indicar si se aplica algún descuento.

Por ejemplo: Los precios de cada componente se encuentran listados en la tabla 11.1:

CÓMPUTO Y PRESUPUESTO DE MATERIALES					
N°	Concepto	Cantidad	Costo Unitario	IVA 21%	Total
1	Arduino Mega 2560 R3	1	3946,71	828,81	4775,52
2	Sensor DS18B20	2	880,63	184,93	2131,12
3	Display LCD1602 I2C	1	1829,88	384,27	2214,15
4	Tarjeta SD 32GB	1	1400,00	294	1694
5	Pila CR1220	1	150,00	31,5	181,5
6	Alambre de estaño (100g)	1	700,00	147	847
Costo Total (AR\$):					11843,29

Tabla 11.1: Coste de los componentes utilizados para el desarrollo del proyecto.

Por ejemplo: En la Tabla 11.2 se muestra el presupuesto relacionado con la mano de obra:

CÓMPUTO Y PRESUPUESTO DE MANO DE OBRA				
N°	Asignado a	Tarea	Descripción	Costo
1	Perez, José	Instalación	Realización de instalación eléctrica.	1000,00
2	Gonzales, Jorge	Programación	Programación de PLC.	2000,00
3	Gimenez, Matias	Pruebas	Pruebas del sistema.	3000,00
Costo Total (AR\$):				6000,00

Tabla 11.2: Discriminación de roles, tareas y costo de mano de obra.

# Conclusiones

La idea de esta sección es resaltar cuáles son los principales aportes del trabajo realizado y cómo se podría continuar. Debe ser especialmente breve y concisa. Es buena idea usar un listado para enumerar los logros obtenidos.

## Próximos pasos

Acá se indica cómo se podría continuar el trabajo más adelante.



# Agradecimientos

Acá van los agradecimientos, es la sección más personal de la memoria.

# Bibliografía

Las referencias bibliográficas se pueden citar utilizando diferentes tipos de normas. En este texto sugerimos la norma *ISO 690* [9] que proporciona las directrices básicas para la preparación de referencias bibliográficas de materiales publicados, como monografías y publicaciones seriadas, capítulos, artículos de publicaciones seriadas (como revistas y diarios), recursos electrónicos, materiales cartográficos, grabaciones sonoras, fotografías, obras audiovisuales y documentos de patentes. Abarca las referencias a materiales publicados en forma impresa o no impresa. Sin embargo, no se aplica a referencias de manuscritos u otros materiales no publicados, ni a citas legales.

Aunque propone un orden prescrito para los elementos de la referencia y establece convenciones para la transcripción y presentación de información derivada de la publicación fuente, no prescribe ni recomienda un estilo concreto de referencia o de cita. El estilo y la puntuación que muestran los ejemplos incluidos en la norma no se consideran prescriptivos, sino simples ejemplos de usos conformes a la norma.

Otra posibilidad es utilizar las *Normas APA (7ª edición)* para elaborar los tipos básicos de referencias, y las referencias a material consultado en Internet:

1. Libros. - Autor/a (apellido –solo la primera letra en mayúscula–, coma, inicial de nombre y punto; en caso de varios autores/as, se separan con coma y antes del último con una "y"), año (entre paréntesis) y punto, título completo (en letra cursiva) y punto; ciudad y dos puntos, editorial.
  - Ejemplo: Tyrer, P. (1989). *Classification of Neurosis*. London: Wiley
2. Capítulos de libros. - Autores/as y año (en la forma indicada anteriormente); título del capítulo, punto; "En"; nombre de los autores/as del libro (inicial, punto, apellido); "(Eds.)", o "(Drs.)", o "(Comps.)"; título del libro en cursiva; páginas que ocupa el capítulo, entre paréntesis, punto; ciudad, dos puntos, editorial.
  - Ejemplo: Singer, M. (1994). Discourse inference processes. En M. Gernsbacher (Ed.), *Handbook of Psycholinguistics* (pp. 459-516). New York: Academic Press.
3. Artículos de revista. - Autores/as y año (como en todos los casos); título del artículo, punto; nombre de la revista completo y en cursiva, coma; volumen en cursiva; número entre paréntesis y pegado al volumen (no hay espacio entre volumen y número); coma, página inicial, guion, página final, punto.
  - Ejemplo: Gutiérrez Calvo, M. y Eysenck, M.W. (1995). Sesgo interpretativo en la ansiedad de evaluación. *Ansiedad y Estrés*, 1(1), 5-20.
4. Material consultado en Internet. Estas referencias deben proveer, al menos, el título del recurso, fecha de publicación o fecha de acceso, y la dirección (URL) del recurso en el Web. Formato básico Autor/a de la página. (Fecha de publicación o revisión de la página, si está disponible). Título de la página o lugar. Recuperado (Fecha de acceso), de (URL-dirección).
  - Ejemplo: Suñol, J. (2001). Rejuvenecimiento facial. Recuperado el 12 de junio de 2001, de <http://drsunol.com>.

## Referencias bibliográficas

- [1] Colaboradores de Wikipedia. Derecho de autor [en línea]. Wikipedia, La enciclopedia libre, 2023 [fecha de consulta: 8 de diciembre del 2023]. Disponible en <[https://es.wikipedia.org/w/index.php?title=Derecho\\_de\\_autor&oldid=155881425](https://es.wikipedia.org/w/index.php?title=Derecho_de_autor&oldid=155881425)>.
- [2] Colaboradores de Wikipedia. Diagrama de Gantt [en línea]. Wikipedia, La enciclopedia libre, 2022 [fecha de consulta: 24 de enero del 2023]. Disponible en <[https://es.wikipedia.org/w/index.php?title=Diagrama\\_de\\_Gantt&oldid=147156828](https://es.wikipedia.org/w/index.php?title=Diagrama_de_Gantt&oldid=147156828)>.
- [3] Colaboradores de Wikipedia. Diagrama de bloques funcional [en línea]. Wikipedia, La enciclopedia libre, 2022 [fecha de consulta: 24 de enero del 2023]. Disponible en <[https://es.wikipedia.org/w/index.php?title=Diagrama\\_de\\_bloques\\_funcional&oldid=141197037](https://es.wikipedia.org/w/index.php?title=Diagrama_de_bloques_funcional&oldid=141197037)>.
- [4] IEEE (2016, Jun 26). IEEE Citation Reference [en línea]. Available: <http://www.ieee.org/documents/ieeecitationref.pdf>.
- [5] Proyecto CIAA (2016, Jun 26). EDU-CIAA-NXP [en línea]. Available: <http://www.proyecto-ciaa.com.ar/devwiki/doku.php>.
- [6] Colaboradores de Wikipedia. Arduino Uno [en línea]. Wikipedia, La enciclopedia libre, 2022 [fecha de consulta: 24 de enero del 2023]. Disponible en <[https://es.wikipedia.org/w/index.php?title=Arduino\\_Uno&oldid=148031703](https://es.wikipedia.org/w/index.php?title=Arduino_Uno&oldid=148031703)>.
- [7] Colaboradores de Wikipedia. Pseudocódigo [en línea]. Wikipedia, La enciclopedia libre, 2023 [fecha de consulta: 24 de enero del 2023]. Disponible en <<https://es.wikipedia.org/w/index.php?title=Pseudoc%C3%B3digo&oldid=148514209>>.
- [8] Colaboradores de Wikipedia. Diagrama de flujo [en línea]. Wikipedia, La enciclopedia libre, 2022 [fecha de consulta: 24 de enero del 2023]. Disponible en <[https://es.wikipedia.org/w/index.php?title=Diagrama\\_de\\_flujo&oldid=147386318](https://es.wikipedia.org/w/index.php?title=Diagrama_de_flujo&oldid=147386318)>.
- [9] Colaboradores de Wikipedia. ISO 690 [en línea]. Wikipedia, La enciclopedia libre, 2022 [fecha de consulta: 24 de enero del 2023]. Disponible en <[https://es.wikipedia.org/w/index.php?title=ISO\\_690&oldid=146221431](https://es.wikipedia.org/w/index.php?title=ISO_690&oldid=146221431)>.

# Anexo I

## Código del proyecto

```
1 /*
2   Blink
3
4   Turns an LED on for one second, then off for one second, repeatedly.
5
6   Most Arduinos have an on-board LED you can control. On the UNO, MEGA and
7   ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is
8   set to the correct LED pin independent of which board is used. If you want
9   to know what pin the on-board LED is connected to on your Arduino model,
10  check the Technical Specs of your board at:
11  https://www.arduino.cc/en/Main/Products
12
13  modified 8 May 2014
14  by Scott Fitzgerald
15  modified 2 Sep 2016
16  by Arturo Guadalupi
17  modified 8 Sep 2016
18  by Colby Newman
19
20  This example code is in the public domain.
21
22  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
23 */
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
34   delay(1000);                     // wait for a second
35   digitalWrite(LED_BUILTIN, LOW);  // turn the LED off
36   delay(1000);                     // wait for a second
37 }
```

# Anexo II

## Hoja de datos del sensor DS18B20

Click [here](#) for production status of specific part numbers.

### DS18B20

### Programmable Resolution 1-Wire Digital Thermometer

#### General Description

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

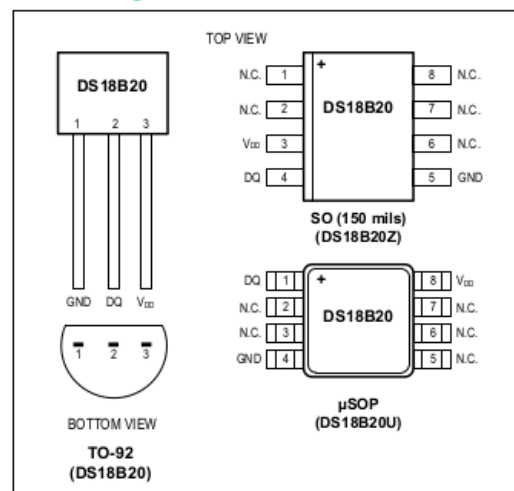
#### Applications

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

#### Benefits and Features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Reduce Component Count with Integrated Temperature Sensor and EEPROM
  - Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
  - ±0.5°C Accuracy from -10°C to +85°C
  - Programmable Resolution from 9 Bits to 12 Bits
  - No External Components Required
- Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability
  - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits
- Available in 8-Pin SO (150 mils), 8-Pin  $\mu$ SOP, and 3-Pin TO-92 Packages

#### Pin Configurations



Ordering Information appears at end of data sheet.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

# Anexo III

## Listado de software útil

### 1. Edición:

- **LibreOffice**, suite ofimática. <https://www.libreoffice.org>
- **GIMP**, editor de imágenes. <https://www.gimp.org>
- **Inkscape**, gráficos vectoriales. <https://inkscape.org>
- **PDF Arranger**, edición de archivos pdf. <https://github.com/pdfarranger/pdfarranger>
- **Impress.js**, presentaciones en el navegador. <https://impress.js.org>
- **Online Gantt Chart**, diagramas de Gantt. <https://www.onlinegantt.com>

### 2. Programación:

- **PSeInt**, pseudo lenguaje y diagramas de flujo. <https://pseint.sourceforge.net>
- **Dia**, editor de diagramas. <http://dia-installer.de>
- **Arduino**, API de desarrollo. <https://www.arduino.cc>
- **Visual Studio Code**, entorno de desarrollo. <https://code.visualstudio.com>
- **Thonny**, entorno de desarrollo Python. <https://thonny.org>
- **Zinjal**, entorno de desarrollo C++. <https://zinjai.sourceforge.net>

### 3. Electrónica:

- **Wokwi**, simulador en línea. <https://wokwi.com>
- **Tinkercad**, simulador en línea. <https://www.tinkercad.com>
- **SimulIDE**, simulador de circuitos. <https://simulide.com>
- **WaveDrom**, diagramas de tiempo digitales. <https://wavedrom.com>
- **FidoCadJ**, editor para electrónica. <https://darwinne.github.io/FidoCadJ>
- **KiCad EDA**, suite de diseño electrónico. <https://www.kicad.org>
- **Autodesk EAGLE**, suite de diseño electrónico. <http://eagle.autodesk.com>
- **QElectroTech**, editor de diagramas eléctricos. <https://qelectrotech.org>
- **DIY LC**, editor para prototipado. <http://bancika.github.io/diy-layout-creator>
- **Fritzing**, suite de diseño electrónico. <https://fritzing.org>

## Anexo IV

### Licencias

Para más información sobre derechos de autor, referirse al artículo Derecho de autor en Wikipedia [1]. Sobre licencias Creative Commons, consulte su sitio web<sup>10</sup>.



Figura IV.0: Mapa de licencias. Créditos: Dario Badagnani (Vamox)<sup>11</sup>.

<sup>10</sup> <https://creativecommons.org/>

<sup>11</sup> <http://vamox.blogspot.com/search/label/licencias>