CARRERA DE ESPECIALIZACIÓN EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

Controlador CAN de servomotores

Autor:
Ing. Alejandro Virgillo

Director:
Esp. Ing. Gabriel Gavinowich (FIUBA)

Jurados:
Nombre del jurado 1 (pertenencia)
Nombre del jurado 2 (pertenencia)
Nombre del jurado 3 (pertenencia)

Este trabajo fue realizado en la Ciudad Autónoma de Buenos Aires, entre Octubre de 2021 y Abril de 2023.

Resumen

En el presente trabajo se aborda el proceso de planificación, diseño y fabricación de un dispositivo que actúe de interfaz para que un operario de planta pueda monitorear y programar motores paso a paso que usan una plaqueta de control a medida conectados a través de un bus CAN (Controller Area Network). El desarrollo se plantea para uso industrial y se ensaya en la planta de la empresa argentina Cambre ICyFSA.

El documento detalla las herramientas de planificación y gestión empleadas, la selección de componentes electrónicos, la descripción de los protocolos de comuncación empleados, el desarrollo del software embebido, el diseño y fabricación del hardware y su gabinete y las consideraciones de manufactura que se tomaron. Todo esto acompañado con su documentación apropiada y con las pruebas de validación realizadas.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción general

En esta sección se da una explicación general de los temas sobre los cuales se basa este trabajo. Se dará una introducción al protocolo CAN - *Controller Area Network* y a qué es un servomotor. También se hablará sobre el proyecto SN-17 y la motivación y alcance del trabajo, así como el estado del arte de esta tecnología.

1.1. Controller Area Network - CAN

Controller Area Network o CAN es un protocolo de comunicación desarrollado por Bosch [**wikipedia_CAN**] orientado originalmente a la industria automotriz, y hoy en día es empleado en muchas otras aplicaciones. En el año 1991 Bosch publicó la especificación CAN 2.0 donde se diferencian aspectos de identificación de dispositivos conectados en la red y, en el año 1993, se estandarizó el protocolo bajo la norma internacional ISO 11898 [**web_ISO_CAN**]. CAN se caracteriza por su robustez y bajos requerimientos de cableado. En su concepción, se pensó para proveer comunicaciones deterministicas en sistemas complejos, teniendo las siguientes cualidades [**Understanding_CAN**]:

- Prioridad de mensajes y latencia máxima asegurada.
- Comunicaciones a varios dispositivos al mismo tiempo.
- Bus multi-maestro.
- Detección de errores en nodos y mensajes.
- Protocolo asincrónico - sin línea de clock.

El estándar abarca solamente las capas física y de enlace de datos, es decir, las 2 capas más bajas en un modelo OSI - *Open System Interconection*, como el que se puede ver en la Figura 1.1. Para las capas superiores, existen otros estándares, como CANOpen[1], del cual se tomaron ideas para este proyecto, pero no se implementa en si.

Para realizar la transferencia de información CAN emplea líneas de transmisión, comunmente llamados CAN High y CAN Low, y estos son los únicos enlaces requeridos. Al comienzo de cada mensaje, se emplea un identificador, que indica la prioridad del mensaje, así como a quien está dirigido. Usando esta información cada nodo de la red determina que hacer. En la Figura 1.2, se puede ver un esquema básico de una red CAN.

[2] https://www.can-cia.org/canopen/

[2] https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/

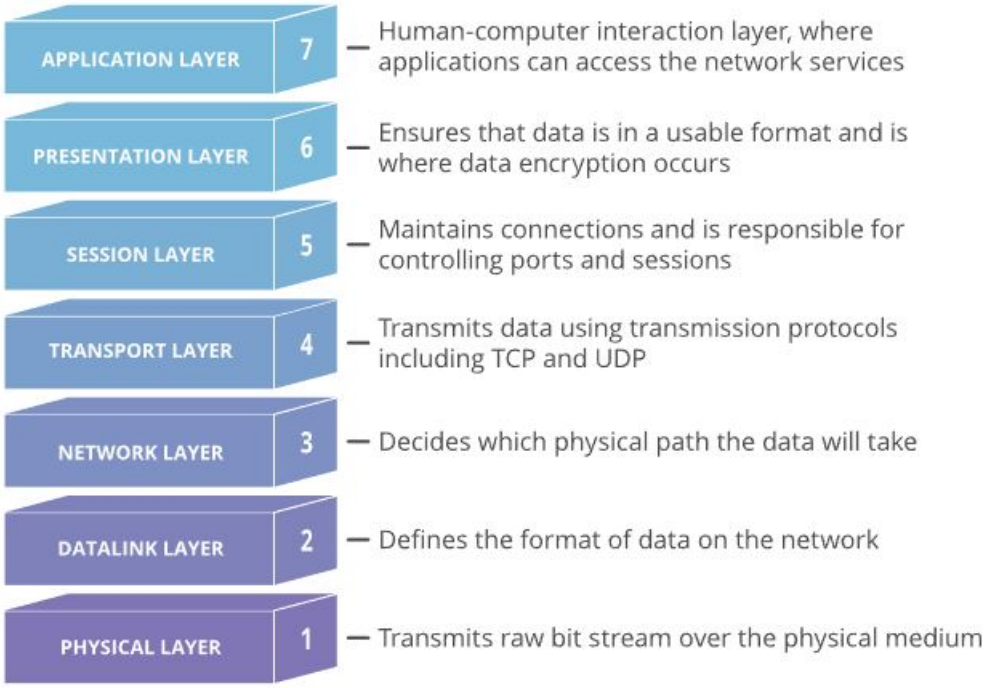


FIGURA 1.1. Modelo de capas OSI[2]

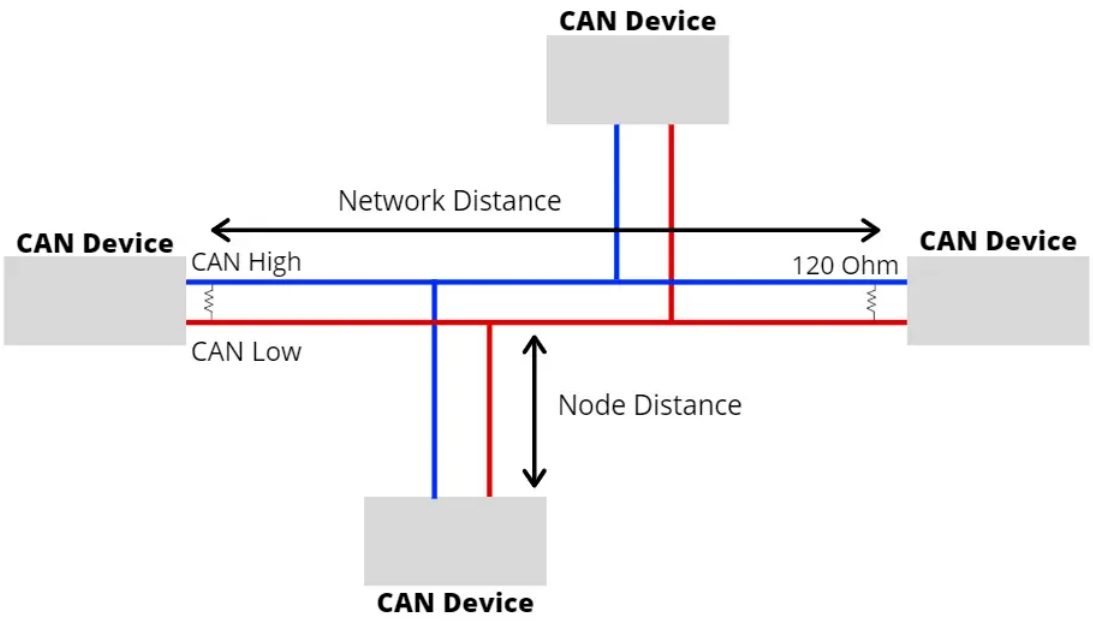


FIGURA 1.2. Esquema de red CAN[3]

1.2. Servomotores

Un servomotor es un tipo de motor eléctrico que tiene la capacidad de controlar la posición angular del eje, así como la velocidad de rotación y el torque. En general, el tipo de motor eléctrico que se emplee no determina si es o no un servomotor, sino que cuente las cualidades de control mencionadas. Para lograr esto, suele emplearse un sistema de lazo cerrado que se retroalimenta con la información proporcionada por un sensor que mide la posición angular del eje, llamado encoder. Esto es procesado por un controlador, quien se encarga de proporcionar la corriente adecuada a las bobinas del motor para que este se mueva de la forma indicada. En la Figura 1.3 se puede ver un servomotor con sus distintas partes.

FIGURA 1.3. Esquema un servomotor[4]

1.3. Proyecto SN-17 - Servo Nema 17

El proyecto SN-17 es un sistema desarrollado por la organización A3 Engineering que permite convertir motores eléctricos del tipo paso a paso o *steppers* en servomotores. Este tipo de motores tienen, similar a un servomotor, la capacidad de controlar su posición y velocidad, pero con un lazo de control abierto. En caso de que haya perturbaciones al funcionamiento normal, el motor pierde el control de la posición y debe realizar una rutina de *homing* para recuperarlo. Esto puede traer problemas en ciertas aplicaciones de precisión. Además, son incapaces de entregar torque constante a la salida, por el mismo motivo. El sistema SN-17 agrega un encoder y un controlador al motor, generando un lazo cerrado y logrando las funcionalidades de un servomotor. En la Figura 1.4 se puede observar una placa de control SN-17. Esta se coloca en la parte trasera de un motor paso a paso y se lo conecta directamente.

Además de generar las funcionalidades mencionadas, el sistema SN-17 tiene señales discretas industriales que le permiten comunicarse a través de ese medio con controladores industriales (*Programmable Logic Controllers* o PLCs) o pueden

[3] https://www.seeedstudio.com/blog/2019/11/27/introduction-to-can-bus-and-how-to-use-it-with-arduino/

[4] https://www.logicbus.com.mx/blog/que-es-un-servo-motor/

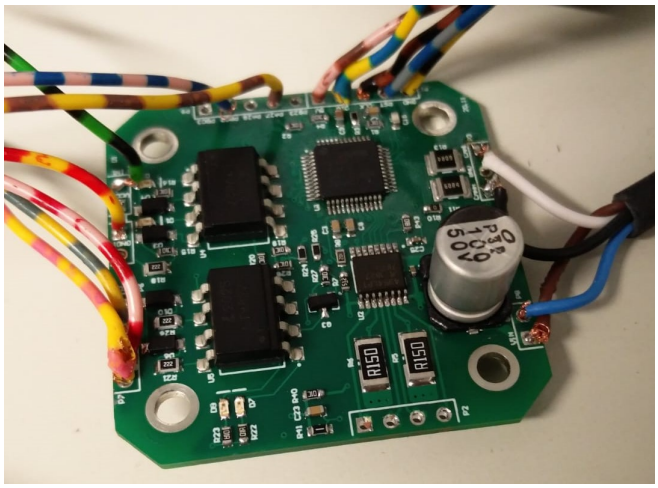


FIGURA 1.4. Plaqueta SN-17

controlar un pequeño proceso por su cuenta. Actualmente, se está trabajando en implementar estos sistemas en la planta de la empresa Cambre ICyFSA, donde se están construyendo distintos dispositivos para aplicaciones industriales con estos motores. En la Figura 1.5 se puede observar un dibujo CAD de un actuador lineal con un sistema SN-17 implementado. Este funciona como un prensador en un proceso industrial.

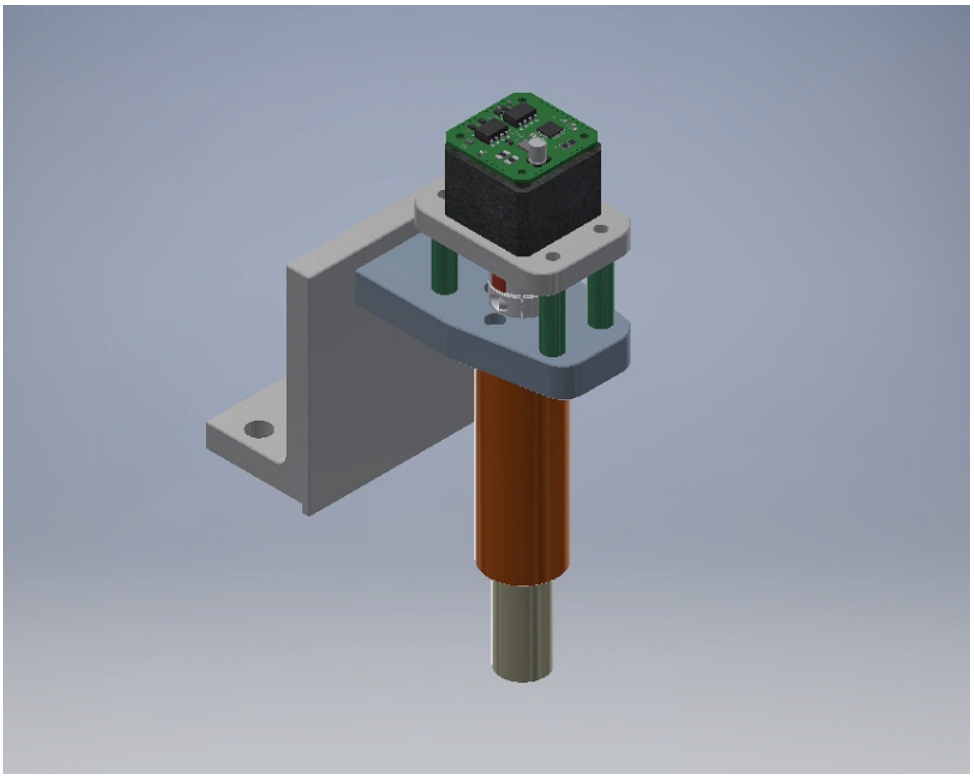


FIGURA 1.5. Actuador lineal con SN-17

1.4. Entorno de trabajo

Ante de comenzar a editar la plantilla debemos tener un editor LaTeX instalado en nuestra computadora. En forma análoga a lo que sucede en lenguaje C, que se puede crear y editar código con casi cualquier editor, existen ciertos entornos de trabajo que nos pueden simplificar mucho la tarea. En este sentido, se recomienda, sobre todo para los principiantes en LaTeX la utilización de TexMaker, un programa gratuito y multi-plantaforma que está disponible tanto para windows como para sistemas GNU/linux.

La versión más reciente de TexMaker es la 4.5 y se puede descargar del siguiente link: http://www.xm1math.net/texmaker/download.html. Se puede consultar el manual de usuario en el siguiente link: http://www.xm1math.net/texmaker/doc.html.

1.4.1. Paquetes adicionales

Si bien durante el proceso de instalación de TexMaker, o cualquier otro editor que se haya elegido, se instalarán en el sistema los paquetes básicos necesarios para trabajar con LaTeX, la plantilla de los trabajos de Especialización y Maestría requieren de paquete adicionales.

Se indican a continuación los comandos que se deben introducir en la consola de Ubuntu (ctrl + alt + t) para instalarlos:

```
$ sudo apt install texlive-lang-spanish texlive-science
$ sudo apt install texlive-bibtex-extra biber
$ sudo apt install texlive texlive-fonts-recommended
$ sudo apt install texlive-latex-extra
```

1.4.2. Configurando TexMaker

Una vez instalado el programa y los paquetes adicionales se debe abrir el archivo memoria.tex con el editor para ver una pantalla similar a la que se puede apreciar en la figura 1.6. Una vez instalado el programa y los paquetes adicionales se debe abrir el archivo memoria.tex con el editor para ver una pantalla similar a la que se puede apreciar en la figura 1.6. Una vez instalado el programa y los paquetes adicionales se debe abrir el archivo memoria.tex con el editor para ver una pantalla similar a la que se puede apreciar en la figura 1.6. Una vez instalado el programa y los paquetes adicionales se debe abrir el archivo memoria.tex con el editor para ver una pantalla similar a la que se puede apreciar en la figura 1.6.

Notar que existe una vista llamada Estructura a la izquierda de la interfaz que nos permite abrir desde dentro del programa los archivos individuales de los capítulos. A la derecha se encuentra una vista con el archivo propiamente dicho para su edición. Hacia la parte inferior se encuentra una vista del log con información de los resultados de la compilación. En esta última vista pueden aparecen advertencias o *warning*, que normalmente pueden ser ignorados, y los errores que se indican en color rojo y deben resolverse para que se genere el PDF de salida.

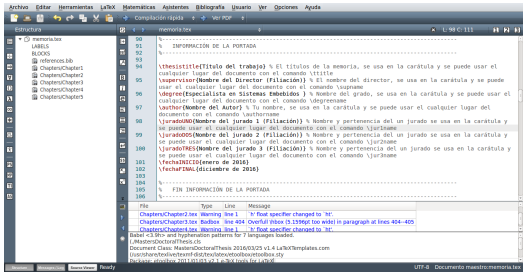


FIGURA 1.6. Entorno de trabajo de texMaker.

Recordar que el archivo que se debe compilar con PDFLaTeX es **`memoria.tex`**, si se tratara de compilar alguno de los capítulos saldría un error. Para salvar la molestia de tener que cambiar de archivo para compilar cada vez que se realice una modificación en un capítulo, se puede definir el archivo **`memoria.tex`** como "documento maestro" yendo al menú opciones ->"definir documento actual como documento maestro", lo que permite compilar con PDFLaTeX memoria.tex directamente desde cualquier archivo que se esté modificando . Se muestra esta opción en la figura 1.7.

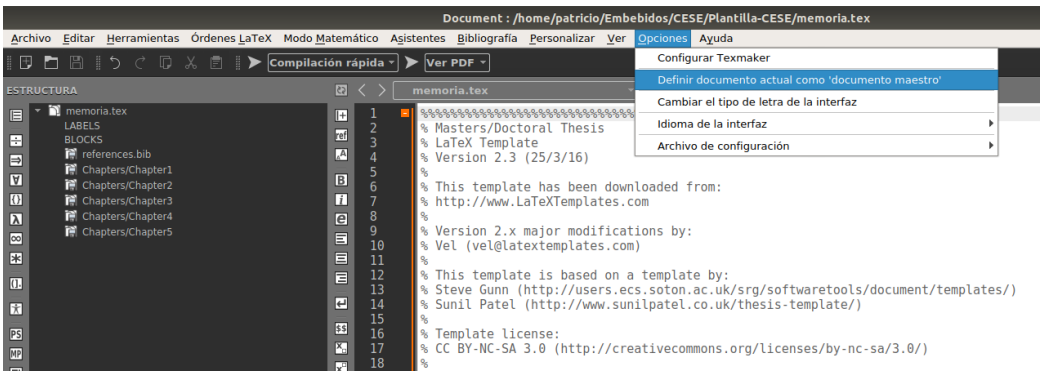


FIGURA 1.7. Definir memoria.tex como documento maestro.

En el menú herramientas se encuentran las opciones de compilación. Para producir un archivo PDF a partir de un archivo .tex se debe ejecutar PDFLaTeX (el shortcut es F6). Para incorporar nueva bibliografía se debe utilizar la opción BibTeX del mismo menú herramientas (el shortcut es F11).

Notar que para actualizar las tablas de contenidos se debe ejecutar PDFLaTeX dos veces. Esto se debe a que es necesario actualizar algunos archivos auxiliares antes de obtener el resultado final. En forma similar, para actualizar las referencias se debe ejecutar primero PDFLaTeX, después BibTeX y finalmente PDFLaTeX dos veces por idénticos motivos.

1.5. Personalizando la plantilla, el archivo `memoria.tex`

Para personalizar la plantilla se debe incorporar la información propia en los distintos archivos **`.tex`**.

Primero abrir **`memoria.tex`** con TexMaker (o el editor de su preferencia). Se debe ubicar dentro del archivo el bloque de código titulado *INFORMACIÓN DE LA PORTADA* donde se deben incorporar los primeros datos personales con los que se construirá automáticamente la portada.

1.6. El código del archivo memoria.tex explicado

El archivo **memoria.tex** contiene la estructura del documento y es el archivo de mayor jerarquía de la memoria. Podría ser equiparable a la función *main()* de un programa en C, o mejor dicho al archivo fuente .c donde se encuentra definida la función main().

La estructura básica de cualquier documento de LaTeX comienza con la definición de clase del documento, es seguida por un preámbulo donde se pueden agregar funcionalidades con el uso de `paquetes` (equiparables a bibliotecas de C), y finalmente, termina con el cuerpo del documento, donde irá el contenido de la memoria.

```
\documentclass{article}  <- Definicion de clase
\usepackage{listings}    <- Preambulo

\begin{document}         <- Comienzo del contenido propio
        Hello world!
\end{document}
```

El archivo **memoria.tex** se encuentra densamente comentado para explicar qué páginas, secciones y elementos de formato está creando el código LaTeX en cada línea. El código está dividido en bloques con nombres en mayúsculas para que resulte evidente qué es lo que hace esa porción de código en particular. Inicialmente puede parecer que hay mucho código LaTeX, pero es principalmente código para dar formato a la memoria por lo que no requiere intervención del usuario de la plantilla. Sí se deben personalizar con su información los bloques indicados como:

- Informacion de la memoria
- Resumen
- Agradecimientos
- Dedicatoria

El índice de contenidos, las listas de figura de tablas se generan en forma automática y no requieren intervención ni edición manual por parte del usuario de la plantilla.

En la parte final del documento se encuentran los capítulos y los apéndices. Por defecto se incluyen los 5 capítulos propuestos que se encuentran en la carpeta /Chapters. Cada capítulo se debe escribir en un archivo .tex separado y se debe poner en la carpeta *Chapters* con el nombre **Chapter1**, **Chapter2**, etc... El código para incluir capítulos desde archivos externos se muestra a continuación.

```
\include{Chapters/Chapter1}
\include{Chapters/Chapter2}
\include{Chapters/Chapter3}
\include{Chapters/Chapter4}
\include{Chapters/Chapter5}
```

Los apéndices también deben escribirse en archivos .tex separados, que se deben ubicar dentro de la carpeta *Appendices*. Los apéndices vienen comentados por defecto con el caracter `%` y para incluirlos simplemente se debe eliminar dicho caracter.

Finalmente, se encuentra el código para incluir la bibliografía en el documento final. Este código tampoco debe modificarse. La metodología para trabajar las referencias bibliográficas se desarrolla en la sección 1.7.

1.7. Bibliografía

Las opciones de formato de la bibliografía se controlan a través del paquete de latex *biblatex* que se incluye en la memoria en el archivo memoria.tex. Estas opciones determinan cómo se generan las citas bibliográficas en el cuerpo del documento y cómo se genera la bibliografía al final de la memoria.

En el preámbulo se puede encontrar el código que incluye el paquete biblatex, que no requiere ninguna modificación del usuario de la plantilla, y que contiene las siguientes opciones:

```
\usepackage[backend=bibtex,
        natbib=true,
        style=numeric,
        sorting=none]
{biblatex}
```

En el archivo **reference.bib** se encuentran las referencias bibliográficas que se pueden citar en el documento. Para incorporar una nueva cita al documento lo primero es agregarla en este archivo con todos los campos necesario. Todas las entradas bibliográficas comienzan con @ y una palabra que define el formato de la entrada. Para cada formato existen campos obligatorios que deben completarse. No importa el orden en que las entradas estén definidas en el archivo .bib. Tampoco es importante el orden en que estén definidos los campos de una entrada bibliográfica. A continuación se muestran algunos ejemplos:

```
@ARTICLE{ARTICLE:1,
    AUTHOR="John Doe",
    TITLE="Title",
    JOURNAL="Journal",
    YEAR="2017",
}

@BOOK{BOOK:1,
    AUTHOR="John Doe",
    TITLE="The Book without Title",
    PUBLISHER="Dummy Publisher",
    YEAR="2100",
}

@INBOOK{BOOK:2,
    AUTHOR="John Doe",
    TITLE="The Book without Title",
    PUBLISHER="Dummy Publisher",
    YEAR="2100",
    PAGES="100-200",
}

@MISC{WEBSITE:1,
    HOWPUBLISHED = "\url{http://example.com}",
    AUTHOR = "Intel",
    TITLE = "Example Website",
```

```
    MONTH = "12",
    YEAR = "1988",
    URLDATE = {2012-11-26}
}
```

Se debe notar que los nombres *ARTICLE:1*, *BOOK:1*, *BOOK:2* y *WEBSITE:1* son nombres de fantasía que le sirve al autor del documento para identificar la entrada. En este sentido, se podrían reemplazar por cualquier otro nombre. Tampoco es necesario poner : seguido de un número, en los ejemplos sólo se incluye como un posible estilo para identificar las entradas.

La entradas se citan en el documento con el comando:

```
\citep{nombre_de_la_entrada}
```

Y cuando se usan, se muestran así: [**ARTICLE:1**], [**BOOK:1**], [**BOOK:2**], [**WEBSITE:1**]. Notar cómo se conforma la sección Bibliografía al final del documento.

Capítulo 2

Introducción específica

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

2.1. Estilo y convenciones

2.1.1. Uso de mayúscula inicial para los título de secciones

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 2.1 se presenta lo que sea", o "En la subsección 2.1.2 se discute otra cosa".

Cuando se quiere poner una lista tabulada, se hace así:

- 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 este:

1. Este es el primer elemento de la lista.
2. Este es el segundo elemento de la lista.

Notar el uso de las mayúsculas y el punto al final de cada elemento.

2.1.2. Este es el título de una subsección

Se recomienda no utilizar **texto en negritas** en ningún párrafo, ni tampoco texto subrayado. En cambio sí se debe utilizar *texto en itálicas* para palabras en un idioma extranjero, al menos la primera vez que aparecen en el texto. En el caso de palabras que estamos inventando se deben utilizar "comillas", así como también para citas textuales. Por ejemplo, un *digital filter* es una especie de "selector" que permite separar ciertos componentes armónicos en particular.

La escritura debe ser impersonal. Por ejemplo, no utilizar "el diseño del firmware lo hice de acuerdo con tal principio", sino "el firmware fue diseñado utilizando tal principio".

El trabajo es algo que al momento de escribir la memoria se supone que ya está concluido, entonces todo lo que se refiera a hacer el trabajo se narra en tiempo pasado, porque es algo que ya ocurrió. Por ejemplo, "se diseñó el firmware empleando la técnica de test driven development".

En cambio, la memoria es algo que está vivo cada vez que el lector la lee. Por eso transcurre siempre en tiempo presente, como por ejemplo:

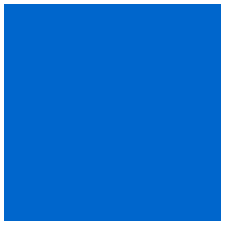
“En el presente capítulo se da una visión global sobre las distintas pruebas realizadas y los resultados obtenidos. Se explica el modo en que fueron llevados a cabo los test unitarios y las pruebas del sistema”.

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.

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, utilizado el formato establecido por IEEE en [**IEEE:citation**]. Por ejemplo, “el presente trabajo se basa en la plataforma EDU-CIAA-NXP [**CIAA**], la cual...”.

2.1.3. Figuras

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 con referencias cruzadas, por ejemplo: “Se eligió utilizar un cuadrado azul para el logo, como puede observarse en la figura 2.1”.

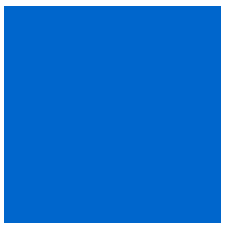


FIGURA 2.1. Ilustración del cuadrado azul que se eligió para el diseño del logo.

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

FIGURA 2.2. Imagen tomada de la página oficial del procesador[1].

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 2.2.

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 2.3. ¿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 2.3, porque sino el lector no comprenderá por qué de pronto aparece la figura en el documento, lo que distraerá su atención.

Otra posibilidad es utilizar el entorno *subfigure* para incluir más de una figura, como se puede ver en la figura 2.4. Notar que se pueden referenciar también las figuras internas individualmente de esta manera: 2.4a, 2.4b y 2.4c.

(A) Un caption. (B) Otro. (C) Y otro más.

FIGURA 2.4. Tres gráficos simples

El código para generar las imágenes se encuentra disponible para su reutilización en el archivo **`Chapter2.tex`**.

2.1.4. Tablas

Para las tablas utilizar el mismo formato que para las figuras, sólo que el epígrafe se debe colocar arriba de la tabla, como se ilustra en la tabla 2.1. Observar que

[1]Imagen tomada de https://goo.gl/images/i7C70w

sólo algunas filas van con líneas visibles y notar el uso de las negritas para los encabezados. La referencia se logra utilizando el comando `\ref{<label>}` donde label debe estar definida dentro del entorno de la tabla.

```
\begin{table}[h]
\centering
\caption[caption corto]{caption largo más descriptivo}
\begin{tabular}{l c c}
\toprule
\textbf{Especie}     & \textbf{Tamaño} & \textbf{Valor}\\
\midrule
Amphiprion Ocellaris & 10 cm           & \$ 6.000 \\
Hepatus Blue Tang    & 15 cm           & \$ 7.000 \\
Zebrasoma Xanthurus  & 12 cm           & \$ 6.800 \\
\bottomrule
\hline
\end{tabular}
\label{tab:peces}
\end{table}
```

TABLA 2.1. caption largo más descriptivo

Especie	**Tamaño**	**Valor**
Amphiprion Ocellaris	10 cm	$ 6.000
Hepatus Blue Tang	15 cm	$ 7.000
Zebrasoma Xanthurus	12 cm	$ 6.800

En cada capítulo se debe reiniciar el número de conteo 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. Por suerte la plantilla se encarga de esto por nosotros.

2.1.5. Ecuaciones

Al insertar ecuaciones en la memoria dentro de un entorno *equation*, éstas se numeran en forma automática y se pueden referir al igual que como se hace con las figuras y tablas, por ejemplo ver la ecuación 2.1.

$$ds^2 = c^2 dt^2 \left(\frac{d\sigma^2}{1 - k\sigma^2} + \sigma^2 \left[d\theta^2 + \sin^2 \theta d\phi^2 \right] \right) \tag{2.1}$$

Es importante tener presente que si bien las ecuaciones pueden ser referidas por su número, también es correcto utilizar los dos puntos, como por ejemplo "la expresión matemática que describe este comportamiento es la siguiente:"

$$\frac{\hbar^2}{2m} \nabla^2 \Psi + V(\mathbf{r}) \Psi = -i\hbar \frac{\partial \Psi}{\partial t} \tag{2.2}$$

Para generar la ecuación 2.1 se utilizó el siguiente código:

```
\begin{equation}
\label{eq:metric}
```

```
ds^2 = c^2 dt^2 \left( \frac{d\sigma^2}{1-k\sigma^2} +
\sigma^2\left[ d\theta^2 +
\sin^2\theta d\phi^2 \right] \right)
\end{equation}
```

Y para la ecuación 2.2:

```
\begin{equation}
\label{eq:schrodinger}
\frac{\hbar^2}{2m}\nabla^2\Psi + V(\mathbf{r})\Psi =
-i\hbar \frac{\partial\Psi}{\partial t}
\end{equation}
```

Capítulo 3

Diseño e implementación

3.1. Análisis del software

La idea de esta sección es resaltar los problemas encontrados, los criterios utilizados y la justificación de las decisiones que se hayan tomado.

Se puede agregar código o pseudocódigo dentro de un entorno lstlisting con el siguiente código:

```
\begin{lstlisting}[caption= "un epígrafe descriptivo"]
las líneas de código irían aquí...
\end{lstlisting}
```

A modo de ejemplo:

```
#define MAX_SENSOR_NUMBER 3
#define MAX_ALARM_NUMBER  6
#define MAX_ACTUATOR_NUMBER 6

uint32_t sensorValue[MAX_SENSOR_NUMBER];
FunctionalState alarmControl[MAX_ALARM_NUMBER]; //ENABLE or DISABLE
state_t alarmState[MAX_ALARM_NUMBER];           //ON or OFF
state_t actuatorState[MAX_ACTUATOR_NUMBER];     //ON or OFF

void vControl() {

  initGlobalVariables();

  period = 500 ms;

  while(1) {

    ticks = xTaskGetTickCount();

    updateSensors();

    updateAlarms();

    controlActuators();

    vTaskDelayUntil(&ticks, period);
  }
}
```

CÓDIGO 3.1. Pseudocódigo del lazo principal de control.

Capítulo 4

Ensayos y resultados

4.1. Pruebas funcionales del hardware

La idea de esta sección es explicar cómo se hicieron los ensayos, qué resultados se obtuvieron y analizarlos.

Capítulo 5

Conclusiones

5.1. Conclusiones generales

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.

Algunas preguntas que pueden servir para completar este capítulo:

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se puedo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

5.2. Próximos pasos

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