

UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
CARRERA DE ESPECIALIZACIÓN EN SISTEMAS
EMBEBIDOS



MEMORIA DEL TRABAJO FINAL

**Sistema de control para galvanoplastia
de Circuitos Impresos**

Autor:
Ing. Francisco Suárez

Director:
Ing. Juan Manuel Cruz (FIUBA,UTN-FRBA)

Jurados:
Esp. Ing. Jorge Fonseca (FIUBA)
Mg. Ing. Diego Brengi (INTI, UNLaM, FIUBA)
Esp. Ing. Ramiro Alonso (FIUBA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires, entre enero
de 2017 y diciembre de 2018 .*

Resumen

Esta memoria presenta el desarrollo de un sistema de control y monitoreo de parámetros físico-químicos asociados a un proceso de galvanizado industrial, realizado para el fabricante de circuitos integrados Daichi SA.

El sistema desarrollado asiste al personal técnico de planta en la optimización y mejora de los procesos vinculados al desarrollo de placas bicapa. A través de la lectura de sensores y almacenamiento de datos otorga funcionalidad similares a un PLC pero con funcionalidades a medida adicionales.

El prototipo se implementó sobre una EDU-CIAA y en el software se utilizaron técnicas de modelado orientado testing (TDD), y división por tareas para funcionar sobre un sistema multitareas. Además se aplicaron sistema control de versiones y trazabilidad de requerimientos.

Agradecimientos

A Yésica mi compañera de la vida por la paciencia y a mi familia por el apoyo incondicional de siempre.

Especial agradecimiento al Ing. Juan Manuel Cruz (FIUBA,UTN-FRBA) por haberme iniciado el mundo de los sistemas embebidos.

Y a todos los profesores de la CESE por el entusiasmo y las ganas que le ponen todos los días para que haya mejores profesionales.

Índice general

Resumen	III
1. Introducción General	1
1.1. Caso de Estudio	1
1.2. Necesidad y motivación	2
1.3. Objetivo	2
1.4. Alcance	3
2. Introducción Específica	5
2.1. Galvanizado de circuitos impresos	5
2.1.1. Proceso de galvanizado por electrólisis	5
2.2. Parámetros críticos del galvanizado	7
2.2.1. Fallas en el galvanizado de PCBs	8
2.3. Determinación de la solución a utilizar	9
2.3.1. Definiciones Generales	10
2.3.2. Casos de Uso	10
2.3.3. Requerimientos funcionales y no funcionales	12
3. Diseño e Implementación	17
3.1. Análisis del Hardware	17
3.1.1. Sensores y actuadores	17
3.1.2. Conexión al «poncho» diseñado	18
3.2. Arquitectura del Software	20
3.2.1. Capas de abstracción	21
3.2.2. Diseño	23
3.3. Sistemas control de entradas	24
3.4. Sistemas control de salidas	24
3.5. Sistemas control de terminal-Shell	25
3.6. Sistemas gestión de datos	25
4. Ensayos y Resultados	29
4.1. Pruebas funcionales del firmware	29
4.1.1. Pruebas unitarias	30
4.2. Ensayos de validación hardware	31
4.2.1. Pruebas de interfaz	32
4.3. Ensayos de integración	33
4.3.1. Validación de requerimientos	33
4.3.2. Pruebas de rendimiento	34
5. Conclusiones	37
5.1. Conclusiones generales	37
5.1.1. Hardware del prototipo	37
5.1.2. Firmware del prototipo	37

5.1.3. Proceso de desarrollo	38
5.2. Próximos pasos	38
A. Casos de Uso	41
B. Esquemáticos	45
Bibliografía	51

Índice de figuras

1.1.	Línea de galvanizada actual operada manualmente.	2
1.2.	Ejemplo de una línea de galvanizada automatizada.	3
2.1.	Proceso de galvanizado químico a través de la inmersión del objeto en distintos baños químicos.	6
2.2.	Proceso de galvanizado electrolítico a través de la inmersión del objeto en distintos baños químicos.	6
2.3.	Proceso electrolítico a través del cual se fija la capa de material sobre la superficie del objeto.	7
2.4.	Diagrama simplificado de las etapas	8
2.5.	Vía entre capas galvanizada en un PCB	8
2.6.	Perfil de vías según la ubicación relativa en los sustratos y el los cátodos.	9
2.7.	Las capas de cobre se alejan del borde del orificio.	9
2.8.	Banco de pruebas de validación del prototipo.	10
2.9.	Diagrama UML caso de uso de puesta en alta.	11
2.10.	Diagrama UML de casos de uso Modo 1 y 2.	12
3.1.	Circuito simplificado de amplificación y compensación de termocuplas.	18
3.2.	Circuito simplificado de amplificación y compensación de termistores.	18
3.3.	Conexión en modo funcionamiento 1	19
3.4.	Conexión en modo funcionamiento 2.	19
3.5.	Render vista de arriba placa de interfaz.	19
3.6.	Render vista de abajo de placa de interfaz.	20
3.7.	Vinculación de las periféricos del sistema con las interfaces internas en el modo 1.	21
3.8.	Diagrama de comunicación entre procesos en el modo 1.	21
3.9.	Vinculación de las periféricos del sistema con las interfaces internas en el modo 2.	22
3.10.	Diagrama de comunicación entre procesos en el modo 2.	22
3.11.	Estructuras de capas del sistema.	22
3.12.	Interacción entre tareas.	23
3.13.	Diagrama de estados de la tarea control de entradas.	24
3.14.	Diagrama de estados de la tarea control de salidas.	25
3.15.	Diagrama de estados de la tarea terminal de comunicación.	26
3.16.	Diagrama de estados de la tarea manejo de datos en memoria.	27
4.1.	Estructura de directorios del proyecto.	29
4.2.	Test de funciones interfaz de temperatura.	30
4.3.	Test de funciones interfaz de memoria.	31
4.4.	Archivos procesados en el test de temperatura.	31

4.5. Archivos procesados en el test de memoria.	32
4.6. Banco de pruebas.	32
4.7. Lista de comando implementados.	33
B.1. Esquemático jerárquico del poncho EduCiaa.	46
B.2. Esquemático entradas de termocupla y termistor.	47
B.3. Esquemático de salidas digitales con relays.	48
B.4. Esquemático de conectores de expansión con la eduCIAA.	49

Índice de Tablas

2.1. Descripción modos de funcionamiento.	11
3.1. Descripción tablas de comandos.	26
4.1. Cumplimiento de los requerimientos.	34
4.2. Uso de recursos por tarea.	35
A.1. Descripción caso de uso puesta en alta	42
A.2. Descripción caso de uso en Modo 1	43
A.3. Descripción caso de uso en Modo 2	44

Dedicado a los que impulsan la ingeniería nacional.

Capítulo 1

Introducción General

En este primer capítulo se presenta el contexto que dio origen al proyecto, durante el desarrollo de la materia *Gestión de Proyectos* y tras varias reuniones de trabajo con los interesados. Se explica el origen de la necesidad y motivación de realizar este sistema de control embebido, el objetivo principal y por ultimo los alcances que se pre-establecieron antes de comenzar el desarrollo.

1.1. Caso de Estudio

El presente proyecto corresponde a un contexto muy común dado en la manufactura industrial a baja escala: los operarios son los encargados de manipular y operar en prácticamente todas las etapas las maquinas y sus procesos. Cuando surge la necesidad de mejorar la calidad del producto final se vuelve necesario modernizar parte o el total de la linea de producción. Consecuencia de esto es entonces la automatización de dicha linea, lo cual implica en una gran reestructuración e inversión inicial. Esto ultimo es mayor aun cuando dicha solución se deja en manos de empresas dedicadas al rubro que normalmente utilizan tecnologías extranjeras.

La forma mas básica de modernizar un proceso industrial es asistiendo a través indicadores de estado de proceso, de alarmas y señales luminosas a los operadores. Así estos últimos solo se limitan a manipular llaves, pulsadores y atender el estado la linea en caso de alguna interrupción por anomalía, pero obligando a que estén presentes durante todo el proceso. En cambio en un sistema totalmente automatizado el trabajo del operador es mucho menor y solo se limita a dar inicio al proceso y a atender las alarmas en caso de alguna falla o anomalía hasta la finalización del mismo. En ambos casos el objetivo final es lograr un producto final con prácticamente el mismo índice de calidad y evitar perdidas de productos por no detectar fallas tempranas a tiempo.

Las soluciones para este tipo de entornos se logra a partir de dos maneras. Una es con sistema de control distribuido conformado por un modulo principal y varios submódulos secundarios, en la cual el modulo principal es el encargado de tomar la información de los submódulos y ejecuta las acciones correspondientes de los demás. La otra es usando módulos independientes entre si en donde cada uno se encarga de monitorear y controlar parámetros de una determinada etapa.

1.2. Necesidad y motivación

En el caso puntual de Daichi S.A. La necesidad radico en incrementar los niveles de calidad y el volumen de producción de placas de circuito integrados (PCB o Printed Circuit Board por sus siglas en ingles) doble faz. Debido a que la linea actual ya se encontraba próxima a estar obsoleta, se comenzó con el proyecto de construir una nueva linea en otro sector del edificio con mejores instalaciones y con un nivel de automatización intermedio en una primera etapa. Esto determinó la decisión de construir un sistema a medida de monitoreo de parámetros críticos del proceso pero con el control integro en manos de los mismos operadores de la planta actual.

Adicionalmente era vital tener registro histórico de los parámetros del proceso y de las acciones ejecutadas por los operadores, a modo de poder verificar y auditar el proceso en caso de detectar anomalías en el producto.

Por ultimo se planteo que dicha solución debería funcionar de puente entre esta etapa y una posterior de automatización integra del sistema. En la Figura 1.1 se puede ver como es la linea de proceso actual.



FIGURA 1.1: Linea de galvanizada actual operada manualmente.

En la Figura 1.2 se puede ver como es una linea de producción totalmente automatizada del fabricante Eurocircuits de Bélgica¹.

1.3. Objetivo

El objetivo fue diseñar un sistema de control sobre un hardware prototipo basado en un sistema embebido confiable y robusto, el cual funcionara como un modulo independiente de control y monitoreo por lo que tendría que ser fácilmente adaptable a los distintos configuraciones que podrían surgir en las distintas etapas de las linea de producción de los PCBs. A su ves debería usarse un hardware adicional que implemente las interfaces necesarias para los distintos sistemas de control a utilizar en la futura planta y al mismo poder probarlo en un banco de prueba a cargo del fabricante.

¹<https://www.eurocircuits.com>



FIGURA 1.2: Ejemplo de una linea de galvanizada automatizada.

1.4. Alcance

Previo al desarrollo se limito el alcance del proyecto en los siguientes puntos:

- Estudio preliminar de las arquitecturas adecuadas para la implementación del sistema principal y subsistemas.
- Diseño de alto nivel (arquitectura) del sistema.
- Diseño del sistema en lenguaje C.
- Plan de pruebas unitarias y ensayos (testbenchs) para cada subsistema.
- Plan de pruebas de integración y ensayos (testbenchs) para agrupaciones de subsistemas.
- Plan de pruebas del sistema y ensayos (testbenchs) para el sistema completo.
- Documentación del sistema y subsistemas que incluye:
 1. Descripción de entradas y salidas (frecuencias, tamaño y tipos de datos, señales de control, etc.)
 2. Descripción de parámetros del sistema.
 3. Requerimientos funcionales implementados trazables a los requerimientos del proyecto (matriz de trazabilidad).
 4. Hipótesis de diseño, justificación de la elección del diseño, estudios previos y marco teórico.
 5. Diagrama de arquitectura.
 6. Reporte de ensayos realizados.
 7. Referencias bibliográficas.
- Análisis y construcción del banco de pruebas.

Por otro lado se determino que el proyecto no incluiría el desarrollo de los siguientes puntos:

- Estudio de los sensores y actuadores, se basará dicha información en los datos dados por el cliente.
- Análisis de mejor solución para implementación de sistema de reporte remoto de variables y registros históricos.
- Análisis y construcción del banco de prueba necesario.
- Test del sistema en lugar de producción. La planta se encontraba en reestructuración y modernización.

Capítulo 2

Introducción Específica

Aquí se abordan con mas detalle el proceso de fabricación, los parámetros críticos y las fallas que pueden derivarse. Luego se exponen los criterios de diseño del prototipo a partir de los casos de uso del mismo, modos de funcionamiento. Finalmente se detalla la lista de requerimientos completos, a saber: funcionales y no funcionales, de interfaz, de diseño y futuros.

2.1. Galvanizado de circuitos impresos

El proceso de galvanizado consiste en adherir a un objeto metálico una delgada capa de otro metal, de unos pocos micrones de espesor. Este material le otorga mejores propiedades al objeto como ser conductividad eléctrica, mayor resistencia mecánica o resistencia a la corrosión.

Los métodos para lograrlos pueden ser diversos. Entre los mas comunes están el baño químico y por electrólisis. El método químico es utilizado generalmente en estructuras con superficies homogéneas y consiste en sumergir el producto dentro de un baño con el metal galvanizante en forma de solución química o bien a alta temperatura, incluyendo algunas etapas de pre-acondicionado. La figura 2.1 es un diagrama simplificado de método químico. El método electrolítico insume un gran consumo de energía eléctrica y mas etapas de pre-tratamiento de la superficie, pero tiene la ventaja de ser mas controlable el grosor del galvanizado, se suele hacer a temperatura ambiente y es mas versátil para aplicar sobre superficies no metálicas con un pre acondicionado previo. En la figura 2.2 se un proceso electrolítico donde básicamente el objeto recibe distintos pre-tratamientos para preparar la superficie antes y después del efecto electrolítico.

La forma utilizada en las placa de circuito impreso (PCB) es a través de la electrólisis del objeto dentro de una solución con sales metálicas, con una fuente de muy alta corriente continua de bajo voltaje, mas bloques del metal necesario para el galvanizado. En la figura 2.3 se muestra un caso donde el ánodo es de cobre y el cátodo es el PCB a galvanizar.

2.1.1. Proceso de galvanizado por electrólisis

En la fabricación de PCBs se parte de un material base de sustrato de láminas epoxi FR4 revestido con láminas de cobre. Este debe ser sometido previo a realizar la

Proceso de Galvanización

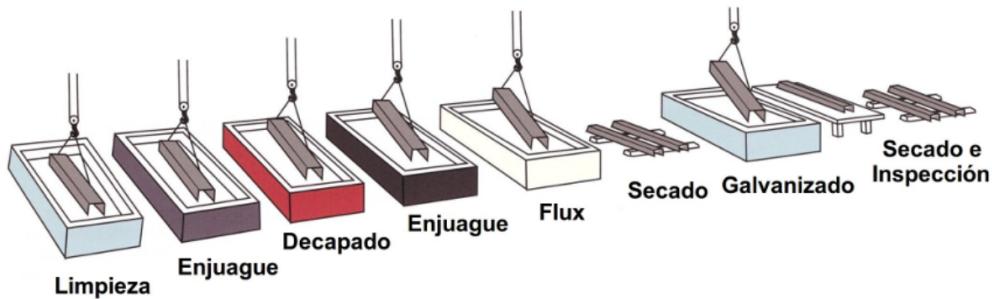


FIGURA 2.1: Proceso de galvanizado químico a través de la inmersión del objeto en distintos baños químicos.

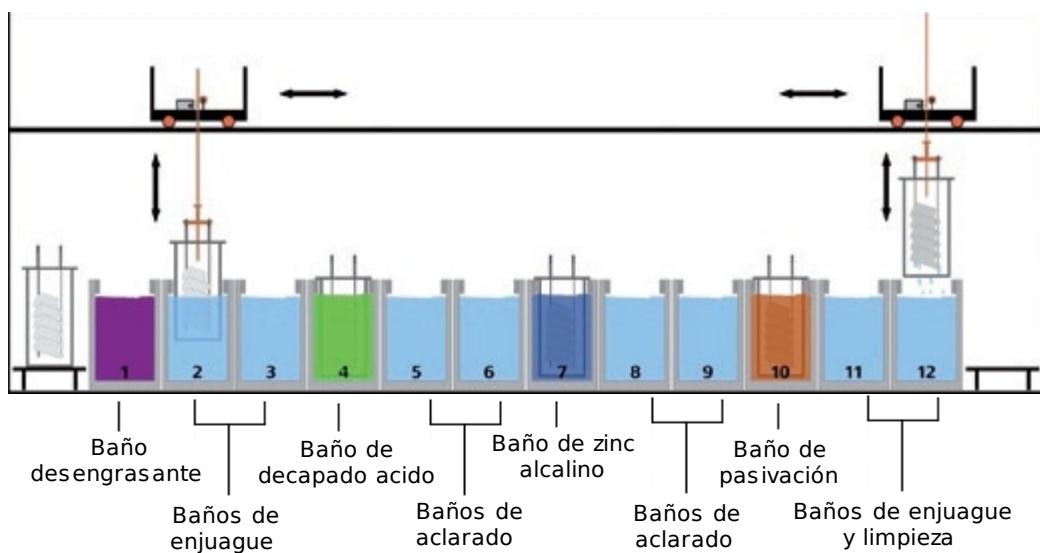


FIGURA 2.2: Proceso de galvanizado electrolítico a través de la inmersión del objeto en distintos baños químicos.

electrólisis en sí, a una sucesión de baños químicos para limpiar y homogeneizar toda las partes metálicas de superficie. El proceso es repetido con variantes del metal galvánico para lograr el circuito final.

En el proceso de fabricación de un PCB se utilizan aproximadamente 10 etapas distintas entre dos procesos. El primer proceso incluye el galvanizado con cobre de las vías entre capas, sobre el material laminado previamente perforado. El segundo proceso incluye el grabado del circuito impreso en si, previamente fijada la imagen negativa sobre las superficies del mismo. Existe una similitud en muchos de los pasos de inmersión de las placas en los productos químicos, que resulta en controlar mismos parámetros en las distintas cubas a lo largo del proceso. En la Figura 2.4 se muestra los efectos producidos sobre un perfil de una vía en un sustrato FR4, para las etapas mas relevantes.

²<https://www.4pcb.com/media/presentation-how-to-build-pcb.pdf>

²SASE2013-Circuitos Impresos de Marcos Mayer

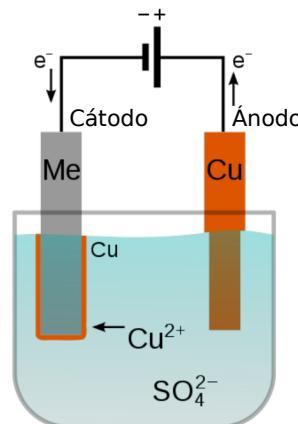


FIGURA 2.3: Proceso electrolítico a través del cual se fija la capa de material sobre la superficie del objeto.

En el proceso desarrollado en Daichi S.A. la serie completa, desde que se cortan los paneles FR4 hasta que se coloca la mascara antisoldante, incluye entre 20 a 25 de pasos los cuales se resumen a continuación lo mas importantes.

La figura 2.5 muestra el resultado final de un PTH (Plated Through Hole o agujero pasante metalizado) o vía, de un un PCB.

2.2. Parámetros críticos del galvanizado

En el proceso de galvanización de PCBs el control de ciertos parámetros físicos-químicos en cada una de las etapas es fundamental para garantizar la uniformidad y calidad del baño metálico.

En las diferentes bateas de la linea de galvanizado es necesario controlar, según cada etapa uno o algunos de los siguientes parámetros físicos-químicos del proceso:

- Temperatura.
- Nivel de líquidos.
- Conductividad o concentración de iones.
- Corriente entregada (electrólisis).
- Inyección de aire.

Actualmente en la linea de galvanizado los operadores cuentan solo con su experiencia y con el soporte de instrumental de medición básico que poseen en algunas las etapas de la linea. Para saber cuando una etapa esta finalizada se basan en el tiempo que saben funciona y por observación del estado del proceso. Esto puede acarrear muchos problemas si no se han estandarizado procesos. Además es causante de que el producto final muchas veces no tenga una calidad estándar y de que no se puedan prevenir fallas del proceso a tiempo.

Por todo esto para saber cuando un proceso fue realizado correctamente, es necesario tener información precisa, detallada y respetar los tiempos e indicadores preestablecidos para lograr el tipo de galvanizado buscado.

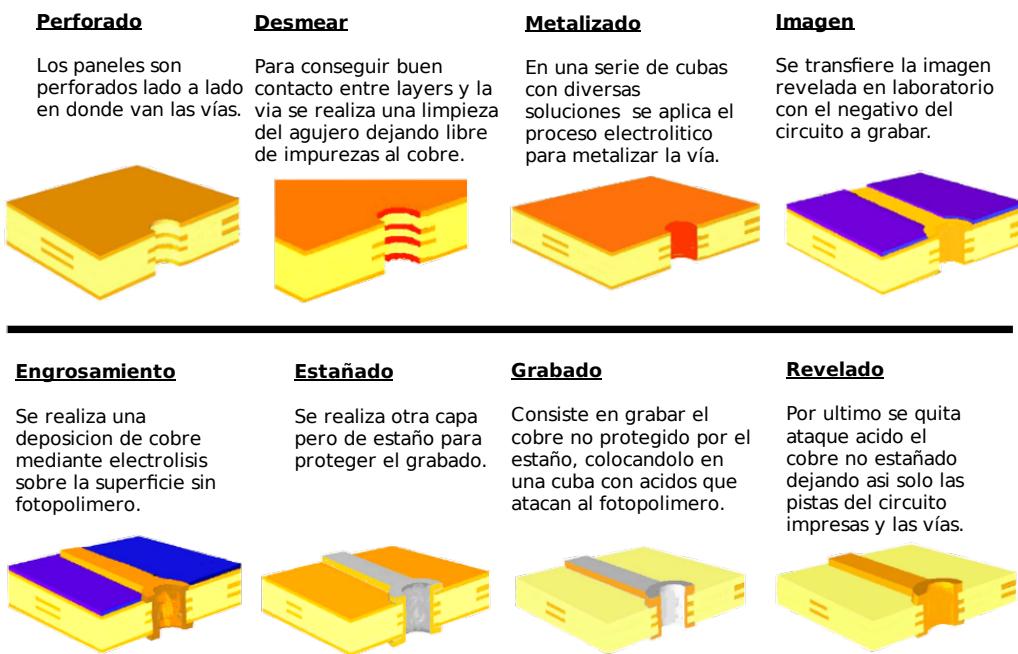


FIGURA 2.4: Diagrama simplificado de las etapas
12

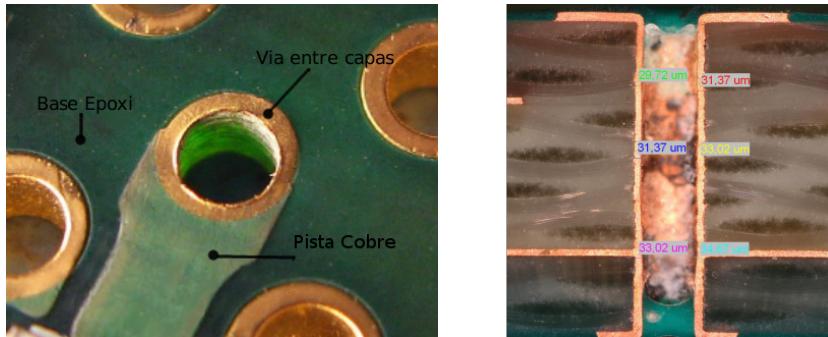


FIGURA 2.5: Vía entre capas galvanizada en un PCB

2.2.1. Fallas en el galvanizado de PCBs

En la producción de PCBs el resultado final correcto es aquel en donde el espesor del baño metálico es uniforme en toda la superficies y paredes de las vías. Cuando este proceso no ocurre correctamente se originan distintas fallas, de las cuales las más comunes son:

- Vías sin galvanizar.
- vías obstruidas por exceso de cobre.
- Laminado no uniforme de metal cobre.
- Micro cortes del cobre y puntos de no conductividad.

En la figura 2.6 se observan las distintos grosores de cobre con vías mal galvanizadas en función de su ubicación en el sustrato del PCB y la cercanía al ánodo de cobre.

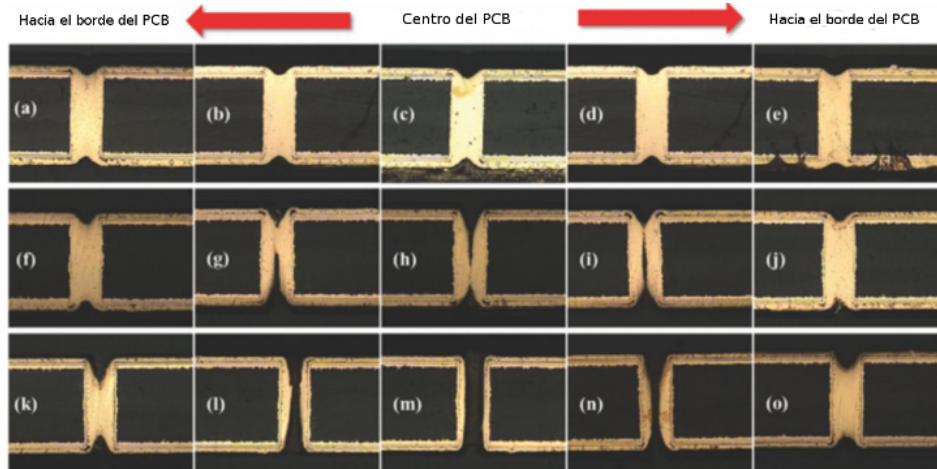


FIGURA 2.6: Perfil de vías según la ubicación relativa en los sustratos y los cátodos.

En la figura 2.7 se observa los problemas originados por la mala preparación previa de las superficies metálicas del sustrato, originado puntos sin continuidad o circuitos abiertos que lo volverán no aptos para la producción de circuitos.

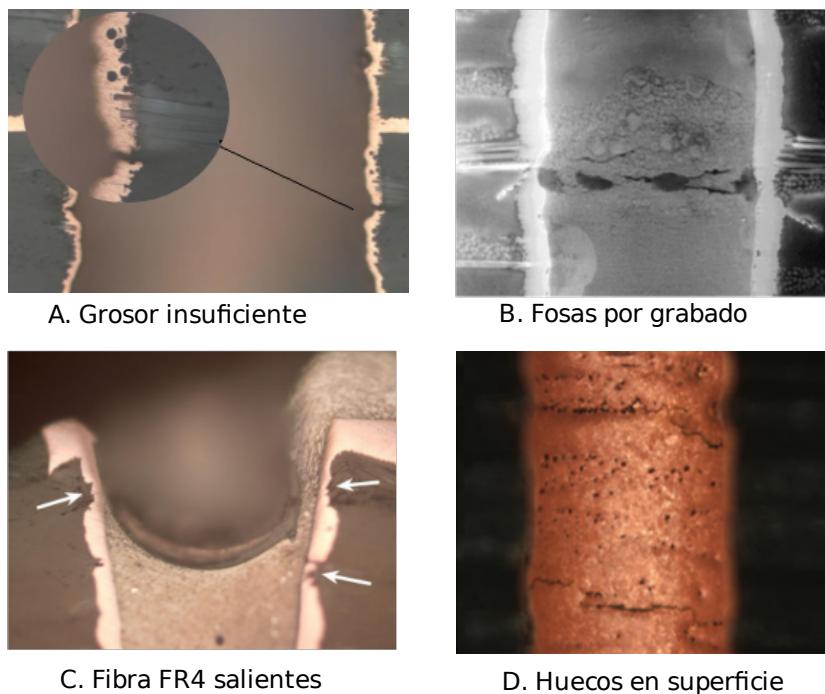


FIGURA 2.7: Las capas de cobre se alejan del borde del orificio.

2.3. Determinación de la solución a utilizar

A fin de lograr recopilar los detalles de la arquitectura del sistema desarrollado, es necesario primero definir y determinar casos de uso, requerimientos técnicos y definiciones generales originados por la nueva planta, su operación y las necesidades del cliente.

2.3.1. Definiciones Generales

La linea cuenta con varias etapas en forma consecutiva independientes entre si pero con parámetros característicos similares, se determinó implementar un solo prototipo capaz de procesar el conjunto de parámetros del proceso, capaz de interactuar con los distintos usuarios del entorno y configurable segun las etapas a monitorear.

El sistema debe funcionar entonces en modo predeterminado segun los distintos escenarios posibles dentro de las etapas del proceso. De esta manera con un mismo hardware mas los sensores y actuadores necesarios, se puede supervisar y controlar las partes del proceso que se crean necesarias. En la figura 2.8 se muestra la implementación del banco de prueba que se diseño para la validación del prototipo.

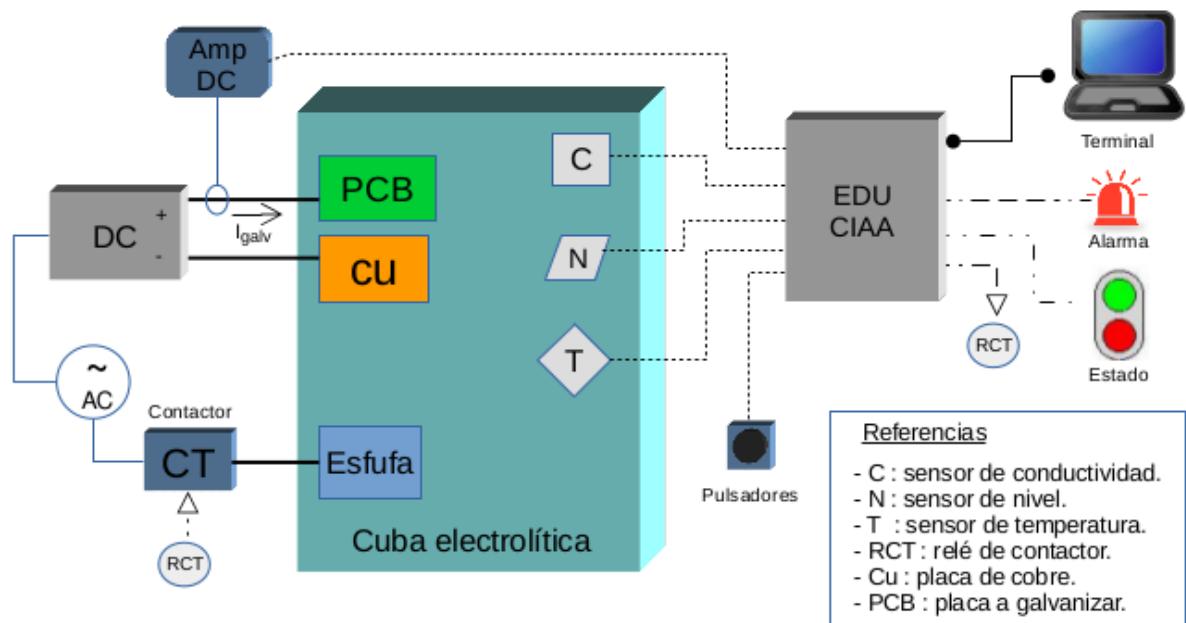


FIGURA 2.8: Banco de pruebas de validación del prototipo.

En una etapa posterior luego de la validación del prototipo el cliente desarrollará un hardware a medida para usar en su planta. El mismo será producido en las cantidades necesarias segun el numero de etapas que se determinen. Por ultimo estos distintos módulos funcionarán de manera independiente entre si.

2.3.2. Casos de Uso

Los casos de uso del sistema se definen para determinar como se vincula éste con los distintos usuario, en función del modo de funcionamiento posible. Estos quedan entonces predeterminados por los escenarios posibles que se pueden originar en la operación de la linea. Los casos mas relevantes son:

1. Puesta en alta y configuración del dispositivo, usado por personal técnico.
2. Puesta en marcha en modo de funcionamiento 1, usado por el operario y el técnico.

3. Puesta en marcha en modo de funcionamiento 2, usado por el operario y el técnico.

Los modos 1 y 2 se refieren a dos configuraciones a partir de los escenarios mas comunes de conexión del prototipo. A fin de simplificar la puesta en marcha en campo y por la limitación de numero de entradas analógicas de la EDU-CIAA. En la tabla 2.1 se observa un resumen de los periféricos involucrados en cada caso.

Modo	Función	Entradas	Salidas
Func 1	Controla nivel, temperatura, conductividad del liquido en la cuba.	Nivel tanque, temperatura del liquido, conductividad del liquido.	Contactor calefactor, contactor válvula de recambio, alarma de estado.
Func 2	Controla nivel, temperatura, energía consumida en la cuba.	Nivel tanque, temperatura del liquido, corriente entrante.	Contactor calefactor, contactor válvula de recambio, alarma de estado.
Config.	Modifica pines asignados, valores de alarma, tiempos de muestreo.	Comandos por terminal de pc.	Confirmación por terminal de pc y leds.

TABLA 2.1: Descripción modos de funcionamiento.

En la Figuras 2.9 y 2.10 se notan las interacciones entre los distintos submódulos del sistema en función de los casos de usos y los usuarios vinculados.

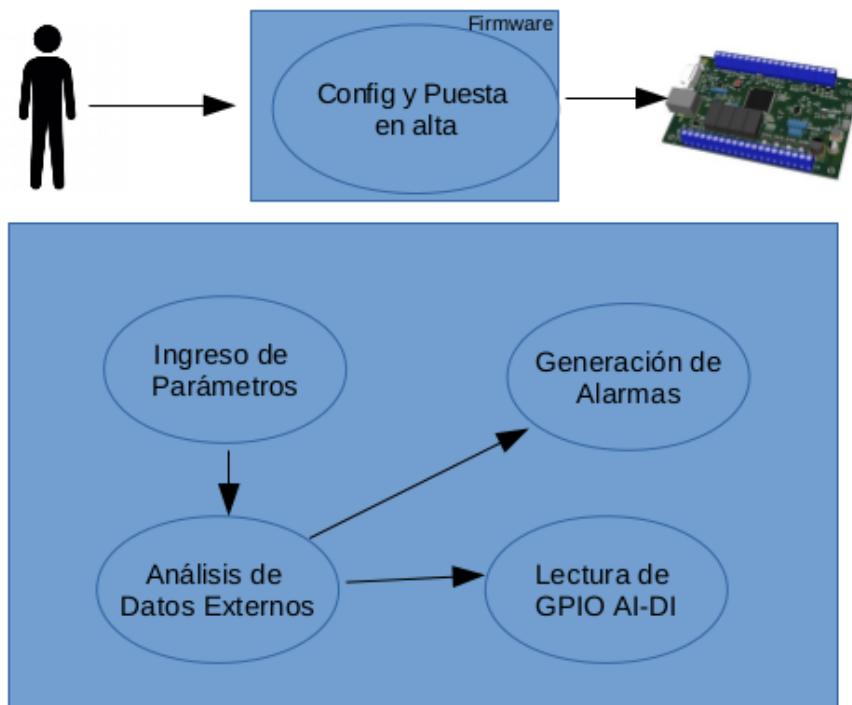


FIGURA 2.9: Diagrama UML caso de uso de puesta en alta.

La puesta en alta es el momento en que se determinan a través de la interfase terminal HMI (Humman Machine Interface o interfase hombre maquina) las parámetros a medir y configuraciones de los puertos del hardware.

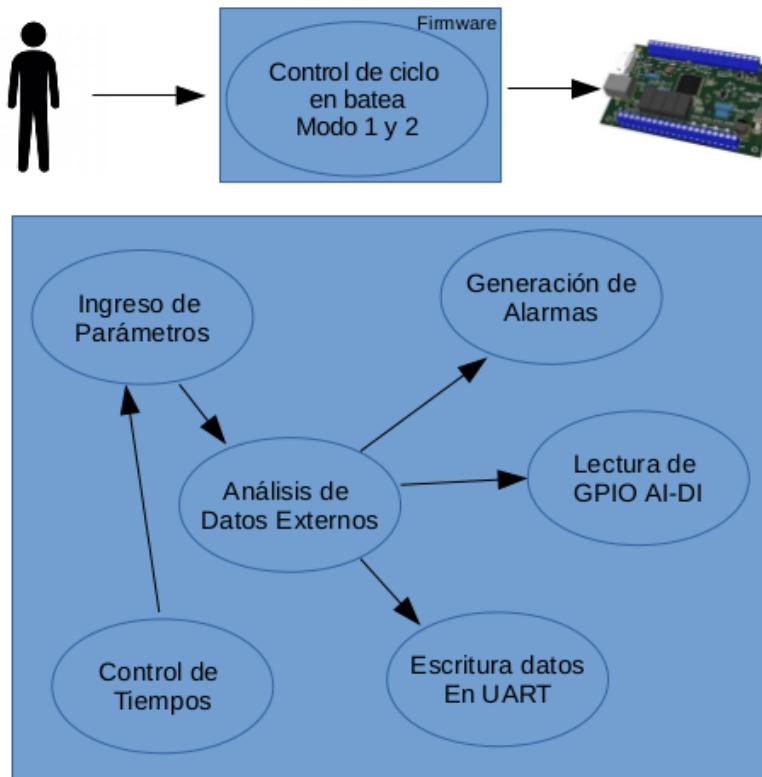


FIGURA 2.10: Diagrama UML de casos de uso Modo 1 y 2.

En el Apéndice A se detallan la secuencias involucradas en cada una de los casos de uso.

2.3.3. Requerimientos funcionales y no funcionales

A continuación se enumera lista completa con los requerimientos surgidos del análisis en conjunto con el personal de la planta tanto del problema, los casos uso y las prestaciones que tendrá la nueva planta. Cabe mencionar que la lista de requerimientos se originó previamente en la materia *Gestión de Proyectos* del posgrado.

1. Requerimientos Funcionales

a) Temperatura (RFTEM)

- 1) El sistema medirá la temperatura con un resolución de 1 grado , cada 15 segundos.
- 2) El sistema mantendrá la temperatura controlada dentro de rangos prefijados según la cuba a cual corresponda, con un PID con auto aprendizaje a no más de 5 grados de dispersión.
- 3) El sistema elevará la temperatura a través de la activación de una salida digital conectada a una resistencia.
- 4) La señal de temperatura será pasada a través de una ventana Smith Trigger (para evitar la intermitencia y generación de ruido en activations bruscas de la resistencia).

5) En caso de que la temperatura salga de rango por más de 5 minutos, se considerará interrumpir el proceso y emitir una alarma.

6) El sistema almacenará valores de temperatura de al menos 6 horas de funcionamiento en un archivo en memoria flash.

b) Energía (RFENE)

1) El software medirá la corriente total (DC) entregada al proceso de electrólisis cada 15 segundos, con 1 Amp de resolución.

2) El software medirá la tensión aplicada (DC) entre los bornes del electrólisis cada 15 segundos con 1V de resolución.

3) El sistema almacenará valores de corriente y tensión de por lo menos 6 horas de funcionamiento en un archivo en memoria flash.

4) Los rangos de valores óptimos de tensión y corriente serán tomados de los parámetros de lote ingresados por el usuario.

c) Conductividad (RFCOND)

1) El software medirá la conductividad, sólo en los tanques de agua des-ionizada, a través de una entrada analógica cada 60 segundos con resolución a definir.

2) El sistema debe activar una alarma si después de 10 minutos el nivel de conductividad no baja del valor máximo admisible, a definir luego.

d) Tiempos (RFTI)

1) El software llevará un conteo del tiempo entre cada baño en las bateas, desde el momento que se inicia hasta el final del proceso.

2) El sistema llevará un conteo regresivo de duración variable según la etapa, que inicia y detiene con un pulsador manual. Al completarse el tiempo debe emitir un señal lumínico/sonora para que el operador prosiga a la siguiente etapa.

3) En cada etapa deberá avisar y esperar a que un operario habilite la iniciación de la siguiente etapa.

e) Niveles de bateas (RFNB)

1) Evaluará que los niveles de líquido dentro del galvanizador estén dentro de los rangos permitidos de operación.

2) En caso de superar algún nivel crítico por más de 60 segundos se emitirán alarmas y se considera la interrupción del proceso.

f) Interfaces Hombre-Máquina (RFHMI)³

1) El sistema mostrará por pantalla gráfica el estado del sistema de modo con variables a definir.

2) El sistema contará con un método de ingreso de parámetros de lote, a procesar en modo texto.

3) Deberá permitir ingresar parámetros en modo manual y en modo automático, como letras y números.

- 4) Deberá brindar a través de una interfaz ethernet los históricos almacenados en memoria flash de variables del proceso que necesiten ser auditadas tras una etapa o tras el proceso completo. El máximo de registros será de XX número de puntos en formato YY.
4
- 5) El sistema mostrará por pantalla los rangos de valores óptimos de tensión y corriente según el lote ingresado, que deberán ser configurados manualmente en la interfaz de la fuente de alimentación principal del galvanizador. El sistema esperará una confirmación de que la fuente se configuró. El valor se podrá consultar una vez iniciado el proceso.

2. Requerimientos de Interfaz

a) Temperatura (RITEM)

- 1) La temperatura se medirá en los tanques, con un sensor por tanque multiplexadas a una entrada analógica.
- 2) La temperatura se controlará en los tanques por accionamiento de 1 resistencia térmica por tanque. En total se controlará en tres tanques.

b) Energía (RIENE)

- 1) Tendrá un sensor de alta corriente (500A) por cuba galvánica.
- 2) Tendrá un sensor de tensión (5V) conectados a los bornes de las cubas de galvanización.

c) Conductividad (RICOND)

- 1) Tendrá una electroválvula conectada a una salida digital por cada batea donde se tenga agua des-ionizada.
- 2) Tendrá un sensor de conductividad conectado a cada batea con agua des ionizada. En total se tienen seis sensores en entrada analógica.

d) Niveles de bateas (RINB)

- 1) Tendrá sensores de nivel en los tanques donde haya control de temperatura.

e) Interfaces Hombre-Máquina (RIHMI)

- 1) Mostrará información a través de un puerto VGA/HDMI con una taza de refresco menor a 5 segundos.
- 2) Tomará de una entrada serie USB los valores de lote. (+1 USB)
- 3) Accionará a través de una salida digital una alarma sonora/lumínica en caso de algún tipo de falla.
- 4) Contará con pulsadores a fin de poder detener y accionar el procesos de galvanización, conectados a una entrada binaria.
- 5) Contará con balizas de señalización de estado del sistema, verde funcionando, rojo detenido.

6) Dispondrá de una conexión remota a través de Ethernet. (+1 ETH)

3. Requerimientos no Funcionales (RNF)

a) Deberá ser probada la funcionalidad a través de un banco de pruebas que se ajuste al comportamiento del sistema.

4. Restricciones de Diseño (RD)

a) De los requerimientos de interfaz se resume que como mínimo el hardware deberá contar con las siguientes interfaces:

- Entradas analógicas (AI) ≥ 3
- Entradas digitales (DI) ≥ 3
- Salidas analógicas (AO) ≥ 1
- Salidas digitales (DO) ≥ 4
- Puerto USB (USB) = 2
- Puerto RED (ETH) = 1

5. Requerimientos a Futuro (RAF)

a) Brindar información acerca de si es necesario realizar una limpieza de sistema. Se puede utilizar como parámetro el número de procesos que se ejecutaron.

b) Deberá permitirloguearse al iniciar el proceso como un responsable de operación.

c) Deberá interactuar con una cinta de transportación automática que llevará las placas de una batea a otra. Accionara los motores de transporte y de elevación.

d) Si no se respetan los tiempos el sistema deberá dejar asentado el técnico y las acciones manuales ejecutadas a fin de tener un histórico antes posibles fallas en el lote.

e) Deberá detectar si el sistema de extracción de gases está activo.

f) Deberá controlar el ventilador del sistema extractor de gases con un regulador por frecuencia o ancho de pulso.

g) Deberá detectar si la inyección de aire está activa en los tanques que lo requieren.

En función de estos se propuso implementar el proyecto sobre una plataforma EDU-CIAA [6] en conjunto con una placa de interfaz hecha a medida.

⁴Los RFHMI relacionados con el ingreso de lotes no pudieron implementarse en esta primer etapa del proyecto al no tener un sistema centralizado que controle las etapas.

⁴Los parámetros definidos con XX son aquellos que no pudieron ser definidos en su momento con el cliente.

Capítulo 3

Diseño e Implementación

Aquí se presenta el detalle del hardware planteado como un «poncho» sobre la EDU-CIAA, y la interfaz con los elementos de la planta. Luego se describe la estructura lógica del software con detalles del diseño sobre diagramas de interacción entre tareas, diagramas de capas y otras consideraciones relevantes del código.

3.1. Análisis del Hardware

El prototipo se implementó sobre una EDU-CIAA en conjunto con un hardware de adaptación de las interfaces. Para ello se utilizaron los proyectos de código abierto Kicad [9] para la elaboración del poncho de la EDU-CIAA [2] que adapta las entradas y salidas de los conectores de expansión con los sensores y actuadores utilizados en la planta.

3.1.1. Sensores y actuadores

Para la adaptación de las interfaces con termocuplas y termistores se necesitaron circuitos de adaptación de señal. Para el primer caso se usó un circuito integrado compensador de juntura debido a la alinealidad en la respuesta de ese tipo de sensores. Para ello se eligió el MAX31855KASA+¹ como una solución simplificadora. También existen soluciones implementadas con amplificadores multietapas que son más económicos pero que obligan a implementar la transducción de la señal en el microprocesador [4].

En la Figura 3.1 se observa una configuración típica de adaptación para termocupla.

En el caso del termistor el circuito de adaptación es más simple debido a que señal obtenida es prácticamente lineal y solamente es necesario amplificar a los niveles de trabajo de la entrada analógica del procesador. En la Figura 3.2 se observa en tipo de circuito que se implementó.

En el caso de la medición de corriente y tensión continua para las pruebas solo se consideraron sensores de baja potencia a fin de poder validar el comportamiento. Cabe aclarar que en línea de producción las fuentes de galvanizado son de muy alta potencia y la intensidad es leída actualmente con pinzas amperímetros AC/DC. La medición de tensión se toma en principio con un circuito aislado

¹bucarlaurconlahojadedatos

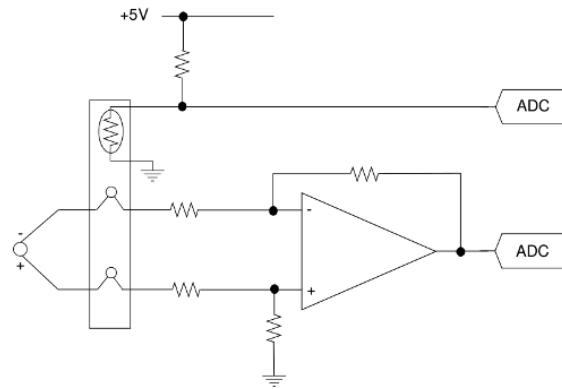


FIGURA 3.1: Circuito simplificado de amplificación y compensación de termocuplas.

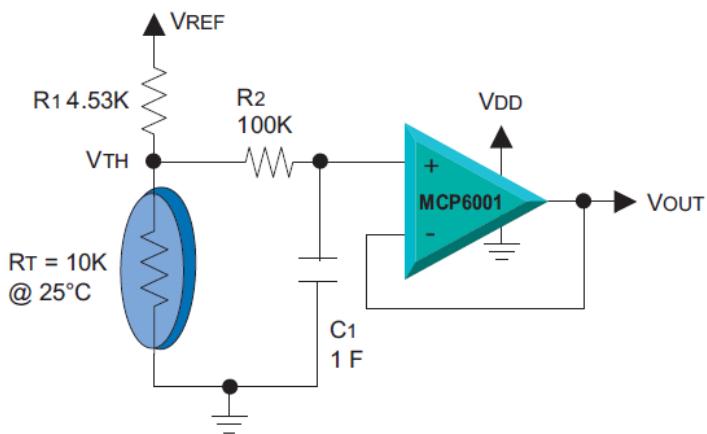


FIGURA 3.2: Circuito simplificado de amplificación y compensación de termistores.

con foto-transistores para obtener mayor aislación de la fuente de corriente. La medición de continuidad o salinidad del agua se resolvió implementarla con un circuito de terceros conectado a través de una de las entradas analógicas del prototipo. Las señales a medir son tensiones capacitivas entre dos varillas metálicas de muy bajo valor, la cuales se necesita acondicionar con operacionales de muy bajo piso de ruido o circuitos conversores analógicos digitales cerca del sensor.

Respecto a los actuadores se resolvió implementarlos con relés electromecánicos mas los circuitos de protección por sobre-picos de tensión. Estos irán conectados a contactores que habilitan los distintos dispositivos como son: resistencias eléctricas, aireadores, alarmas, indicadores luminosos, etc.

3.1.2. Conexión al «poncho» diseñado

En función de los periféricos que se acordaron utilizar en el prototipo, se diseñó un placa de expansión o poncho para utilizar con la EDU-CIAA a fin de tener un plataforma de prueba mas confiable.

Según las configuraciones de la sección 2.3.2 y en función de los puertos disponibles de la EDU-CIAA, se establecieron dos modos de conexión de sensores y

actuadores. En la figuras 3.3 y 3.4 se observa como se corresponderían las conexiones a los periféricos del prototipo en función de los modos de uso.

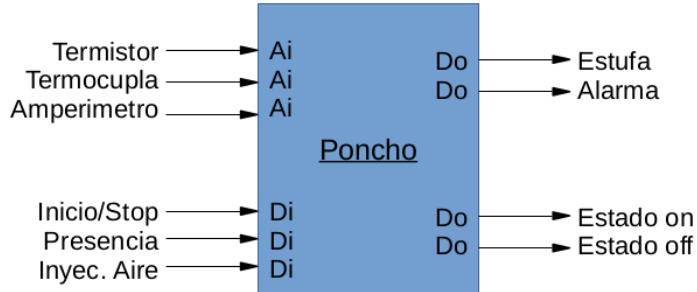


FIGURA 3.3: Conexión en modo funcionamiento 1

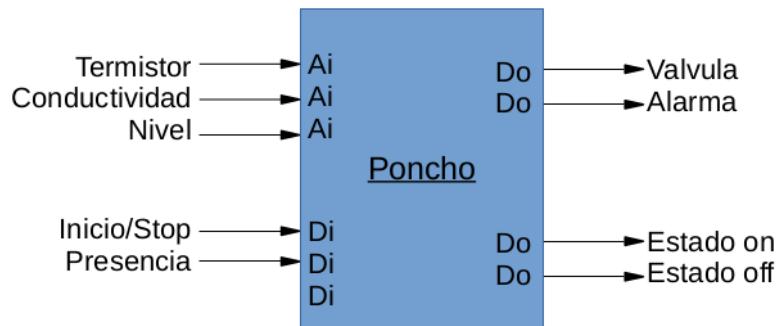


FIGURA 3.4: Conexión en modo funcionamiento 2.

Los puertos están preconfigurados y tienen asignaciones de pines predeterminadas, no obstante estos pueden modificarse desde la terminal al ingresar al modo de configuración.

En las figuras 3.5 y 3.6 se muestran las vistas del prototipo propuesto para desarrollar en el sistema piloto.

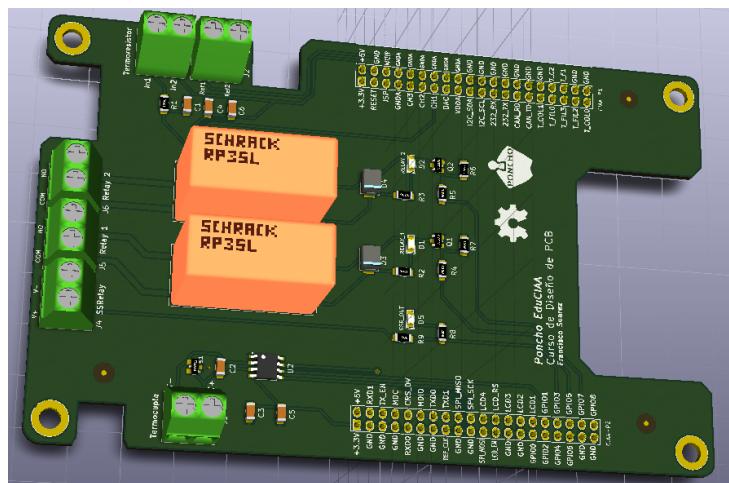


FIGURA 3.5: Render vista de arriba placa de interfaz.

La placa de la figura no llegó a construirse hasta la fecha debido a que no se habían terminado de definir detalles relevantes respecto al numero y tipos de sensores y actuadores a utilizar.

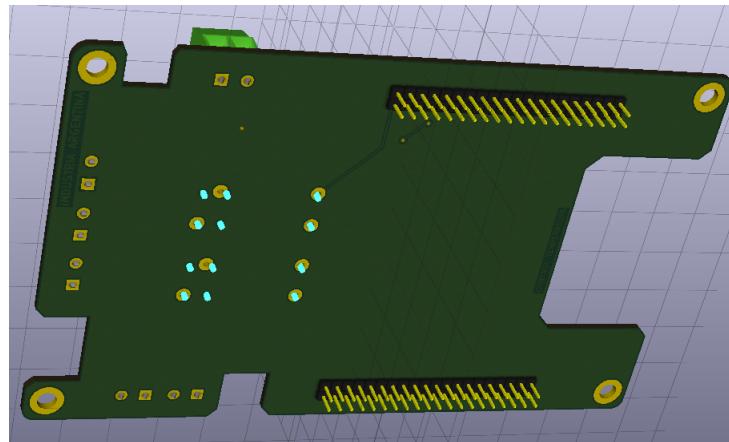


FIGURA 3.6: Render vista de abajo de placa de interfaz.

En el Apéndice B se encuentran los esquemáticos completos de las placa de expansión.

3.2. Arquitectura del Software

En función de los requerimientos vistos en 2.3.3 y de implementar los conocimientos aprendidos durante el posgrado, se determinaron los criterios para esta etapa.

En primer lugar se eligió usar lenguaje C y el sistema operativo FreeRTOS para la implementación. En segundo se diseñó el código de manera modularizado en tareas las cuales pueden funcionar de modo independiente según si la implementación lo requiere. En tercer lugar se tiene como modelo patrón el denominado «control ambiental» el cual se emplea cuando es un sistema con sensores que proporcionan información sobre el entorno y los actuadores pueden cambiar.

La aplicación cuenta los siguientes componentes:

- a. Interfaz de usuario (ingreso de parámetros, uart, lcd, alarmas).
- b. Control de temperatura.
- c. Monitoreo de nivel.
- d. Monitoreo de conductividad.
- e. Monitoreo de energía.
- f. Control de tiempos.

Según el modo de funcionamiento seleccionado se tendrá una diferente configuración del software siempre dentro de la misma arquitectura y con pocas diferencias entre ellos. En la figura 3.7 se muestran los diagramas de interfaces y procesos internos en el modo de funcionamiento 1.

En la figura 3.8 se muestra el diagrama de comunicación entre los procesos originados para el caso de funcionamiento 1. Se nota su relación con los componentes de la aplicación: a,b,e y f.

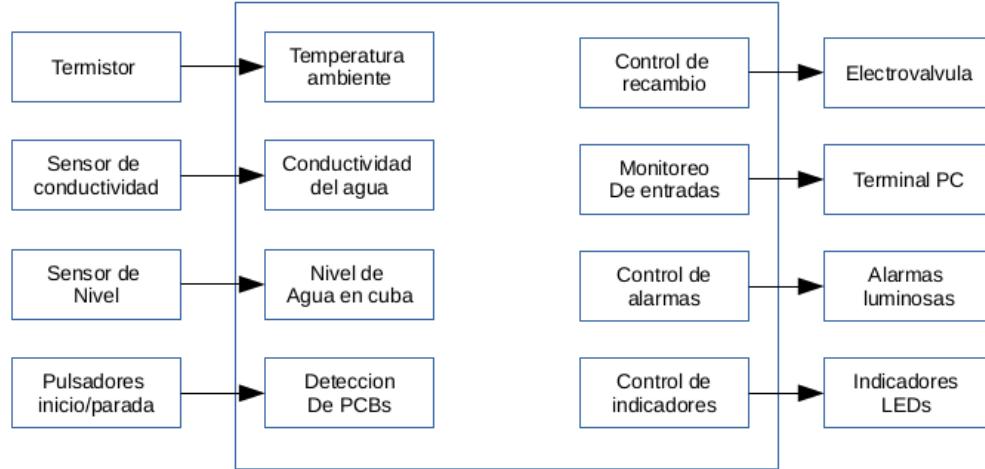


FIGURA 3.7: Vinculación de las periféricos del sistema con las interfaces internas en el modo 1.

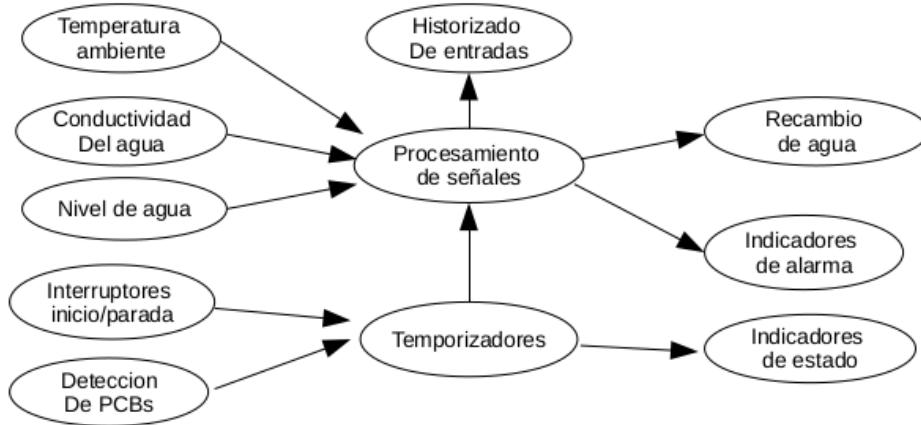


FIGURA 3.8: Diagrama de comunicación entre procesos en el modo 1.

En la figura 3.9 se muestran los diagramas de interfaces y procesos internos en el modo de funcionamiento 2. Se nota que hay poca diferencia con el caso anterior, solo tienen activos otros procesos de interfaz.

En la figura 3.10 se muestra el diagrama de comunicación entre los procesos originados para el caso de funcionamiento 2. Se nota su relación con los componentes de la aplicación: a,b,c, y f.

3.2.1. Capas de abstracción

el sistema se divide en capas que otorgan abstracción del hardware y además permiten obtener código mas reutilizable e independiente de la plataforma. Estos son cambios que el cliente puede solicitar a futuro. En la figura 3.11 se observa el patrón de capas de la arquitectura.

Las capas son las siguientes:

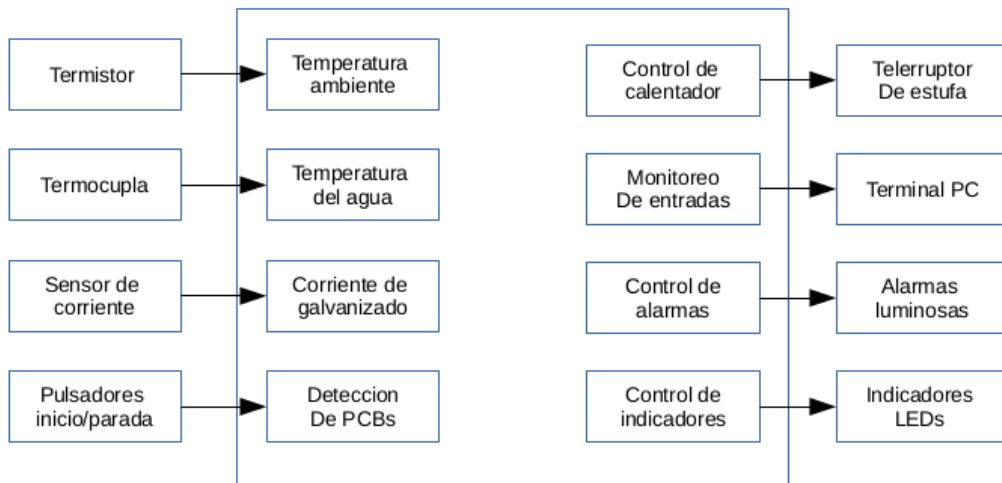


FIGURA 3.9: Vinculación de las periféricos del sistema con las interfaces internas en el modo 2.

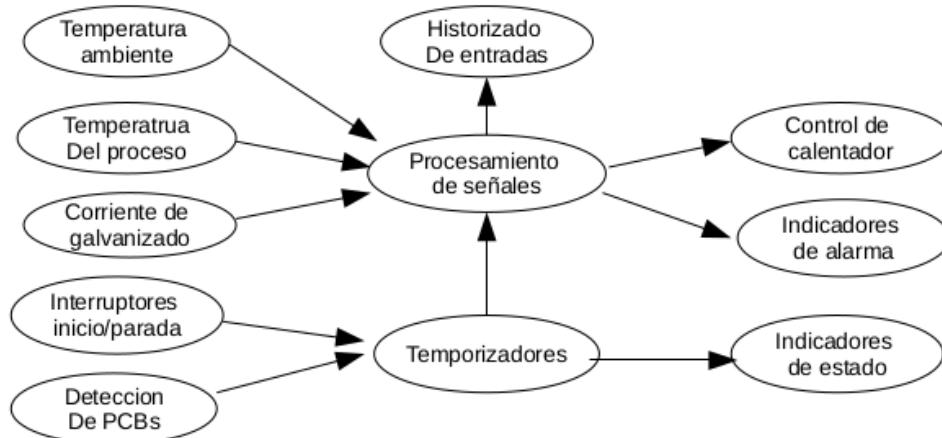


FIGURA 3.10: Diagrama de comunicación entre procesos en el modo 2.

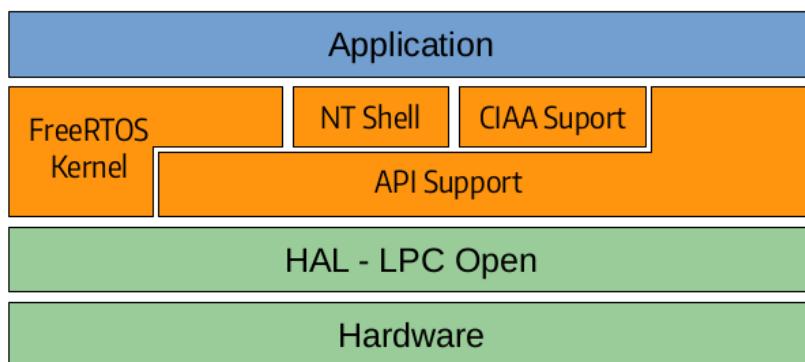


FIGURA 3.11: Estructuras de capas del sistema.

- *Aplicación*: Rutinas creadas para cada una de las tareas.
- *NT Shel*: Implementa el interprete de comandos ingresados externamente e

internamente, a fin de configurar o de obtener información del dispositivo [3].

- *FreeRTOS*: Sistema operativo, controla la comunicación entre tareas, acceso a recursos compartidos y el scheduler. [1].
- *API Support*: implementa las rutinas de acceso y configuración de los distintos periféricos.
- *HAL*: rutinas de acceso al hardware, implementada con la librería LPC OPEN para el uP LPC43XX [5].

3.2.2. Diseño

Por la necesidad de dividir el problema en partes menores para hacerlo mas escalable, mantenable y debugueable, el sistema se divide en subsistemas o tareas cooperativas entre si. Las mismas están definidas como:

- Monitoreo de entradas analógicas-digitales.
- Control de salidas digitales.
- Manejo de terminal y comandos.
- Manejo de memoria y log de datos.

En la figura 3.12 muestra un diagrama de estado general de las tareas y la interacción entre ellas.

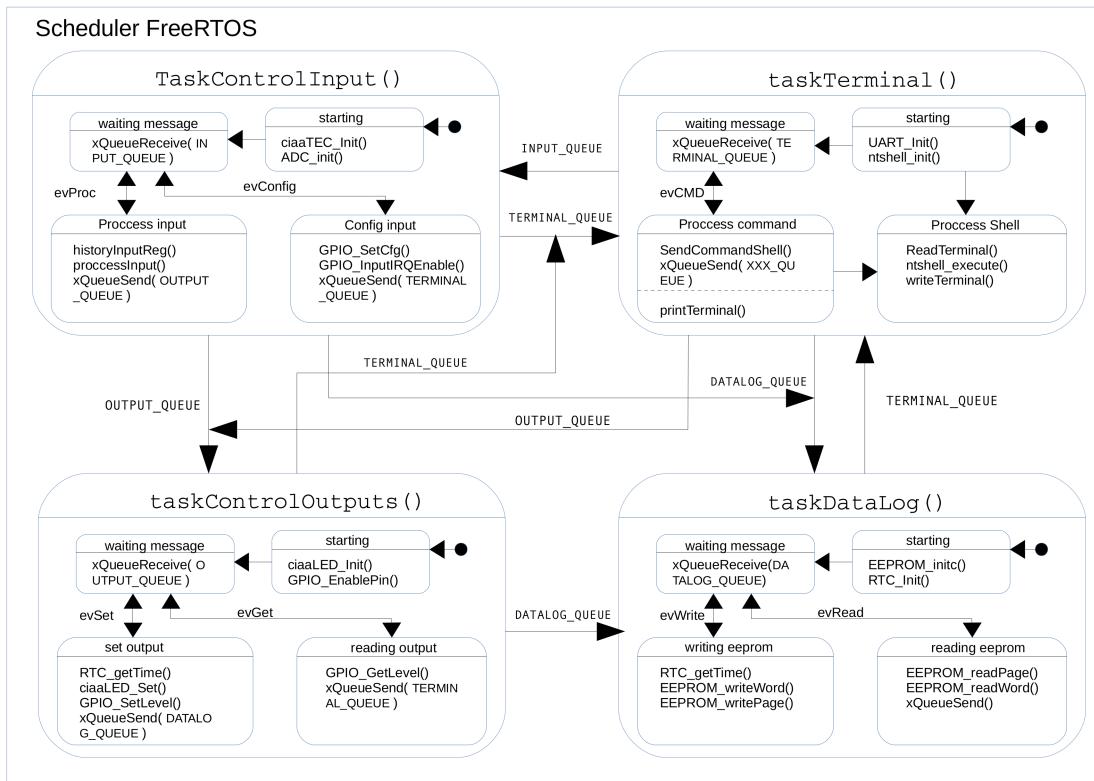


FIGURA 3.12: Interacción entre tareas.

3.3. Sistemas control de entradas

El monitoreo de las entradas se divide en subestados según la opción de funcionamiento elegida, y a la vez se puede encontrar en funcionamiento normal, en estado falla o en reconfiguración. La lectura de las entradas digitales las hace por interrupción mientras que las analógicas las hace por polling segun tiempos por default a través del terminal. La figura 3.13 muestra el diagrama de estados de la tarea.

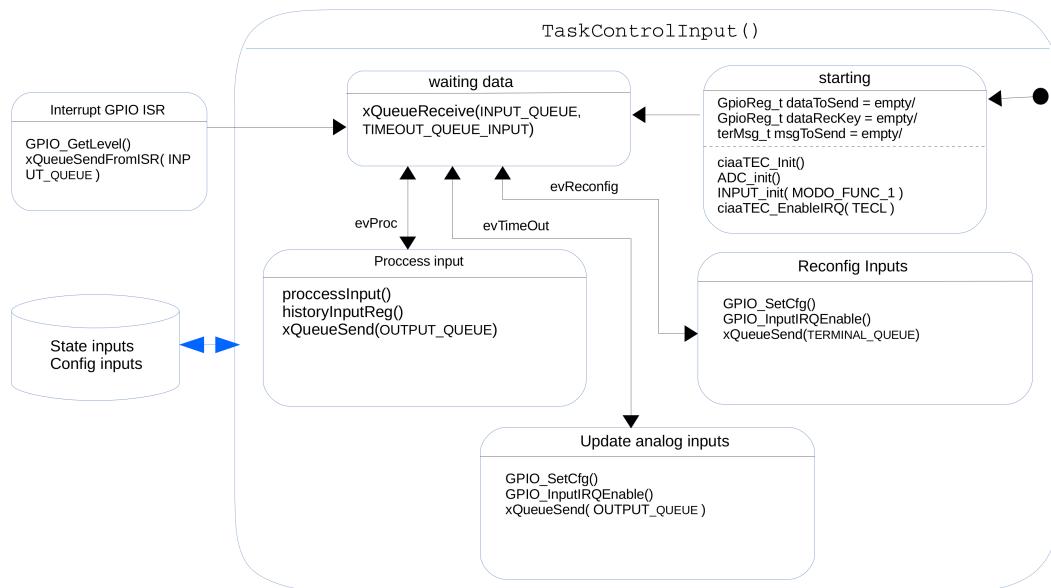


FIGURA 3.13: Diagrama de estados de la tarea control de entradas.

Se observa que a medida que se actualizan los valores de los registros de las entradas, la tarea ejecuta acciones vinculadas con cada una de ellas. Luego en función de estas acciones evalúa si debe enviar un mensaje hacia otra de las tareas, siempre y cuando la tarea destino esté habilitada y en modo activo. El manejo de los temporizadores de duración de cada etapa es implementado por esta tarea, ya que es dependiente del estado de entradas digitales.

3.4. Sistemas control de salidas

El control de las salidas está ligado expresamente a las acciones externas. Las acciones llegan por cambios ocurridos en las entradas o pueden ser forzadas a través del terminal de PC. Por esta razón éste modulo no es pausado en ningún momento del funcionamiento.

Las acciones que activan las salidas están parametrizadas en un registro interno del módulo. Cuando la tarea verifica que se debe ejecutar un cambio en una salida entonces verifica también si debe reenviar algún parámetro hacia otra tarea. La figura 3.14 muestra el diagrama de estados de la tarea.

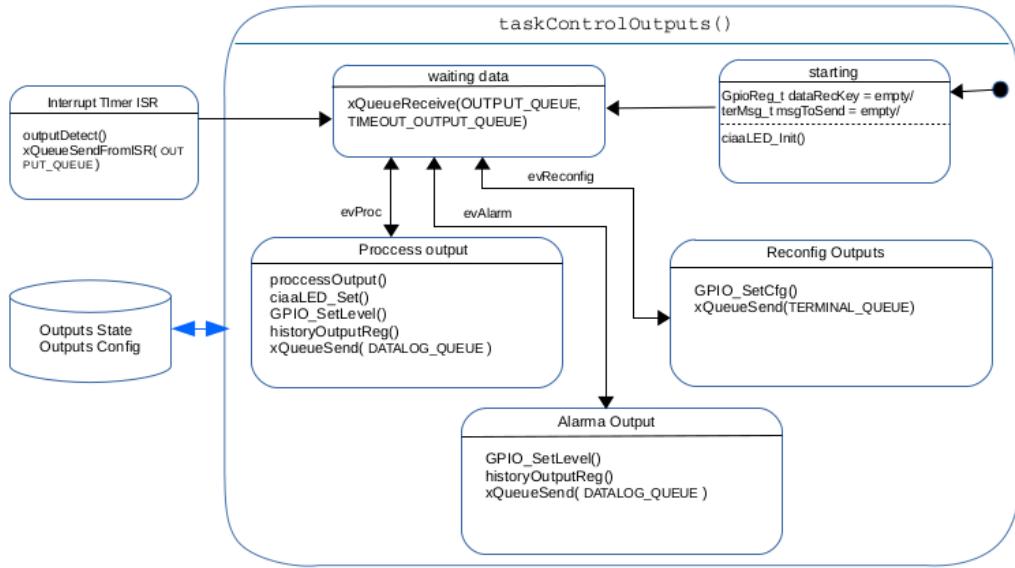


FIGURA 3.14: Diagrama de estados de la tarea control de salidas.

3.5. Sistemas control de terminal-Shell

Este modulo se encarga de emular la terminal con la PC procesando comandos y consultas. Esta tarea es opcional para el sistema pero es importante para obtener información de históricos y realizar test de los puertos del hardware. Originariamente se pensó también que la misma ajustaría los parámetros de alarma y de tiempos por etapa a partir del ingreso de información del lote a procesar. Luego como se determinó que el sistema fueran modular este concepto no es aplicable. No obstante se pueden ajustar parámetros de tiempos y de niveles de alarma.

La figura 3.15 muestra el diagrama de estados de la tarea.

Los estados están predeterminados por los comandos implementados de los cuales se puede agregar hasta 255. En la tabla 3.1 se resumen los comandos implementados.

3.6. Sistemas gestión de datos

Se encarga de manejar los registros que deben ser guardados y leídos de la memoria EEPROM interna. Los datos a ser grabados son leídos de espacios de memoria RAM compartidos y los datos leídos son almacenados en un buffer compartido de la tarea para luego ser enviados a la tarea solicitante. La figura 3.16 muestra el diagrama de estados de la tarea.

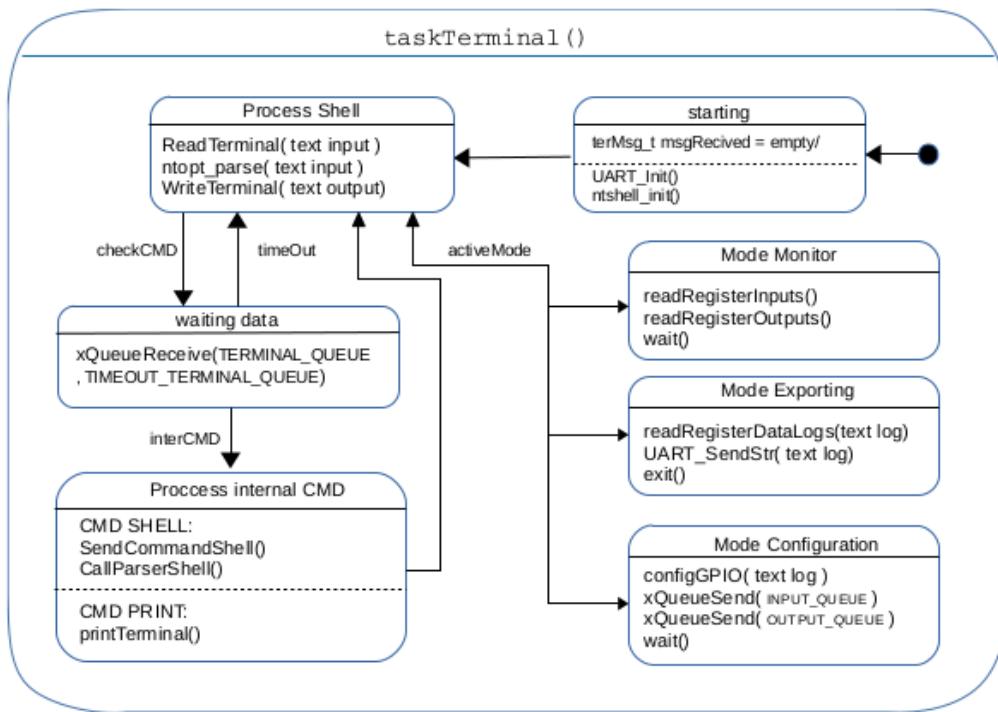


FIGURA 3.15: Diagrama de estados de la tarea terminal de comunicación.

Comando	Función	Descripción	Parámetros
<i>hist</i>	Registros históricos	Devuelve una lista con los últimos registros guardados en memoria.	Ninguno
<i>mon</i>	Monitoreo de puertos	Muestra el valor y la variación de las entradas y salidas del dispositivo.	Ninguno.
<i>conf</i>	Configuración de parámetros	Configura valores máximo, mínimo y tiempos de muestreo de un pin.	nXmax, nXmin, tXmax, tXmin. X: Número de parámetro. n: Nivel. t: Tiempo.
<i>test</i>	Testing de puertos	Modifica el valor de una entrada para evaluar el comportamiento del sistema.	PX val, donde P=I/O, X=id puerto, val= valor del puerto..
<i>time</i>	Configura el calendario	Modificar la hora y fecha del calendario interno usado en la grabación de históricos.	c DDMMAA, calendario día(DD), mes(MM), año(AA).t HHMMSS, tiempo hora(HH), minutos(MM), segundos(SS).

TABLA 3.1: Descripción tablas de comandos.

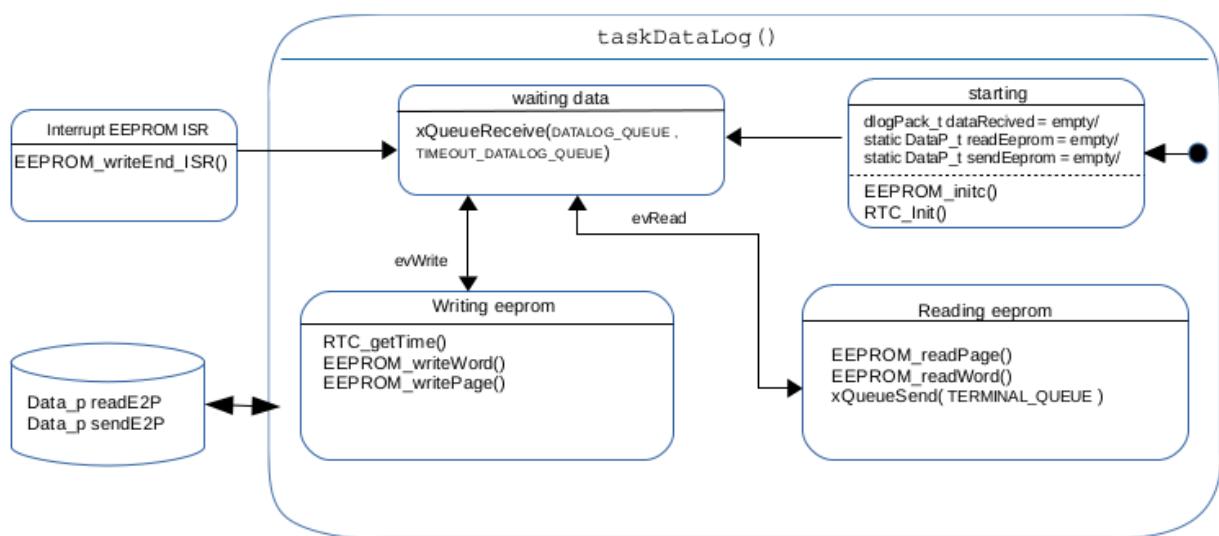


FIGURA 3.16: Diagrama de estados de la tarea manejo de datos en memoria.

Capítulo 4

Ensayos y Resultados

Para la validación del prototipo se realizaron distintas pruebas en hardware y el software a fin de cubrir los requerimientos explicados en 2.3.3. Aquí se desarrollan las resultados para ambos partes de manera separada.

4.1. Pruebas funcionales del firmware

El firmware esta basado en módulos propios mas módulos de terceros. En la figura 4.1 se muestra la organización del repositorio [8] con los módulos de rutinas incluidos en cada uno. La estructura del firmware en si se basa en el repositorio utilizado en la materia de *Sistemas Operativos de Tiempo Real* [7].

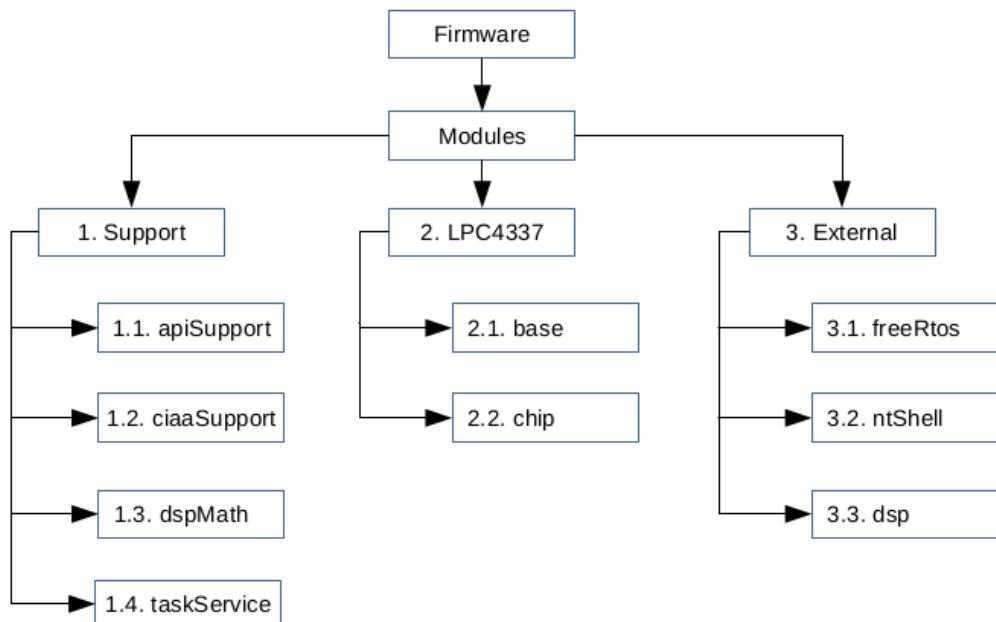


FIGURA 4.1: Estructura de directorios del proyecto.

En sub-rama *support* se encuentran las rutinas de desarrollo propio, que consisten en:

- *apiSupport*: rutinas API (Application Programming Interface) interfaz de programación de aplicaciones.

- *ciaoSupport*: rutinas de soporte del hardware propio de la EDU-CIAA.
- *dspMath*: rutinas de soporte matemático a operaciones de procesado y conversión de señales físicas.
- *taskServices*: rutinas principales de los servicios o tareas del programa.

El directorio *LPC4337* básicamente se componen de las librerías del fabricante denominada *LPC-OPEN* [5]. Allí se encuentran los subdirectorios *chip* y *base* donde se alojan los módulos de capa HAL¹ y capa CMSIS² respectivamente.

En la ubicación *External* se encuentran los módulos con librerías de terceros. En principio se muestran solo los que se usaron, los que no se dejan para futuras implementaciones. Se utilizaron los siguientes módulos:

- *freeRTOS*: sistema operativo de tiempo real [1].
- *ntShell*: natural tiny Shell [3].
- *dsp*: rutinas de filtrado digital de señales.

4.1.1. Pruebas unitarias

Se realizaron sobre los módulos propios ubicados en el directorio *Support* del repositorio. Los mismos interactúan entre los servicios de la aplicación y las rutinas de capa HAL. Sobre la capa HAL y los librerías de terceros no se realizaron pruebas, se confió en el funcionamientos de los mismos.

A fin de validar el comportamiento de los algoritmos de procesamiento de los sensores de temperatura se realizaron pruebas sobre archivos con datos de entradas y se generaron archivos de salida con los resultados obtenidos. En la figura 4.2 se observa los tipos de archivos que se procesan.

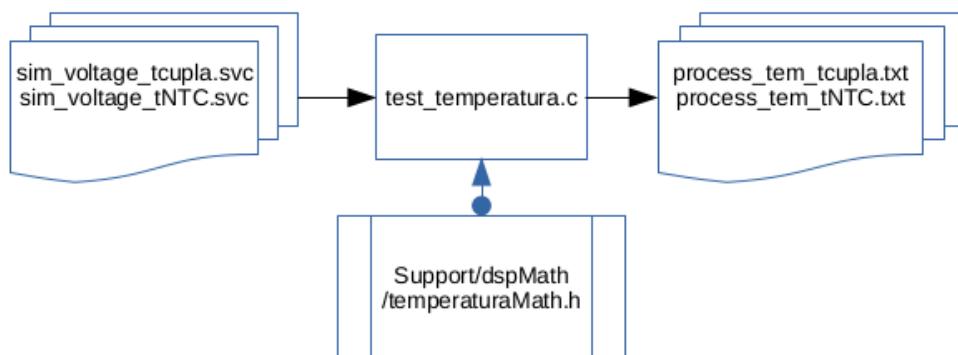


FIGURA 4.2: Test de funciones interfaz de temperatura.

La escritura en memoria interna se probó un procedimiento de emulación de datos con archivos de texto plano de entrada y salida, validando la escritura de bloques binarios según los formatos establecidos en la lógica del programa. En la figura 4.3 se observa los archivos procesados en el test.

¹HAL: Hardware Abstraction Level.

²CMSIS: Cortex Microcontroller Software Interface Standard.

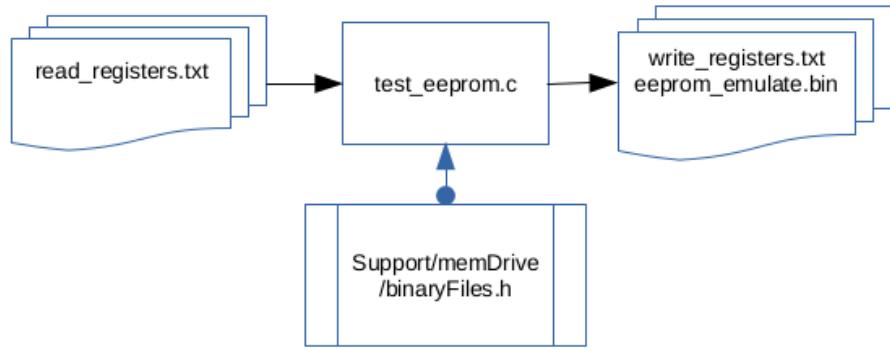


FIGURA 4.3: Test de funciones interfaz de memoria.

Para estas pruebas se predefinieron los archivos de ingreso de datos de manera de simplificar la lectura de los bloques de información. En las figuras 4.5 y 4.4 se observan la estructura de los archivos procesados.

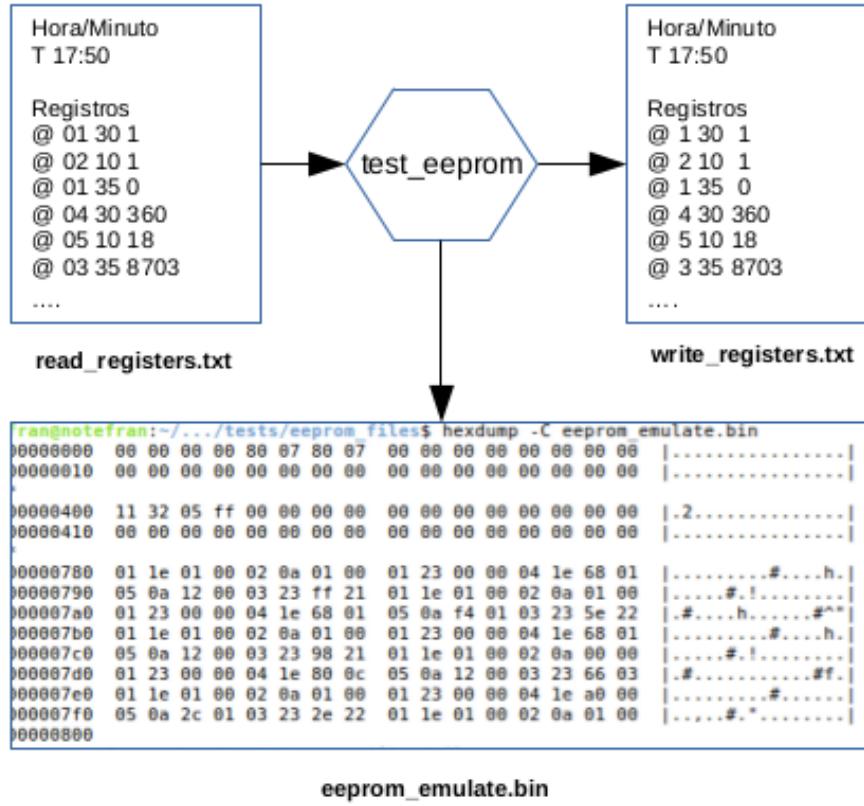


FIGURA 4.4: Archivos procesados en el test de temperatura.

4.2. Ensayos de validación hardware

Para esto se realizaron las pruebas sobre los puertos de la EDU-CIAA emulando el comportamiento de los sensores en función de los modos de funcionamiento preestablecidos. En la figura 4.6 se observa el banco de pruebas que emula las entradas y salidas que se tendrían en el entorno real.

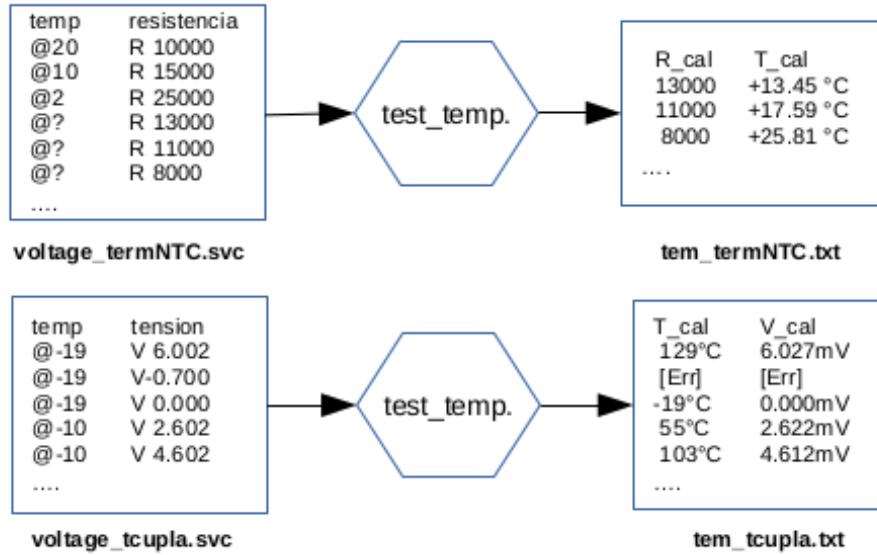


FIGURA 4.5: Archivos procesados en el test de memoria.

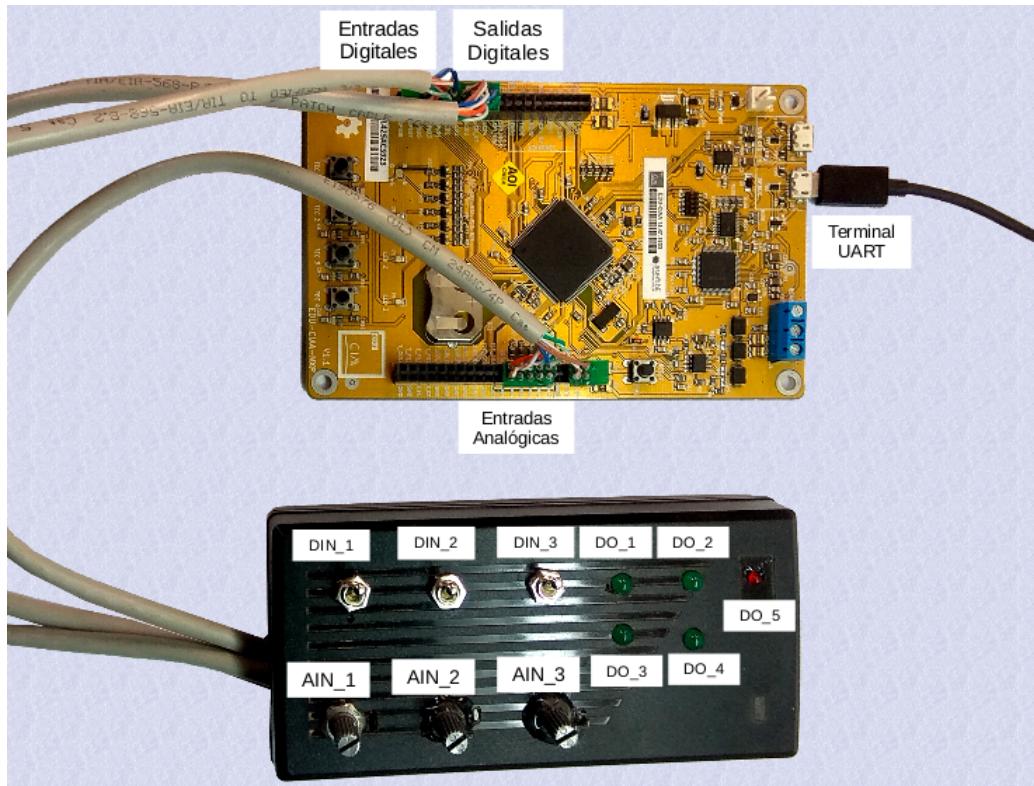
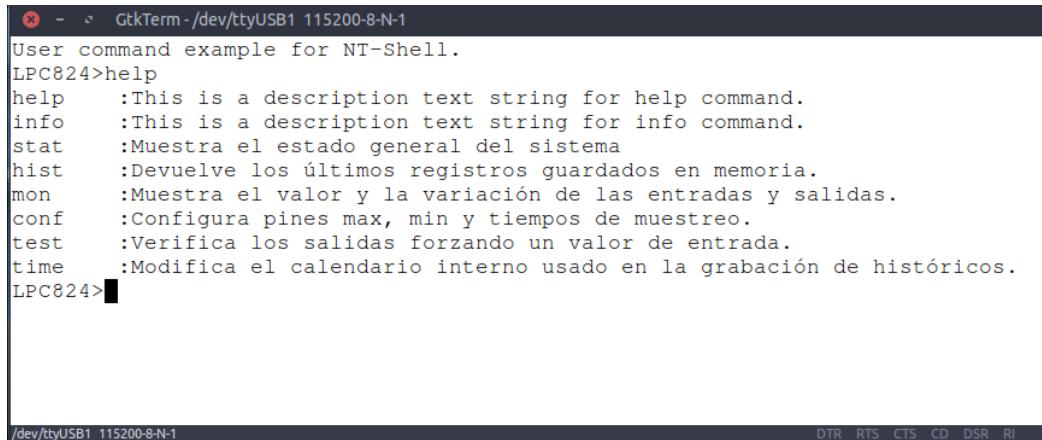


FIGURA 4.6: Banco de pruebas.

4.2.1. Pruebas de interfaz

Para esto simplemente se ingresaron los comandos establecidos y verificaron que los mismos ejecuten correctamente las acciones internas así también como los mensajes por la terminal. Se tomaron luego capturas de los estados de salida así como de la terminal y de log registrado por el sistema. En las figuras ?? se muestran las ejecuciones de algunos de ellos.

Se probó el funcionamiento de los comandos vía consola y su impacto sobre las configuraciones de los puertos y la devolución de estados por pantalla.



The screenshot shows a terminal window titled "GtkTerm - /dev/ttyUSB1 115200-8-N-1". The window displays a help menu for the NT-Shell. The commands listed are:

- help : This is a description text string for help command.
- info : This is a description text string for info command.
- stat : Muestra el estado general del sistema
- hist : Devuelve los últimos registros guardados en memoria.
- mon : Muestra el valor y la variación de las entradas y salidas.
- conf : Configura pines max, min y tiempos de muestreo.
- test : Verifica los salidas forzando un valor de entrada.
- time : Modifica el calendario interno usado en la grabación de históricos.

The terminal prompt is "LPC824>". At the bottom of the window, there is a status bar showing "/dev/ttyUSB1 115200-8-N-1" and a set of serial port control indicators: DTR, RTS, CTS, CD, DSR, RI.

FIGURA 4.7: Lista de comando implementados.

4.3. Ensayos de integración

A fin de validar que el sistema completo, se realizaron pruebas de cada una de las partes del prototipo. Se probaron utilizando el módulo externo de simulación de entradas y salidas mas la terminal serial de pc conectado. A partir de la pre-configuración de los puertos del prototipo realizando variaciones en las entradas y verificando las salidas y los mensajes por la terminal.

4.3.1. Validación de requerimientos

Para esto se tomaron los criterios desarrollados conjuntamente con los requerimientos previamente en la materia *Gestión de Proyectos* del posgrado. Básicamente consiste en variar los valores de los entradas al prototipo y verificar que cumpla con la respuesta adecuada. En la tabla 4.1 se observa un resumen del cumplimiento de los requerimientos funcionales y de interfaz. Para los mismos se evaluó si cumple si o no, y en caso negativo porque no.

Los requerimientos a Futuro (RAF) no se han agregado a la tabla pero fueron considerados durante el desarrollo del prototipo como un sistema distribuido. Esta arquitectura fue descartada y dejada para más adelante en función de los resultados obtenidos con este primer diseño.

A su vez muchos de los requerimientos originales estaban asociados a realizar un sistema distribuido de control con módulos de entradas y salidas interconectados a la EDU-CIAA. Si bien ese proyecto fue modificado los requerimientos se conservaron a modo ilustrativo de como se transformó el proyecto desde su idea inicial.

Requerimiento	Número	Cumple	Observaciones
RFTEM	1.1.1	SI	Ninguna.
	1.1.2	SI	Ninguna.
	1.1.3	SI	Ninguna.
	1.1.4	SI	Ninguna.
	1.1.5	SI	Ninguna.
	1.1.6	NO	Memoria interna insuficiente.
RFENE	1.2.1	SI	Ninguna.
	1.2.2	NO	Se supone constante 5V.
	1.2.3	NO	Memoria interna insuficiente.
	1.2.4	SI	Por ahora solo se pueden ingresar comandos.
RFCOND	1.3.1	SI	Ninguna.
	1.3.2	SI	Ninguna.
RFTI	1.4.1	NO	No implementado sistema distribuido.
	1.4.2	SI	Ninguna.
	1.2.3	NO	No implementado sistema distribuido.
RFNB	1.5.1	SI	Ninguna.
	1.5.2	SI	Ninguna.
RFHMI	1.6.1	SI	Ninguna.
	1.6.2	NO	No implementado sistema distribuido.
	1.6.3	SI	Ninguna.
	1.6.4	NO	No implementado sistema distribuido.
	1.6.5	NO	No implementado sistema distribuido.
RITEM	2.1.1	SI	Ver sección 3.1.
	2.1.2	SI	Ver sección 3.1.
RIENE	2.2.1	SI	Ninguna.
	2.2.2	NO	Se supone parámetro constante 5V.
RICOND	2.3.1	SI	Ninguna.
	2.3.2	SI	Ninguna.
RINB	2.4.1	SI	Ninguna.
RIHMI	2.5.1	SI	Ninguna.
	2.5.2	SI	Solo con comandos.
	2.5.3	SI	Ninguna.
	2.5.4	SI	Ninguna.
	2.5.5	SI	Ninguna.
	2.5.6	NO	Se puede implementar en la misma plataforma a futuro.
RNF	3.1	NO	No se construyó banco de pruebas.
RD	4.1	SI	Ver sección 3.1.

TABLA 4.1: Cumplimiento de los requerimientos.

4.3.2. Pruebas de rendimiento

Se probó la estabilidad del mismo dejando en funcionamiento durante un día completo, de esta manera se tiene mejor información de la performance de tiempo de uso y memoria de los servicios en el sistema operativo. En la tabla 4.2 se observa el resultado obtenido en uso de memoria y tiempo de CPU.

Id	Nombre	CPU(%)	Memoria(Byte)
0	Task Idle	95	128
1	Task Outputs	1	256
2	Task Inputs	1	256
3	Task Terminal	3	512
4	Task Eeprom	1	256

TABLA 4.2: Uso de recursos por tarea.

Considerando que este microprocesador (LPC-4337) tiene 32KB de memoria RAM, con mas la mitad disponible las tareas solo consumen 1408 Bytes y menos del 4 % del CPU. Por lo tanto se tienen recursos de sobra para poder seguir implementando servicios y funciones en la plataforma.

Capítulo 5

Conclusiones

En este ultimo capitulo se vuelcan las conclusiones acerca del prototipo y la funcionalidad lograda. También se analizan si las opciones en el proceso de desarrollo fueron las correctas. Finalmente se describen los cambios y mejoras que se le realizaran a futuro a fin de mejorar las funcionalidades y optimizar la mantenibilidad del mismo.

5.1. Conclusiones generales

Se describen las conclusiones separando los entornos de hardware y de software.

5.1.1. Hardware del prototipo

A nivel particular de la implementación realizada se logro cumplir con gran parte de los requisitos originales y con la mayoría de los requisitos originados para el prototipo final. El sistema originalmente se pensó para ser implementado sobre un microprocesador con menores prestaciones que el del micro de la EDU-CIAA. Como sobre éste demostró estar holgado de recursos, se concluye que la transición hacia una plataforma mas reducida es realizable.

Los elementos seleccionado como interfaz de hardware demostraron ser suficientes para integrar en una placa dedicada, de esta manera se pueden reducir los costos al no usar módulos con soluciones de terceros que muchas veces elevan altamente los costos de los sensores de campo.

5.1.2. Firmware del prototipo

El presente sistema desarrollado sobre una necesidad especifica permitió lograr un sistema base de control y monitoreo de parámetros físicos en un proceso de fabricación. El firmware a su vez cuenta con los siguientes características mas sobresalientes:

- Es un sistema simple y modularizado en tareas independientes, lo cual lo hace mantenable y testeable.
- Los procesos que funcionan sobre el sistema operativo están pensado para ser compilados de forma opcional.

- Se pueden incrementar los servicios o quitarlos en función de modificar su funcionalidad.
- Tiene gran disponibilidad de tiempo de procesador y de memoria para agregarle funcionalidad.
- Es portable hacia otras plataformas de LPC M4 con solo modificar las librerías lpc open.

5.1.3. Proceso de desarrollo

A nivel del proceso de desarrollo se pueden definir una serie de hitos positivos y otro negativos. Como positivos están:

- Ajustarse a la estructura del proceso de análisis de sistema, casos de uso y requerimientos permitió una trazabilidad entre las características del prototipo.
- Se logró un producto mínimo viable óptimo.
- Utilizar un sistema de control de versiones (GIT) del firmware permitió tener un seguimiento claro de las funcionalidades logradas durante el proceso.
- Diseñar el firmware en módulos independientes permitió lograr un análisis de calidad del mismo de manera mas simple y confiable.

Como rasgos negativos se pueden destacar:

- Dedicarle extenso tiempo hasta la definiciones específicas de los requerimientos resultó contraproducente por las demoras para iniciar con el proyecto.
- Implementar los módulos de interfaces con sensores de temperatura implicó mas tiempo del pensado.
- No haber desarrollado una arquitectura completa del sistema antes de iniciar produjo mas demoras que si se hubiera modelizado integro desde un principio.

5.2. Próximos pasos

A nivel de software quedaron varios puntos abiertos para darle mas prestaciones. Entre ellos se implementaran:

- Agregar mas comandos sobre la comunicación vía terminal-UART para tener mas prestaciones.
- Desarrollar servicios de comunicación vía I2C para interactuar con módulos externos, particularmente sensores.
- Desarrollar servicios de comunicación MODBUS-RS485 para interactuar con otros plataformas.

- Implementar un módulo de servicios UDP-Ethernet para poder comunicarse a futuro con un sistema central tipo SCADA.
- Mejorar los módulos desarrollados estandarizando más las nombres de las rutinas y reestructurando archivos headers.
- Agregar más protocolos de prueba y validación para poder agregarlos a futuro a un servidor de verificación tipo Jenkins.

A nivel de hardware se piensan a futuro los siguientes pasos:

- Probar la portabilidad de sistema hacia una plataforma con ARM M4 de prestaciones reducidas.
- Elaborar el "poncho" de la EDU-CIAA como se pensó originalmente pero agregándole integrado más acordes a los que se pueden conseguir en el mercado local.

Apéndice A

Casos de Uso

Referencias:

- Ref 1- Disparadores: Evento comienza el caso de uso.
- Ref 2 - Flujo básico: Pasos del escenario, desde que es disparado hasta que alcanza su objetivo.
- Ref 3 - Flujo alternativo: Pasos alternativos al flujo básico.
- Ref 4- Pre-Condiciones: Condiciones que deben estar presentes para que se pueda iniciar el caso de uso.
- Ref 5- Pos-Condiciones: Condiciones que deben estar presentes para que se pueda finalizar el caso de uso.
- Ref 6- Modo 1: Medición de conductividad, nivel y control de temperatura.
- Ref 7- Modo 2: Medición de corriente, tensión, nivel y control de temperatura.

Título	Descripción
1. Nombre	Puesta en alta del dispositivo.
1.1. Descripción	Secuencia de instalación de dispositivo en las cubas de control.
1.2. Actor principal	Técnico instalador.
1.3. Disparadores	Alimentación de la unidad.
2. Flujo de eventos	
2.1. Flujo básico	<p>2.1.1. El sistema inicia el ciclo de encendido y se prepara para evaluar los rangos de las entradas de una por vez. 2.1.4 Censa que los de los termostatos sean los correctos. (a definir)</p> <p>2.1.2. Para cada uno emite una señal a definir en caso de ser correcto y otra en caso de ser incorrecto.</p> <p>2.1.3. Censa que los valores de corriente sobre los lazos de control de conductividad sean los correctos, según el máximo y el mínimo admisible.(a definir)</p> <p>2.1.4. Censa que los valores de los termostatos sean los correctos. (a definir)</p> <p>2.1.5. Censa los valores de las entrada de nivel, que deben estar dentro de los establecidos según RINB242?</p>
2.2. Flujo alternativo	2.1.6. Si alguno de los sensores no esta dentro de los valores se bloquea el proceso hasta que se seleccione ignorar o se normalice el parámetro.
3. Requerimientos especiales	Botón de encendido liberado. Selector de modo de funcionamiento en posición.
4. Pre Condiciones	Interfaces de sensores conectados.
5. Pos Condiciones	El dispositivo queda en modo espera a la señal de inicio de ciclo, midiendo según los requerimientos RFTEM112,121,131,153? y mostrándolos por uart los valores de cada uno.

TABLA A.1: Descripción caso de uso puesta en alta

Título	Descripción
1. Nombre	Puesta en funcionamiento en Modo 1
1.1 Breve descripción	Secuencia de control de un ciclo completo dentro de la cuba controlada
1.2 Actor principal	Operario (Op)
1.3 Disparadores	Pulsación de botón inicio
2. Flujo de eventos	
2.1. Flujo básico	<p>2.1.1. El sistema detecta la presencia de las placas con el detector colocado en la cuba.</p> <p>2.1.2. El sistema inicia el conteo del tiempo preestablecido para esa etapa utilizando un indicador luminoso para indicar al Op.</p> <p>2.1.3. El sistema continua controlando los parámetros de temperatura y nivel durante todo el ciclo y transmitiendo los valores por uart.</p> <p>2.1.4. Completado el ciclo el sistema emite una señal al Op para que se disponga a retirar las placas de la cuba.</p>
2.2. Flujo alternativo	<p>2.2.1. El sistema no detecta la placa con los sensores de presencia y no inicia el conteo de tiempo.</p> <p>2.2.2. Si se colocan las placas o se anula la entrada del sensor el proceso vuelve a 2.1.2 del flujo básico.</p>
2.3. Flujo alternativo	<p>2.3.1. El Op retira las placas de la cuba antes de completado el tiempo reglamentario. El sistema emite una señal de alarma para que se restablezca la placa durante 15 segundos.</p> <p>2.3.2. Si se restablece la placa antes de cumplirse los 15 segundos se retorna al paso normal en el que se encontraba. Sino vuelve el contador de tiempo a cero y deja un indicador de alarma encendido hasta que se lo anule manualmente y vuelve al punto 2.1.5 del flujo básico.</p>
2.4. Flujo alternativo	<p>2.4.1. El sistema detecta que la temperatura o el nivel fuera de valor nominal por mas de 5 minutos emite una alarma y continua el proceso normal hasta 2.1.5.</p> <p>2.4.2. Finalizado el tiempo no permite iniciar un nuevo ciclo hasta que las parámetros de control vuelvan a sus valores nominales.</p>
3. Requerimientos especiales	El sistema debe estar configurado en el Modo 1.
4. Pre condiciones	Sistema energizado e inicializado.
5. Pos condiciones	Queda en modo espera controlando los parámetros según los requerimientos RFTEM112,121,131,153?

TABLA A.2: Descripción caso de uso en Modo 1

Título	Descripción
1. Nombre	Puesta en funcionamiento en Modo 2.
1.1 Breve descripción	Secuencia de control de un ciclo completo dentro de la cuba controlada.
1.2 Actor principal	Operario (Op)
1.3 Disparadores	Pulsación de botón inicio.
2. Flujo de eventos	
2.1 Flujo básico	<p>2.1.1. El sistema detecta la presencia de las placas con el detector colocado en la cuba.</p> <p>2.1.2. El sistema detecta la umbral y comienza a integrar el valor hasta llegar al total necesario.</p> <p>2.1.4. Completado el ciclo el sistema emite una señal al Op para que se disponga a retirar las placas de la cuba.</p> <p>2.1.2. El sistema detecta la corriente umbral y comienza a integrar el valor hasta llegar al total necesario.</p> <p>2.1.3. El sistema inicia un indicador luminoso para indicar al Op que la etapa está en proceso.</p>
2.2. Flujo alternativo	2.2.1. El valor de temperatura se va de rango por más de 10 minutos, el sistema emite una alarma sin interrumpir el proceso.
2.3. Flujo alternativo	<p>2.3.1. Si se retiran las placas de la cuba antes de que el proceso finalice, o el valor de corriente cae debajo de un valor mínimo aceptable por más de 1 minuto se considera una situación anormal y se emite una alarma.</p> <p>2.3.2. Si no se restablece el parámetro por más de 30 segundos los contadores se vuelven a cero y se deja una alarma encendida, sino retorna al proceso al momento en que fue interrumpido.</p>
3. Requerimiento especial	El sistema debe estar configurado en el Modo 2.
4. Pre condiciones	Sistema energizado e inicializado.
5. Pos condiciones	Queda en modo espera controlando los parámetros según los requerimientos RFTEM112,121,131,153?

TABLA A.3: Descripción caso de uso en Modo 2

Apéndice B

Esquemáticos

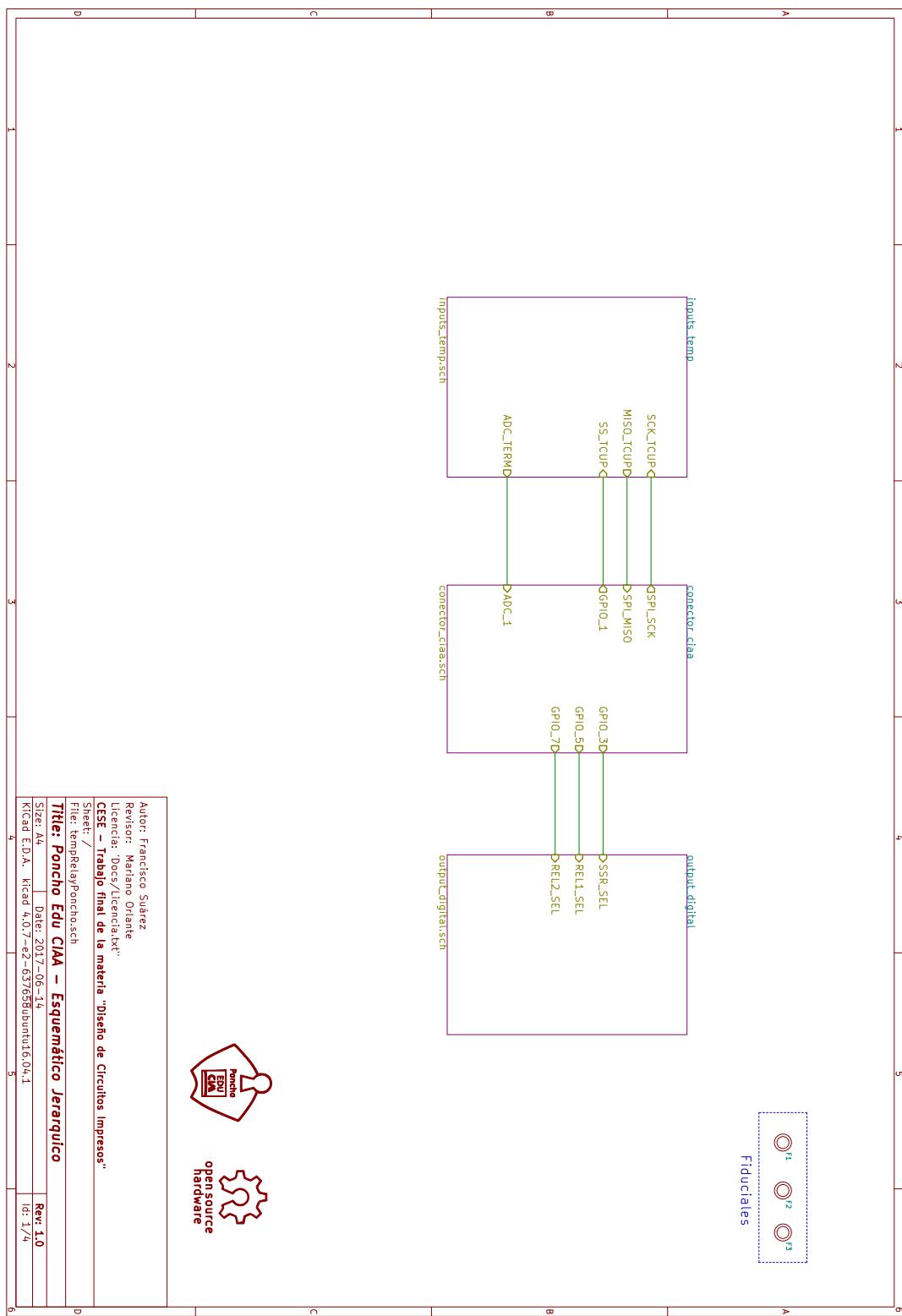


FIGURA B.1: Esquemático jerárquico del poncho EduCiaa.

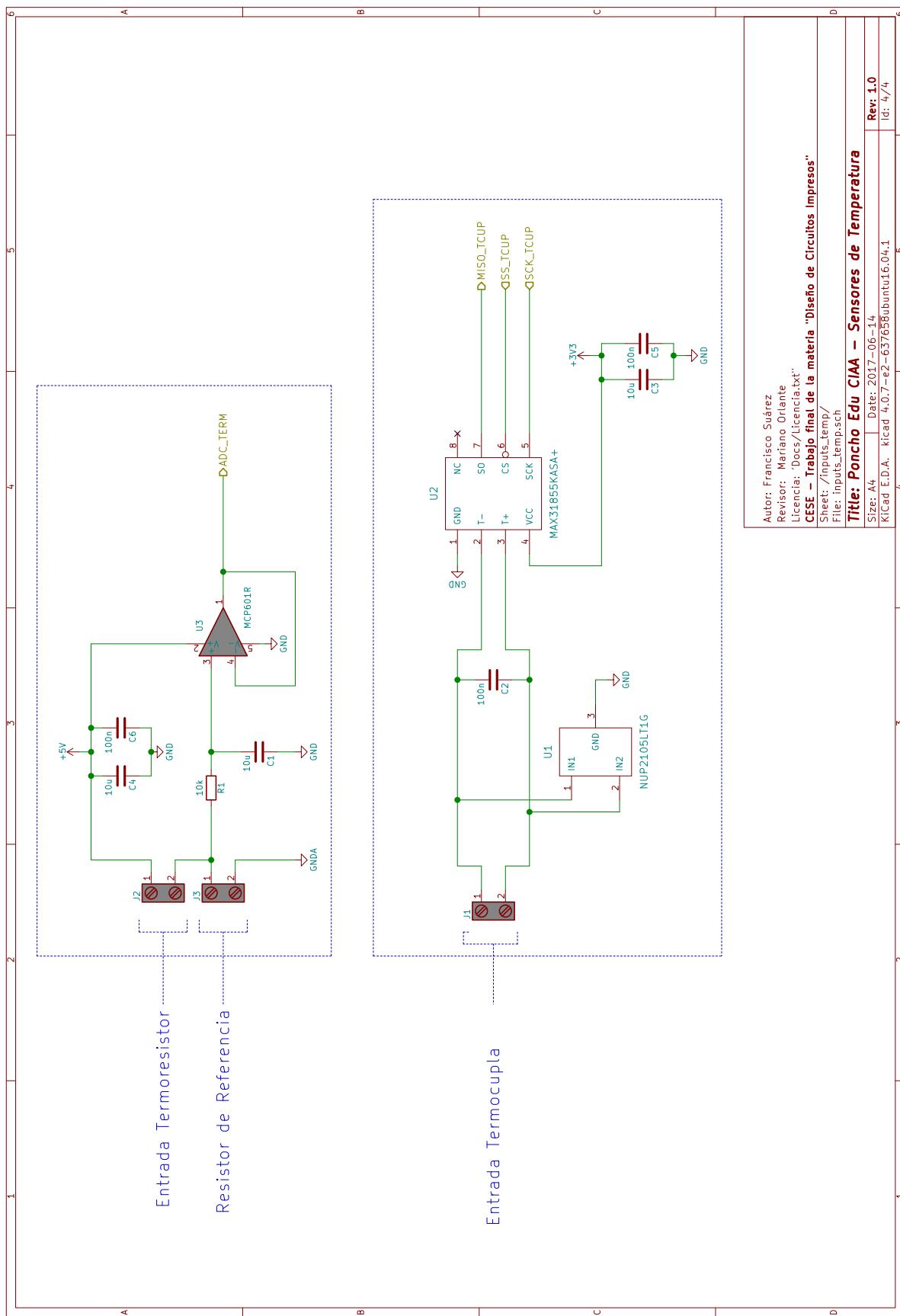


FIGURA B.2: Esquemático entradas de termocupla y termistor.

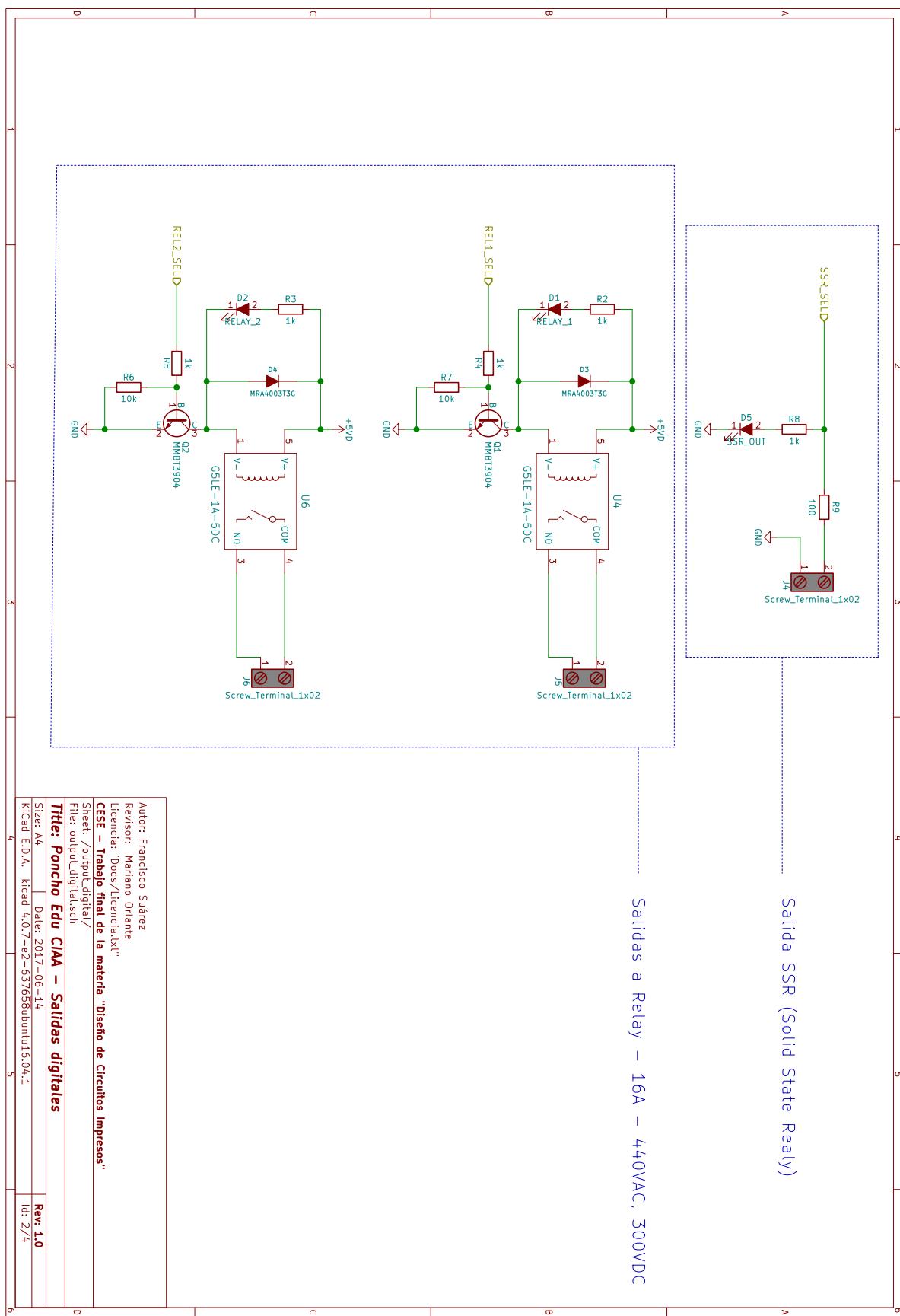


FIGURA B.3: Esquemático de salidas digitales con relays.

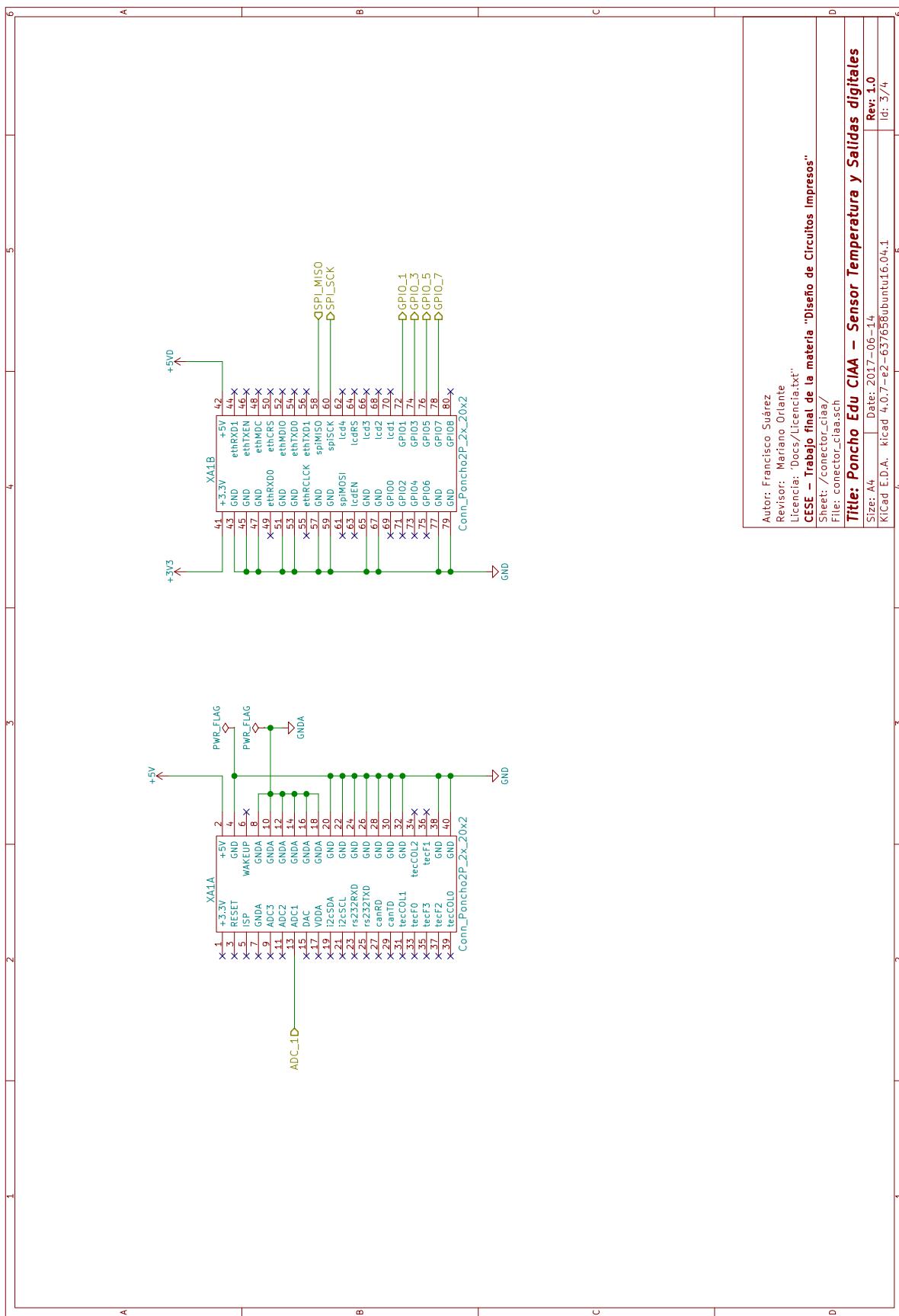


FIGURA B.4: Esquemático de conectores de expansión con la eduCIAA.

Bibliografía

- [1] Richard Barry. *The FreeRTOSTM Kernel*. Disponible: 2016-06-25. URL: <https://www.freertos.org/>.
- [2] Ing. Diego Brengi. *Repositorio de Ponchos para la EDU-CIAA*. Disponible: 2018-11-25. URL: <https://github.com/brengi/Ponchos>.
- [3] cubeatsystems.com. *Natural Tiny Shell*. Disponible: 2016-06-25. URL: <https://www.cubeatsystems.com/ntshell/>.
- [4] Microchip Technology Inc. *Simplified Thermocouple Interfaces and PICmicroAN844*. Disponible: 2016-06-25. URL: <http://ww1.microchip.com/downloads/en/AppNotes/00844a.pdf>.
- [5] NXP Semiconductors. *LPCOPEN v2.16 Drivers, Middleware and Examples*. Disponible: 2016-06-25. URL: <https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/lpc-cortex-m-mcus/lpc4300-cortex-m4-m0/lpcopen-software-development-platform-lpc43xx:LPCOPEN-SOFTWARE-FOR-LPC43XX>.
- [6] Proyecto CIAA. *Computadora Industrial Abierta Argentina*. Disponible: 2016-06-25. 2014. URL: <http://proyecto-ciaa.com.ar/devwiki/doku.php?id=start>.
- [7] Pablo Ridolfi. *Embedded software development workspace for microcontrollers*. Disponible: 2016-06-25. URL: <https://github.com/pridolfi/workspace>.
- [8] Francisco Suárez. *CIAA Firmware Project*. Disponible: 2016-06-25. URL: <https://github.com/fransuarez/repoTpFinal>.
- [9] KiCad Developers Team. *KiCad open source software suite for Electronic Design Automation*. Disponible: 2018-11-25. URL: <http://kicad-pcb.org/download/>.