

TITULO DEL PROYECTO

NOMBRE CORTO DEL PROYECTO

AUTORES: AUTOR 1, AUTOR 2, AUTOR 3

Emails: estudiante1@fing.edu.uy, estudiante2@fing.edu.uy, estudiante3@fing.edu.uy

Tutores: Tutor 1 y Tutor 2

Emails: tutor1@fing.edu.uy, tutor2@fing.edu.uy

Instituto de Ingeniería Eléctrica - Facultad de Ingeniería - UDELAR

RESUMEN

Resumir el proyecto en no mas de 200 palabras. Recordar que la documentación no debe superar las 25 carillas, sin incluir esta carátula ni la tabla de contenidos o anexos. Recuerde que el resumen es una descripción corta, concreta y clara de los temas desarrollados, así como también de los métodos y herramientas utilizados y de los resultados obtenidos. Agregar una muy breve descripción de las conclusiones a las que se ha llegado.

Índice

5	1. Introducción	1
6	1.1. Descripción del problema	2
7	1.2. Antecedentes	2
8	2. Objetivos	2
9	3. Alcance	2
10	4. Descripción del Sistema	3
11	5. Implementación	3
12	5.1. Hardware	3
13	5.2. Software	4
14	6. Pruebas y Mediciones	7
15	7. Conclusiones	9
16	8. Anexo	9
17	8.1. Conceptos del curso aplicados al proyecto	9
18	8.2. Planificación del proyecto	9
19	8.3. Especificación del proyecto	9
20	Bibliografía	9

1. Introducción

Este es un template para la documentación del proyecto de fin del curso Sistemas Embebidos para Tiempo Real (SiSem) y en 2020 fue la primera vez que se puso en práctica. Por esto, en la documentación de proyectos de ediciones anteriores de SiSem presentan una gran diversidad de formatos. De cualquier manera, recomendamos revisar la documentación de proyectos de años anteriores disponibles en la página del curso.

La introducción se puede ilustrar con alguna figura (ver Fig. 1). Tener presente que todas las figuras deben estar referenciadas en el texto. Las figuras deben estar en formato pdf (deben ser generadas en formato vectorizado .eps, .ps, o .svg y luego convertidas a .pdf para incorporar al documento). En la fuente latex de este documento se puede mirar cómo insertar una figura, como la Fig. 1. Las referencias deben estar citadas por orden de aparición (por ejemplo así [1][2][3][4][5][6]) y en formato IEEE con un archivo bibtex. Con este template se adjunta un archivo bibtex de ejemplo (ver *refs.bib*) y el archivo *IEEEtran_bst_HOWTO.pdf* que explica en detalle cómo armar las referencias.

La numeración de líneas a la izquierda (línea a línea) deben dejarse para la entrega del documento borrador que los docentes van a corregir, pero para la entrega final que se va a subir a EVA deben quitarse. Para eso deben comentarse las líneas que dicen `\begin{linenumbers}` y `\end{linenumbers}`.

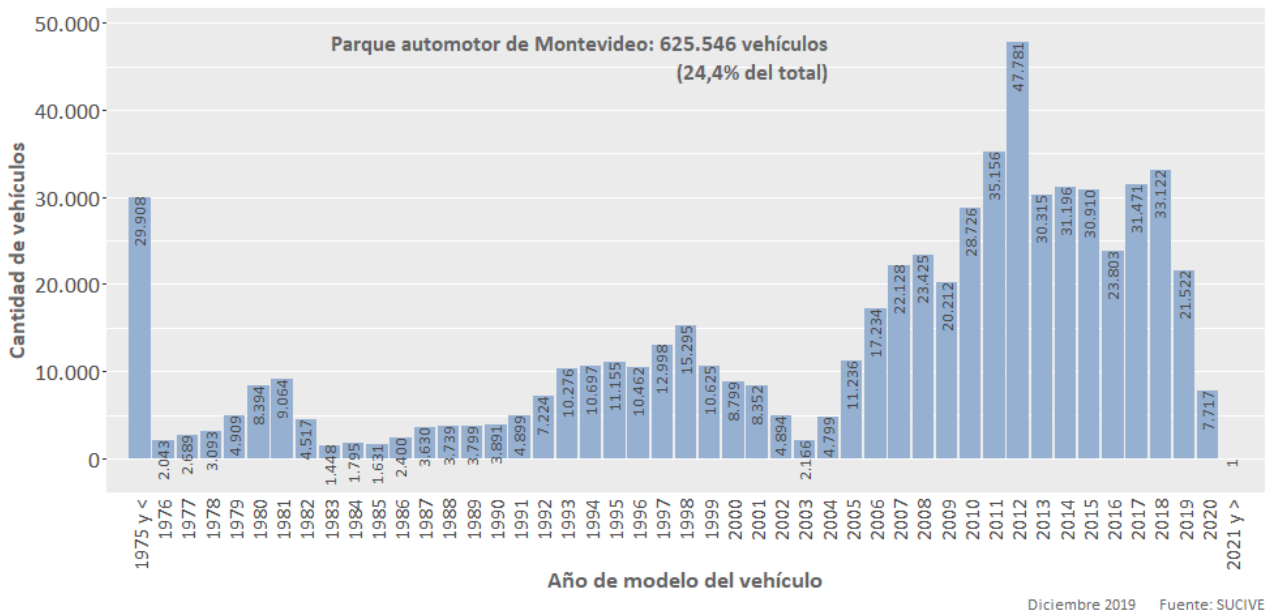


Figura 1: Evolución del parque automotor de Montevideo (reproducida de [7]). *Todas las figuras deben estar referenciadas y explicadas en el texto y además indicar de dónde fueron copiadas si no son originales, por ejemplo decir: figura adaptada o reproducida de [X].*

1.1. Descripción del problema

Describir el problema que será abordado. No solo se debe explicar brevemente el problema a nivel de aplicación sino que deben quedar identificados claramente los sub-problemas de sistemas embebidos que se tuvieron que resolver.

1.2. Antecedentes

Mencionar brevemente proyectos anteriores, artículos, libros, soluciones disponibles, etc., y cómo se relacionan con este proyecto.

2. Objetivos

Indicar brevemente los objetivos del proyecto planteados en la especificación detallada. Se pide los objetivos relacionados al curso y no otros. Pueden dividirse entre un objetivo general, y algunos objetivos específicos.

3. Alcance

Definir el alcance del proyecto, establecer qué parte del problema presentado en la Sección 1.1 se va resolver en el marco del presente proyecto. Decir qué se incluye dentro del proyecto y también qué no.

Si el problema que se está abordando está asociado a un proyecto de fin de carrera, especificar el alcance de ambos, diferenciando qué es lo que entra en el área de SiSem.

4. Descripción del Sistema

Esta sección debe contener la descripción de la solución que este trabajo propone para el problema propuesto en la Sección 1.1. La solución se nutre de los trabajos previos mencionados en la Sección 1.2, y pretende resolver el problema total o parcialmente de acuerdo a los objetivos y alcance definidos en las Secciones 2 y 3.

Se debe realizar una descripción funcional de la implementación (responder a las preguntas ¿Qué hace el sistema? ¿Cómo funciona?). La descripción deberá realizarse en texto con fuerte apoyo de diagramas de alto nivel (ver Figs. 2 y 3). Se trata de la descripción general del sistema y no se debe entrar en la descripción interna de los bloques de hardware ni de software (esto se realiza en la Sección 5). Asimismo, deberá agregarse toda otra información de carácter general que se considere relevante para describir el funcionamiento del sistema. Por ejemplo, si se considera necesario se podrán incluir diagramas de tiempos (pero siempre desde una perspectiva global).

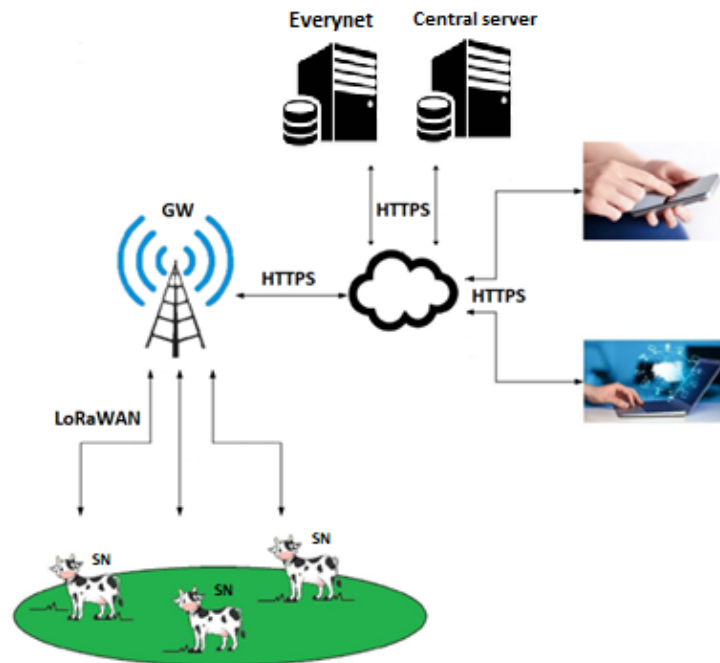


Figura 2: Diagrama funcional de alto nivel, reproducida de [8].

5. Implementación

Esta sección se divide en dos partes y será el núcleo de la documentación. Debe incluir información del tool-chain y/o IDE usado.

5.1. Hardware

En esta sección se describe el hardware utilizado, en particular aclarar y detallar si se tuvo que desarrollar hardware específico. Utilizar diagramas con los bloques constitutivos señalando cómo comunican entre ellos (Diagrama de Bloques) y/o Diagramas funcionales (Ver Figs. 3

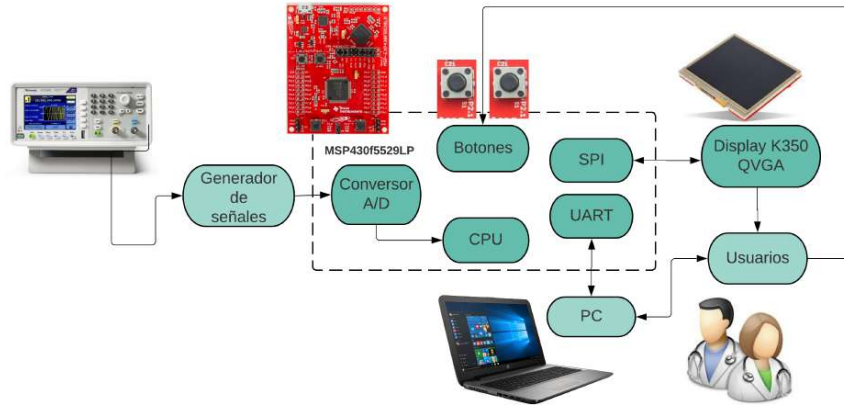


Figura 3: Diagrama funcional con cierto detalle a nivel de bloques, reproducida de [9].

76 y 4). Si corresponde, poner referencias a Notas de Aplicación o cualquier material utilizado.
 77 Poner diagramas esquemáticos de los circuitos y diagrama de conexión de bloques con pin-
 78 out (dependiendo del tamaño y complejidad pueden colocarse en un anexo). Para cada bloque
 79 constitutivo de hardware, así sea parte del interior del microcontrolador o no, debe indicarse
 80 que toma como entrada, de donde, y con qué otros bloques está interconectado. Indicar en el
 81 diagrama los buses y pines I/O usados, tensiones de alimentación, etc.

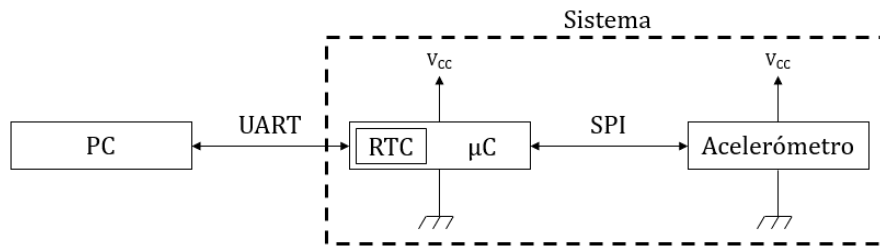


Figura 4: Diagrama de bloques (reproducida de [10]).

82 5.2. Software

83 Se deben adjuntar diagrama de módulos con división jerárquica (Diagrama de Módulos).
 84 Ver Fig. 5. Para cada capa, o módulo, o función se debe indicar qué variables tiene como
 85 entradas, de quién recibe dichas entradas, qué variables internas utiliza (y en qué formato, y el
 86 encapsulamiento, o sea qué variables son accesibles y cuáles privadas). Se debe indicar además
 87 qué funciones cumplen dentro de la implementación y con qué otro bloque, módulo o función
 88 se comunica para proporcionar resultados, comandos, información o manejar periféricos (se
 89 sugiere agregar las dependencias en los diagramas).

90 Se sugiere poner pseudocódigo de las partes que se consideren importantes del código
 91 utilizando el formato presentando para los algoritmos 1 y 2.

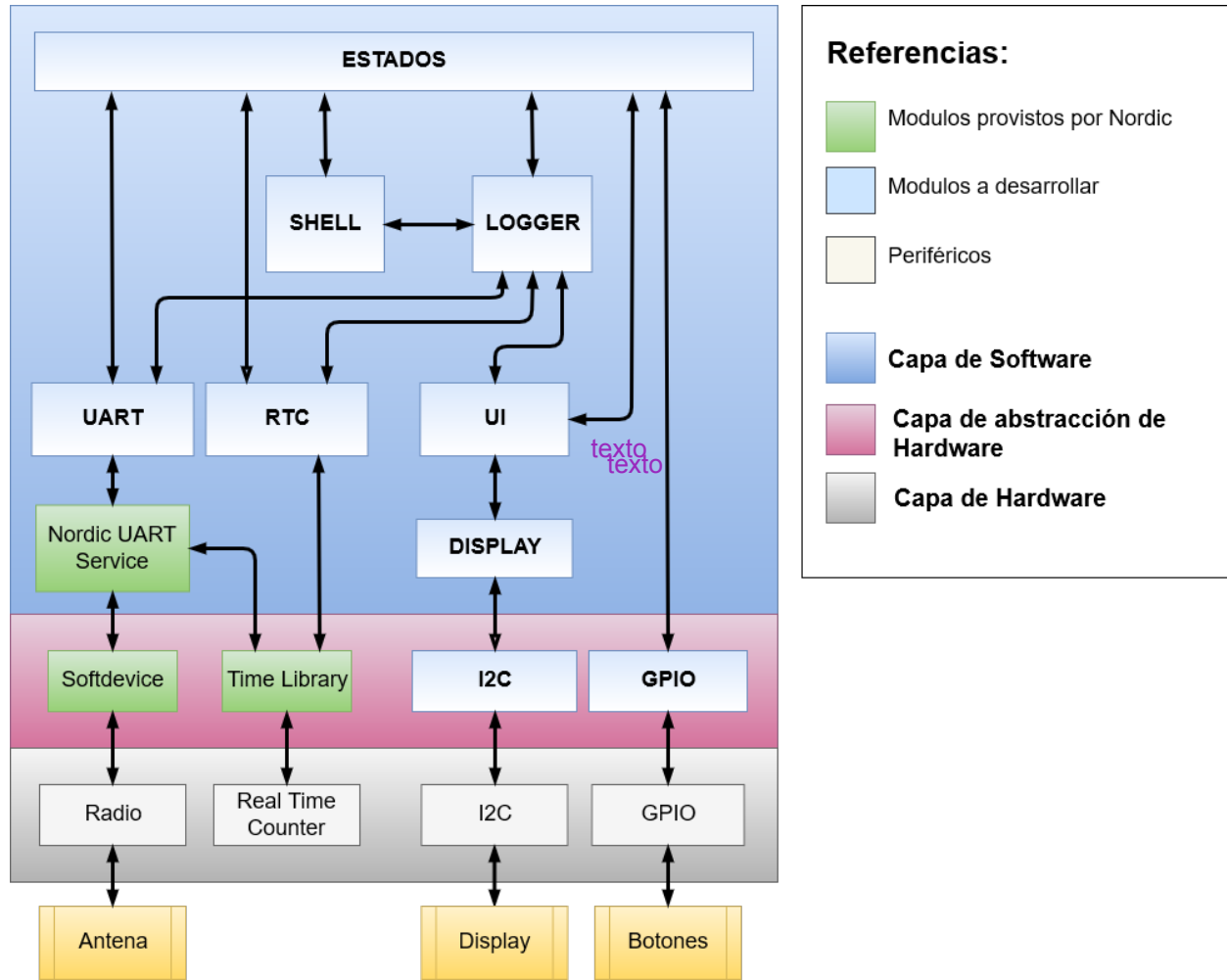


Figura 5: Diagrama de módulos. Se muestra la jerarquía, separando entre hardware y software (reproducida de [11]).

Algoritmo 1: EJEMPLO DE CÓMO ESCRIBIR PSEUDOCÓDIGO: esto es el nombre del algoritmo (puede ser una función, parte del main, una ISR o un algoritmo propiamente).

Input: entradas del algoritmo

Output: salidas del algoritmo

Result: resumir muy brevemente qué hace, a qué se corresponde esa salida (como función de las entradas)

Inicialización (variables locales, y si corresponde variables globales);

while *Condition* ≥ 1 **do**

 instructions;

if *condition something* $\neq 0$ **then**

 instructions1;

 instructions2;

 Reviso flag, TX, entrar a LPM3, etc.;

$d = \min\{c, e\}$;

 // puedo agregar comentario acá

else

$var1 \leftarrow var2$;

end

end

Algoritmo 2: MAXIMO: encuentra el máximo

Input: Un conjunto finito de números enteros $A = \{a_1, a_2, \dots, a_n\}$

Output: El entero mayor que pertenece al conjunto A

$max \leftarrow a_1$

for $i \leftarrow 2$ **to** n **do**

if $a_i > max$ **then**
 $max \leftarrow a_i$

return max

93 También se puede documentar la implementación del software embebido mediante diagramas
 94 de flujo, ver en Figs. 6 y 7. Incorporar diagramas de flujo del *main* y las *ISR*, diagramas de
 95 estados (ver Fig. 8), o de máquinas de estados (ver Fig. 9) si corresponde.

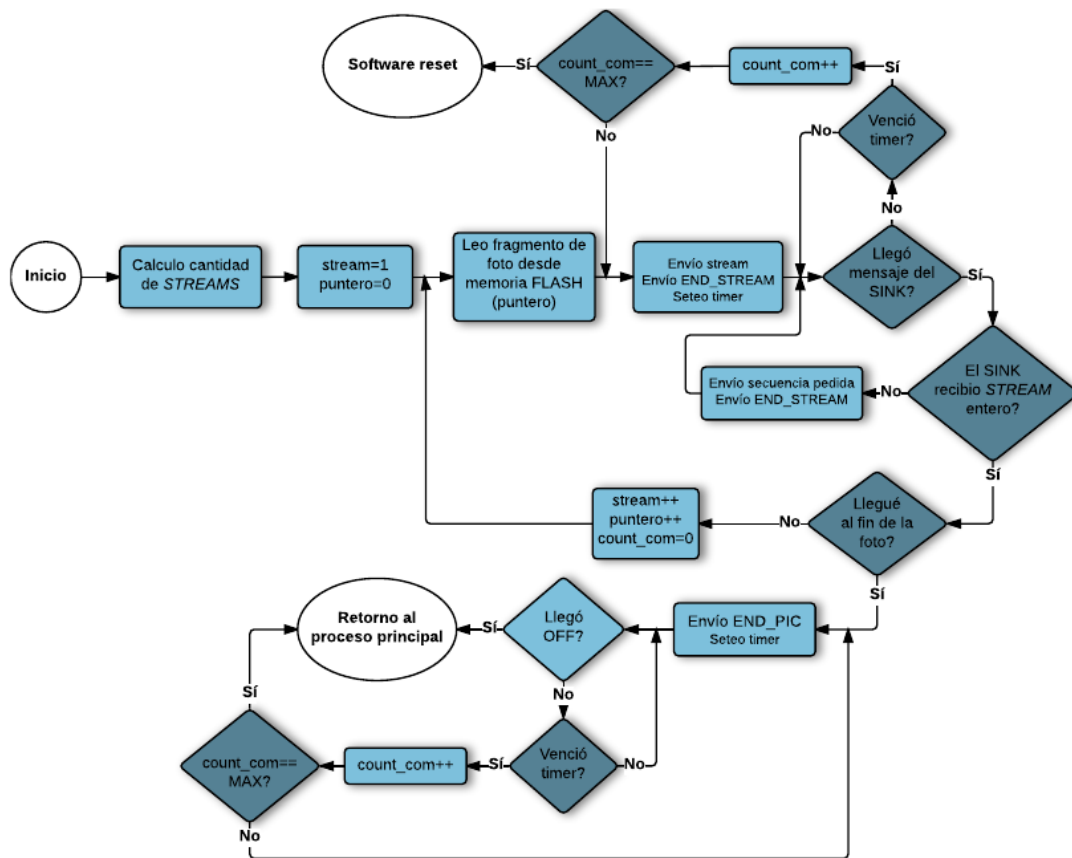


Figura 6: Diagrama de flujo de una función (reproducida de [1]).

96 Explicar la elección de la arquitectura de software (Round-Robin, Round-Robin con interrupciones,
 97 encolado de funciones o RTOS). Justificar en base a tiempos de respuesta y necesidades de
 98 prioridades. Explicar qué se procesa a nivel de *ISR* y *main* y qué criterios se tomaron para esas
 99 elecciones.

100 Mencionar si se tomaron las precauciones para generar código independiente del hardware.
 101 Mencionar qué código representa la HAL (hardware abstraction layer). Discutir grado de
 102 portabilidad, escalabilidad y extensibilidad es decir:

103 1. *portabilidad*: qué se debe cambiar si se cambia HW (por ejemplo microcontrolador o

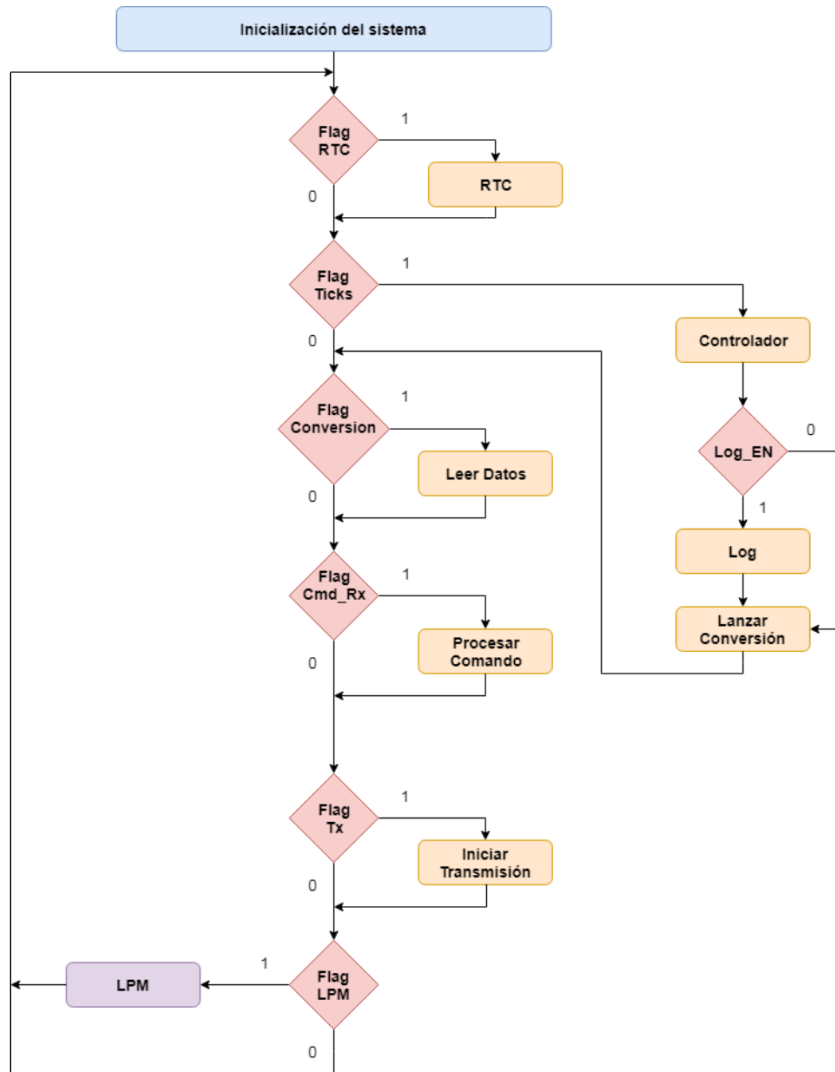


Figura 7: Diagrama de flujo del main, detalle del Round-Robin con interrupciones (reproducida de [12]).

cambio de algún periférico que tenga otro protocolo de comunicación).

2. *escalabilidad*: qué se debe agregar si a la implementación existente se desea agregar nuevas funciones (por ejemplo que acepte nuevos comandos, se agreguen nuevos mensajes de error).

3. *extensibilidad*: qué se debe agregar si a la implementación existente se agrega HW nuevo, por ejemplo otro sensor.

Explicar si hay problemas de datos compartidos, qué precauciones se tomaron y mencionar qué se está haciendo para minimizar el consumo, si es que se minimiza (si no se hace nada indicarlo).

6. Pruebas y Mediciones

Describir el setup de prueba y métodos (es decir descripción de las pruebas y sus resultados).

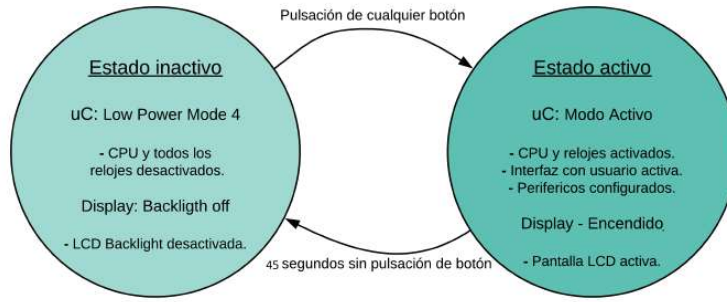


Figura 8: Diagrama según estados del microcontrolador (reproducida de [9]).

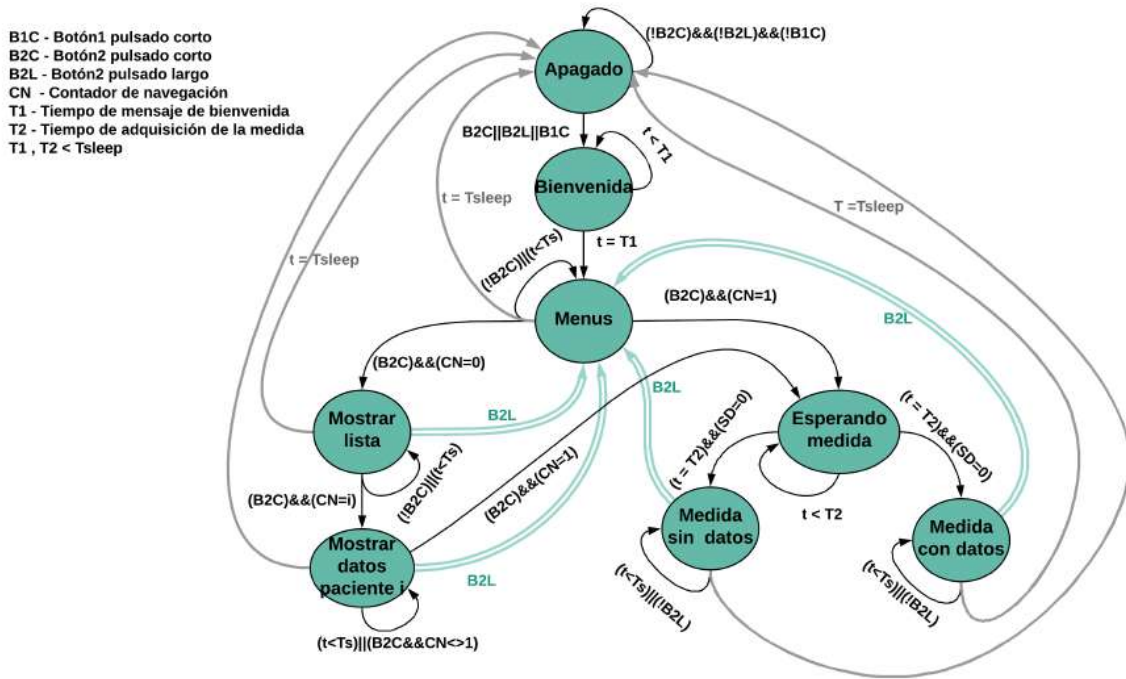


Figura 9: Diagrama en máquinas de estados (reproducida de [9]).

115 Presentar perfil del consumo usando EnergyTraceTM. Mostrar tiempos de ejecución de
 116 funciones, rutinas, métodos o procesos (agregar diagramas de tiempos). Incluir un reporte de
 117 uso del hardware del microcontrolador (mapa de memoria, Flash, cantidad de pines utilizados,
 118 etc.).

119 Agregar toda otra información relevante relacionado con pruebas y mediciones que muestren
 120 si los objetivos se cumplieron o no.

121 Explicar cómo se obtuvieron los resultados y qué funciones del IDE se utilizaron para ver
 122 variables, registros, memoria, etc. Si hace falta, adjuntar capturas de pantallas del IDE en modo
 123 debugger mientras se hacen pruebas.

124 No deben faltar las explicaciones de las gráficas que se adjunten, así como los nombres y
 125 las unidades en los ejes de las mismas. Agregar información en forma de tablas si se considera
 126 también relevante (ver ejemplo en Tabla 1).

Cuadro 1: Ejemplo de Tabla. La leyenda va arriba. No hay líneas verticales. Expresar los números con igual cantidad de cifras significativas. No olvidar poner las unidades.

	Ch1	Ch2	Ch3	Ch4	Ch5
A					
B					
C					

7. Conclusiones

Resumir el trabajo realizado y los resultados relevantes. Mencionar qué fue lo que salió mal o qué etapas del proyecto no se pudieron cumplir y el motivo. Discutir en relación a los objetivos, qué se cumplió y que no. Analizar ventajas y desventajas de las soluciones implementadas así como también la capacidad para escalar o cambiar el hardware de base. Si corresponde y se disponen de métricas para realizarlo, armar una tabla de comparación de desempeño con otros sistemas similares.

8. Anexo

8.1. Conceptos del curso aplicados al proyecto

Conceptos del curso aplicados al proyecto (justificación adecuada del diseño e implementación basados en los conceptos tratados en el curso). Este anexo debe contener un detalle de los conceptos del curso aplicados al proyecto. En el cuerpo del documento eventualmente se mencionarán algunas de estas consideraciones, pero el propósito del presente anexo es agrupar y resumir todas las consideraciones mencionadas, y otras que por razones de extensión no se incluyan en el cuerpo del documento.

8.2. Planificación del proyecto

Planificación del proyecto (crítica, reflexiones, etc. sobre: comparación entre planificación y ejecución, horas estimadas y efectivamente dedicadas).

8.3. Especificación del proyecto

Incluir la especificación del proyecto (documento ya entregado, sin cambios).

Bibliografía

- [1] F. Arbio, F. López, and M. Pereyra, "WSNvision," Proyecto de fin de Carrera, Universidad de la República (Uruguay). Facultad de Ingeniería., 2015. [Online]. Available: <https://hdl.handle.net/20.500.12008/5235>
- [2] *Code Composer Studio (CCS) Integrated Development Environment (IDE)*. [Online]. Available: <http://www.ti.com/tool/CCSTUDIO>
- [3] M. Alley, *The Craft of Scientific Writing*. Springer-Verlag New York, 2018, eISBN 978-1-4419-8288-9.

- 155 [4] N. Olloniego, M. Porto, and A. Rodriguez, “Smart-SUN,” Proyecto de fin de Curso SiSem,
156 Universidad de la República (Uruguay). Facultad de Ingeniería., 2019. [Online]. Available:
157 https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/G1SmartSUN.pdf
- 158 [5] D. Dang, A. Lele, and E. Wakefield, “Designing an Ultra-Low-Power (ULP) Application
159 With SimpleLink MSP432 Microcontrollers,” *Texas Instruments Incorporated. All Rights*
160 *Reserved.*, vol. SLAA668A, 2015. [Online]. Available: [http://www.ti.com/lit/an/slaa668a/](http://www.ti.com/lit/an/slaa668a/slaa668a.pdf?ts=1590097283394)
161 [sla668a.pdf?ts=1590097283394](http://www.ti.com/lit/an/slaa668a/slaa668a.pdf?ts=1590097283394)
- 162 [6] S. Branco, G. Ferreira, and J. Cabral, “Machine Learning in Resource-Scarce Embedded
163 Systems, FPGAs, and End-Devices: A Survey,” *Electronics*, vol. 8(11), 2019. [Online].
164 Available: <https://doi.org/10.3390/electronics8111289>
- 165 [7] Observatorio Nacional de Infraestructura, Transporte y Logística. (2020, Junio) Parque
166 automotor por departamento. Ministerio de Transporte y Obras Públicas. [Online].
167 Available: <https://observatorio.mtop.gub.uy/automotor.php>
- 168 [8] N. Acosta, N. Barreto, P. Caitano, R. Marichal, M. Pedemonte, and J. Oreggioni, “Research
169 platform for cattle virtual fences,” in *2020 IEEE International Conference on Industrial*
170 *Technology (ICIT)*, 2020, pp. 797–802.
- 171 [9] V. Cabrera, M. Lopassio, and M. Peña, “Gurí-Z,” Proyecto de fin de Curso SiSem,
172 Universidad de la República (Uruguay). Facultad de Ingeniería., 2019. [Online]. Available:
173 https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/G7GuriZ.pdf
- 174 [10] J. Facal, N. Gammarano, and A. Gurevich, “Plataforma embebida de base para proyecto
175 de fin de carrera MedAcel,” Proyecto de fin de Curso SiSem, Universidad de la República
176 (Uruguay). Facultad de Ingeniería., 2017. [Online]. Available: [https://eva.fing.edu.uy/](https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/ProcessAccelMemoria.pdf)
177 [pluginfile.php/66263/mod_page/content/38/ProcessAccelMemoria.pdf](https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/ProcessAccelMemoria.pdf)
- 178 [11] J. Bentancour, R. de Soto, and M. Gil, “Insulogger,” Proyecto de fin de Curso
179 SiSem, Universidad de la República (Uruguay). Facultad de Ingeniería., 2018.
180 [Online]. Available: [https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/](https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/insulogger%20documentacion.pdf)
181 [insulogger%20documentacion.pdf](https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/insulogger%20documentacion.pdf)
- 182 [12] M. Iglesias, J. Sanchez, and M. Rolon, “SISEST,” Proyecto de fin de Curso SiSem,
183 Universidad de la República (Uruguay). Facultad de Ingeniería., 2019. [Online]. Available:
184 https://eva.fing.edu.uy/pluginfile.php/66263/mod_page/content/38/G2SISEST.pdf