

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



MEMORIA DEL TRABAJO FINAL

**Control automático del proceso de
fermentación de vino usando la CIAA**

Autor:
Luis Enrique Chico Capistrano

Director:
Ing. Juan Manuel Cruz

Jurados:
Dr. Ing. Pablo Gomez (FIUBA)
Esp. Lic. Agustín Bassi (FIUBA)
Esp. Ing. Patricio Bos (FIUBA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires, entre enero
de 2016 y agosto de 2017.*

Resumen

En el presente trabajo se implementó un control automático de temperatura y sistema de alertas para la bodega Chico Zossi. Al mantener la temperatura estable, se mejora la calidad del vino, lo que afecta directamente al precio por litro.

La implementación se hizo sobre la CIAA NXP, mediante un sistema operativo de tiempo real y se hizo uso de la herramienta de control de versiones. Para garantizar la calidad del producto fueron empleadas técnicas de gestión de proyectos y ensayos que permitieron validar y verificar el correcto funcionamiento.

Agradecimientos

Agradecimientos personales. [OPCIONAL]

No olvidarse de agradecer al tutor.

Índice general

Resumen	III
1. Introducción General	1
2. Introducción Específica	3
2.1. Descripción del trabajo a realizar	3
3. Diseño e Implementación	5
3.1. Hardware	5
3.1.1. Diagrama en bloque	5
3.1.2. Módulos requeridos	6
3.1.3. Características del hardware	6
3.2. Software	8
3.2.1. Aplicación	8
3.2.2. Web	11
4. Ensayos y Resultados	17
4.1. Banco de pruebas	17
4.2. Configuraciones para la PC	18
4.3. Verificación del sistema web embebido	19
4.4. Pruebas funcionales del hardware	21
5. Conclusiones	25
5.1. Conclusiones generales	25
5.2. Próximos pasos	25
Bibliografía	27

Índice de figuras

2.1. Listado de tareas, y planificación de tiempos de la implementación.	4
3.1. Esquema de interacción del dispositivo.	6
3.2. Módem SIM800L.	7
3.3. Módulo SIM800L e índice de esquemáticos.	7
3.4. Fuente de alimentación DC/DC.	8
3.5. Regulador de corriente, de tensión y del sensor de temperatura.	8
3.6. Entradas a leds y salidas de los pulsadores.	9
3.7. Capa superior de la placa simulador de sensores.	9
3.8. Capa inferior de la placa simulador de sensores.	10
3.9. Placa simulador de sensores 3D.	10
3.10. Placa de simulación de temperatura y estado de la batería con el modulo SIM800L integrado.	11
3.11. Diagrama de flujo de la tarea que controla el módem GPRS.	12
3.12. Diagrama de flujo de la tarea de control.	13
3.13. Diagrama de flujo del sensor.	13
3.14. Primera versión de la web de monitoreo.	15
3.15. Versión actualizada haciendo uso del template.	15
3.16. Diagrama de secuencia AJAX.	16
4.1. Placa de simulación de temperatura y estado de la batería con el modulo SIM800L integrado.	17
4.2. Placa básica integrada con la CIAA-NXP.	17
4.3. Configuración de una red estática para la interfaz Ethernet.	18
4.4. Pantalla principal de la web de monitoreo.	18
4.5. Opciones del al presionar sobre el menú desplegable.	19
4.6. Activar/Desactivar actuadores.	20
4.7. Configurar nivel de descarga para accionar la alerta por batería baja.	20
4.8. Configuración de los celulares de quienes van a recibir las alarmas por SMS.	20
4.9. Configuración del rango de temperatura y activación del estado de control automático y alarma.	21
4.10. Formulario de configuración de IP, mascara y puerta de enlace.	21
4.11. Activación por control automático debido a temperatura elevada.	22
4.12. SMS recibidos debido a temperatura máxima. El segundo SMS recibido es por temperatura mínima. Y el último recibido debido a batería baja.	23
4.13. Temperatura en nivel mínimo, se desactiva la bomba y la electroválvula.	23
4.14. Activación manual de actuador 1 y 2, electroválvula y bomba inactivo.	24

Índice de Tablas

2.1. Comparación tiempo planificado vs real.	3
3.1. Configuración por default del servidor	16

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción General

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


Capítulo 2

Introducción Específica

En este sección se detallaran las tareas que se realizaron, que contratiempos se tuvieron y cuales fueron los requerimientos.

2.1. Descripción del trabajo a realizar

Para llevar el proyecto a cabo se desarrollo la siguiente planificación de tareas y se estimaron que tiempos debía emplearse para cada una de ellas.

Pero debido a la falta de experiencia en determinados temas, que fueron más complejos de lo que se estimo. Es por ello que en la siguiente tabla 2.1, se va a mostrar el tiempo real y cuales fueron las tareas que requirieron más tiempo de lo planificado.

Nro	Tarea	Tiempo		% a más
		Estimado	Real	
9	Soldado y verificacion de la placa	15h	24	60
18	Programacion de funciones p/ethernet	30h	50h	66,7
23	Programacion de funciones p/ethernet	25h	35h	40

TABLA 2.1: Comparación tiempo planificado vs real.

	Task Name	Duration	Start	Finish	Predece ssors
1	Inicio	0 horas	04/04/16	04/04/16	
2	Planificación	40h	09/04/16	29/05/16	1
3	Análisis del proyecto	40h	04/06/16	24/07/16	2
4	Diagrama en bloques.	40h	30/07/16	18/09/16	3
5	Diseñar el circuito electrónico para placa base.	35h	24/09/16	06/11/16	4
6	Verificación del circuito.	5h	12/11/16	13/11/16	5
7	Elaboración del PCB.	35h	10/12/16	22/01/17	8
8	Compra de componentes.	15h	19/11/16	04/12/16	6
9	Soldado y verificación de la placa.	15h	28/01/17	12/02/17	7
10	Diagrama en bloques de software de testeo	4h	24/09/16	25/09/16	4
11	Programación de los bloques.	30h	25/09/16	06/11/16	10
12	Verificación del correcto funcionamiento.	4h	18/02/17	19/02/17	11; 9
13	Set up de plataforma de desarrollo	10h	24/09/16	02/10/16	4
14	Definición de arquitectura	5h	08/10/16	09/10/16	13
15	Definición de interfaces	5h	15/10/16	16/10/16	14
16	Programación de funciones p/sensores	30h	22/10/16	27/11/16	15
17	Programación de funciones p/modem	30h	22/10/16	27/11/16	15
18	Programación de funciones p/ethernet	30h	22/10/16	27/11/16	15
19	Programación de funciones de prueba	10h	03/12/16	11/12/16	16; 17; 18
20	Documentación del firmware	20h	17/12/16	08/01/17	19
21	Definición de la Arquitectura	5h	24/09/16	25/09/16	4
22	Definición de la funcionalidad	5h	14/01/17	15/01/17	21; 20
23	Diseño de la web	25h	21/01/17	19/02/17	22
24	Programación de la interfaz	35h	25/02/17	09/04/17	23
25	Pruebas de funcionamiento	5d	15/04/17	29/04/17	24
26	Integración con el firmware	25h	30/04/17	03/06/17	19; 25
27	Manual de uso	20h	04/06/17	01/07/17	26
28	Planificación de ensayos	10h	02/07/17	15/07/17	27
29	Montaje de set up experimental	10h	16/07/17	29/07/17	28
30	Verificación & Validacion	10h	30/07/17	12/08/17	29
31	Informe de Avance	20h	07/09/17	30/09/17	
32	Memoria Técnica	60h	13/08/17	04/11/17	30
33	Presentacion Final	20h	05/11/17	02/12/17	32
34	FIN	0	02/12/17	02/12/17	33

FIGURA 2.1: Listado de tareas, y plafinacion de tiempos de la implementación.

Capítulo 3

Diseño e Implementación

En este capítulo veremos cómo se desarrolló el hardware básico para poder simular el entorno de funcionamiento y cómo fue implementado el firmware.

3.1. Hardware

Se diseño de un circuito electrónico que permite simular el entorno de funcionamiento de una bodega. Para ello se utilizó lo aprendido a lo largo del curso de *diseño de PCBs en KICAD*. Dado que por falta de tiempo no se elaboró el PCB diseñado, pero se realizó una versión simplificada del mismo que se verá más adelante.

Surgieron varios imprevistos, principalmente con el modulo SIM800L, el cual posee grandes inconvenientes a la hora de detectar la SIM. Luego de una serie de búsquedas en internet, donde muchos habían tenido un problema similar dado que:

- La fuente de alimentación fue subdimensionada.
- Módulo es muy sensible y se debe limpiar bien los contactos de la SIM.
- O problemas con las soldaduras del módulo.

Finalmente, como el módulo ya había sido comprado y se encontró la falla a tiempo, se utilizó este en el prototipo. Pero este no se recomienda por los problemas mencionados anteriormente. A futuras implementación este módulo será reemplazado.

3.1.1. Diagrama en bloque

En el siguiente esquema, figura 3.1 podemos apreciar el entorno de trabajo del dispositivo. Donde el mismo interactuar con una bomba de agua que permite llevar desde el deposito del liquido refrigerante a los tanques que están en proceso de fermentación. Recibiendo como información la temperatura de los tanques, este debe controlar la bomba y la electroválvula correspondiente. De esta forma permitirá tener controlada la temperatura acorde a los parámetros definidos por el usuario.

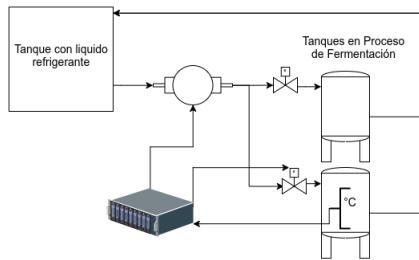


FIGURA 3.1: Esquema de interacción del dispositivo.

3.1.2. Módulos requeridos

Para llevar a cabo el proyecto, se dividió el sistema en los siguientes módulos:

Módem: sera el encargado de transmitir las alertas mediante SMS utilizando el modulo SIM800L.

Actuadores: salidas que nos permiten controlar la bomba y electroválvulas. Debe dido a que se utilizo la CIAA-NXP ya esta resuelto.

Sensores: entradas que permitan simular el comportamiento de los sensores, para esto se utilizaron potenciómetros.

CPU: unidad encargada de controlar el funcionamiento del sistema. Incorporada en la placa CIAA-NXP.

Con lo cual podemos ver que el CPU y los actuadores ya están desarrollados al haber elegido la CIA. Nos quedara ver los sensores y el módem en la próxima sección.

3.1.3. Características del hardware

El sistema cuenta con:

- 1 Salida con regulación de corriente, para simular sensores de 4mA a 20mA.
- 1 Salida con tensión variable, utilizando el regulador LM317.
- 1 Salida del sensor de temperatura LM35.
- 2 Salidas con tensión variable, utilizando potenciómetros.
- 4 Entradas digitales conectadas a leds.
- 4 Salidas conectadas a pulsadores.
- 1 Zócalo para la conexión del modulo SIM800L.
- 1 salida de 5V a 3A y otra de 3.3V a 1A.

El módulo SIM800l que se muestra en la siguiente figura 3.2, es un módem GPRS que cuenta con las siguientes especificaciones:

- Alimentación: 3.4V a 4.4V (4.0V recomendado)
- CuatriBanda 850/900/1800/1900MHz
- GPRS Multi Slot class 8/10



FIGURA 3.2: Módem SIM800L.

- Control mediante comandos AT (GSM 07.07 ,07.05 y comandos AT SIM-COM).

En la figura 3.3 se puede apreciar el esquemático implementado para la conexión del módem GSM. Éste requiere adaptar los niveles de tensión para interactuar con la CIA-NXP, para el cual utilizamos un max3232.

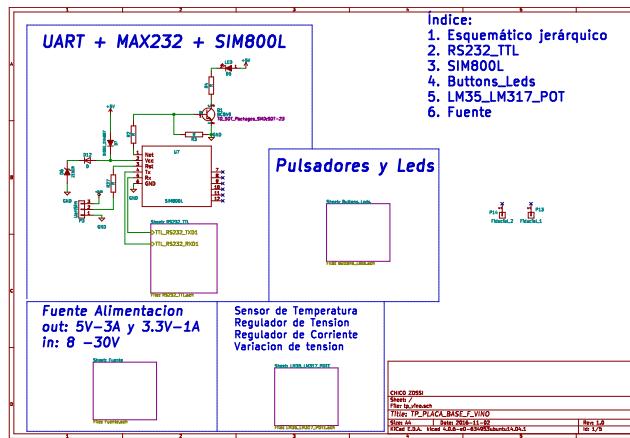


FIGURA 3.3: Módulo SIM800l e índice de esquemáticos.

Para la alimentación del circuito se utilizó el regulador LM2576. El mismo se tomó de la nota de aplicación, es uno de los diagrama típico.[2]. Para regular a 3.3V se utilizó el integrado LM11733 circuito extraído de la nota de aplicación. [1].

Para lograr la simulación de distintos tipos de sensores, se utilizaron circuitos como el que se muestra en la figura 3.5 que permiten generar corrientes 4mA a 20mA. También reguladores de tensión, y hasta un sensor de temperatura. Para estos circuitos se utilizaron las notas de aplicación de Texas Instruments[3] - capítulo 8 páginas 12 y 13 y sensor temperatura: [4] - 8.2.1 Basic Centigrade Temperature Sensor.

Para concluir, se utilizaron unos circuitos básicos que permiten interactuar en forma simple con el sistema y realizar pruebas de funcionamiento. Los mismos fueron extraídos del esquemático utilizado en el proyecto de la EDU-CIA¹.

¹http://www.proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:edu-ciaa:edu-ciaa-nxp:edu-ciaa-nxp_color.pdf

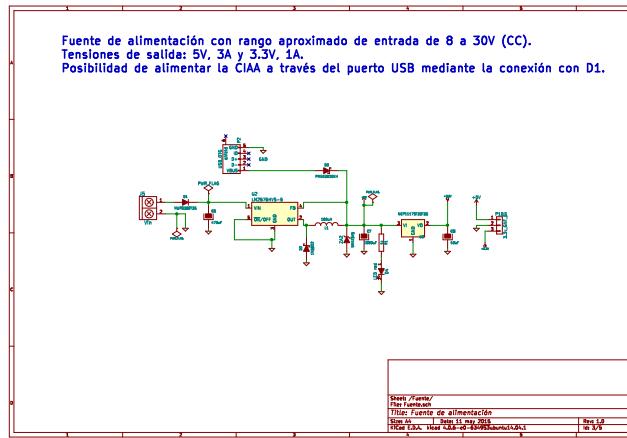


FIGURA 3.4: Fuente de alimentación DC/DC.

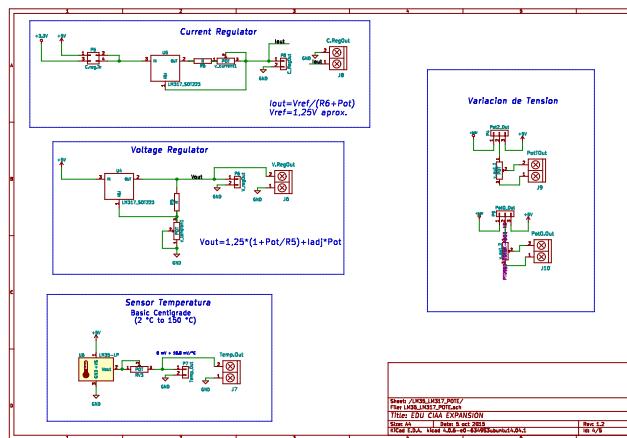


FIGURA 3.5: Regulador de corriente, de tensión y del sensor de temperatura.

Con los esquemáticos terminados, se pasó a la elaboración del PCB. Se realizaron los ruteo necesarios, los cuales podemos verlos en las siguientes figuras 3.7 - 3.8.

Realizando una vista previa de como quedaría el esquema y aprovechando el potencial de la herramienta de KICAD, se realizó una vista en 3D de como quedaría. La que se puede apreciar en la figura 3.9.

No obstante, por cuestiones de tiempo no se llegó a implementar. Simplemente quedó en una primera etapa para una posterior implementación. Se terminó desarrollando una placa experimental que permitió realizar las primeras pruebas. Podemos verla en la siguiente figura 3.10.

3.2. Software

3.2.1. Aplicación

Se dividieron las tareas por recursos:

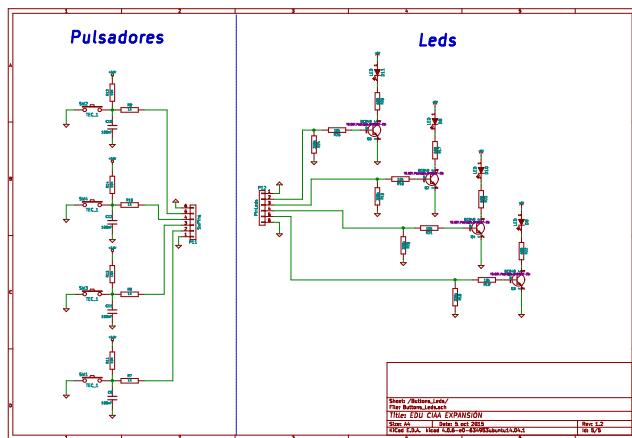


FIGURA 3.6: Entradas a leds y salidas de los pulsadores.

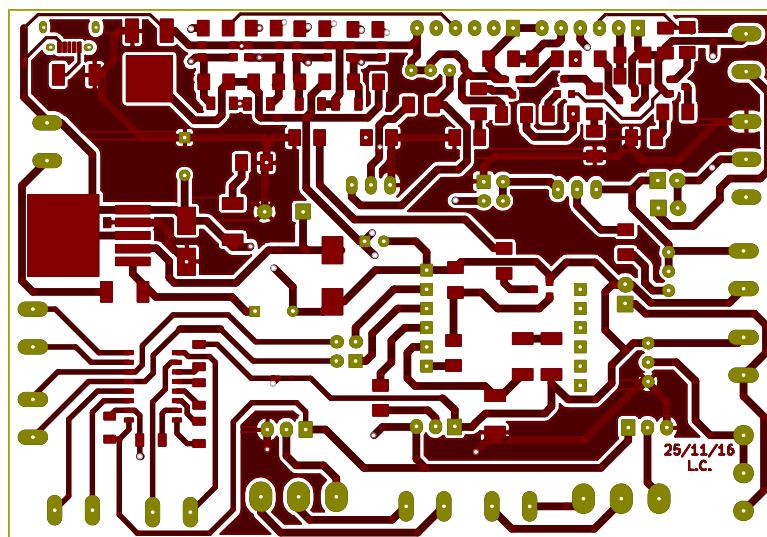


FIGURA 3.7: Capa superior de la placa simulador de sensores.

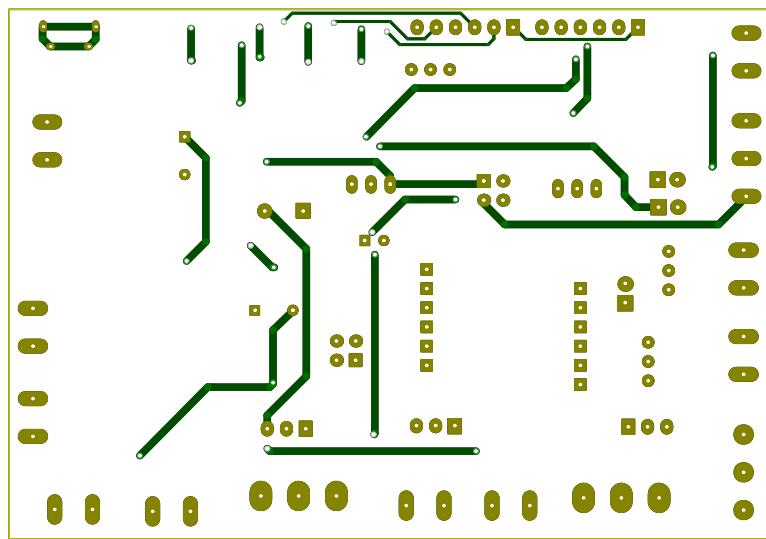


FIGURA 3.8: Capa inferior de la placa simulador de sensores.

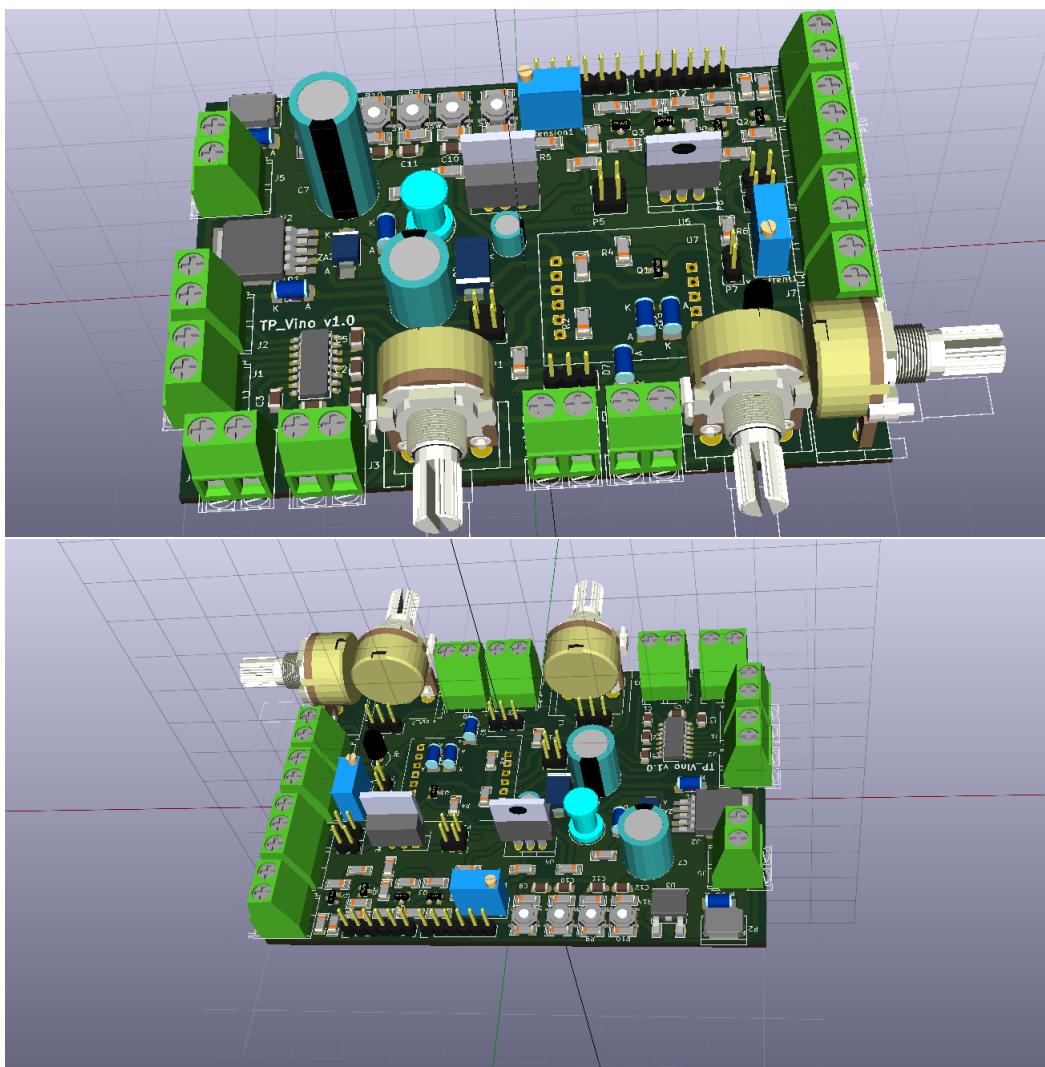


FIGURA 3.9: Placa simulador de sensores 3D.



FIGURA 3.10: Placa de simulación de temperatura y estado de la batería con el modulo SIM800L integrado.

- Módem
- Control
- Sensores

Módem

En esta tarea se inicializa las configuraciones iniciales para el uso del módem. Es la encargada de interactuar con el módulo GPRS, realizando consultas del estado de señal y de enviando los mensajes de alarmas en caso de ser necesarios.

Podemos ver como fue implementado en el diagrama de flujo en la siguiente figura 3.11.

Control

Con esta tarea se analiza el resultado los sensores de temperatura, y de la batería. Se verifica que estén dentro de los parámetros configurados por el usuario y se actúa en consecuencia.

También es la encargada de setear los actuadores según sean especificados en la plataforma web.

La implementación se puede apreciar en la siguiente figura 3.12.

Sensor

Esta consulta el estado de los sensores y calcula el promedio de las ultimas N muestras. Este es el valor que utiliza la tarea de control para definir si es necesario encender los actuadores correspondientes o apagarlos.

Como vemos en el siguiente diagrama de flujo, figura 3.13.

3.2.2. Web

Mediante la plataforma web diseñada se busca que el usuario pueda ver los datos de los sensores, tener control sobre los actuadores y realizar las configuraciones necesarias.

Para poder realizarlo, fue necesario profundizar los siguientes temas:

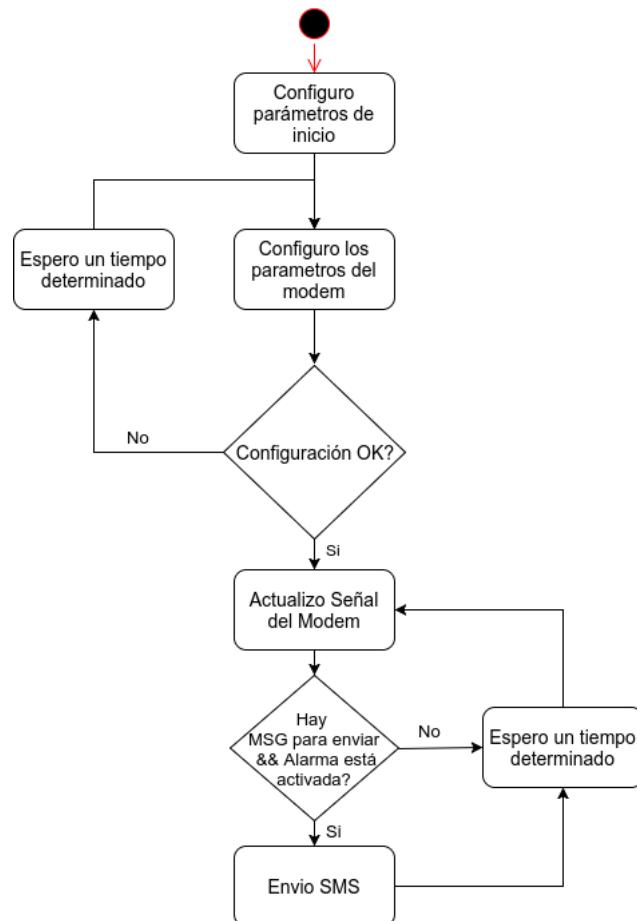


FIGURA 3.11: Diagrama de flujo de la tarea que controla el módem GPRS.

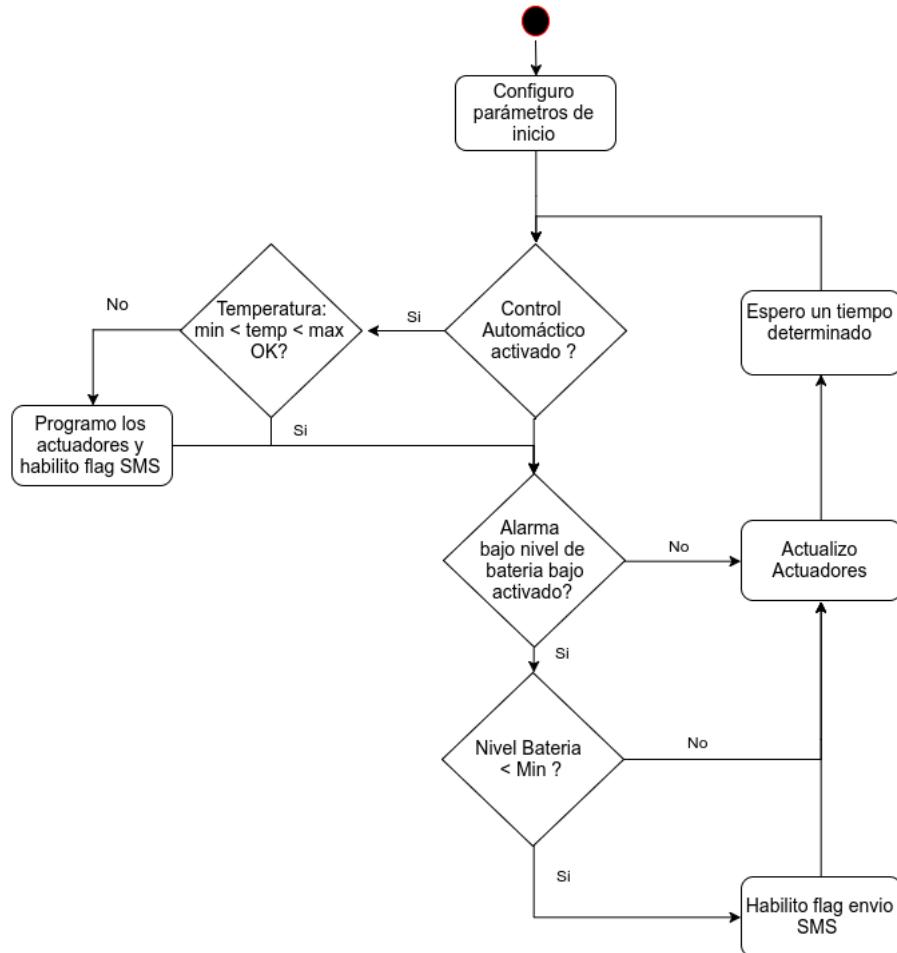


FIGURA 3.12: Diagrama de flujo de la tarea de control.

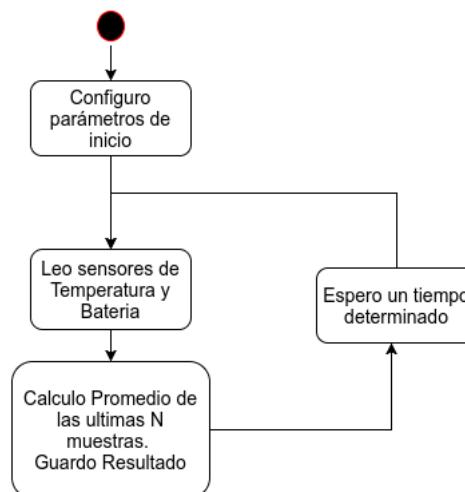


FIGURA 3.13: Diagrama de flujo del sensor.

HTML: En dicho código se desarrolla la página web, este permite darle la estructura básica y se realiza sobre texto plano, es decir no se requiere ninguna aplicación adicional simplemente un editor de texto. Luego es el navegador web quien tiene la tarea de interpretarlo.

CSS: Mediante este código se realiza la presentación de la página, permitiendo así modificar la visualización de la misma y darle un formato más profesional y elegante.

SVG: Mediante este código se implementaron los gráficos correspondientes para mostrar los resultados de los sensores y el estado de los actuadores.

AJAX: es un recurso muy utilizado cuando se desarrollan web en sistemas embedidos. Permite actualizar el estado de la web sin tener que consultar todo el contenido, simplemente se descarga una primera vez y luego se realizan consultas de los valores que quieren actualizar.

SSI: Si bien la implementación que nos ofrece FreeRTOS es reducida. Es más que suficiente para poder agregar contenido a nuestra página en forma dinámica.

CGI: Por medio de este el cliente que navegar por nuestra plataforma puede enviar y solicitar datos, al servidor.

Diseño web

Al iniciar el diseño de la web, no tenía experiencia previa. Es por ello que se inició buscando información en internet, donde se encontraron varias páginas con cursos de HTML, CSS, y AJAX que son de forma online y gratuita. Las que mas se utilizó fueron:

- <https://www.w3schools.com>
- <https://es.khanacademy.org>
- <https://www.codecademy.com>

con lo cual se logró una primera versión de la web, como se puede ver en la figura 3.14.

Cuando ya estaba funcionando en forma correcta, se pasó a mejorar las presentación de la misma. Para ello se usó un template que se ajustara a las necesidades, es decir mediante un archivo que contiene un estilo ya predefinido modifica la apariencia de la web. Pudiendo ver dicho cambio en la siguiente figura 4.4.

Este template fue encontrado en w3school². Siendo que cumplía los siguientes requerimientos:

- Sea amigable.
- Compatibles con distintos navegadores, especialmente los más utilizados como ser: Google-Chrome y Firefox.
- Sea cómodo de navegar en distintos dispositivos, como tablets, celulares y PCs.

²https://www.w3schools.com/w3css/w3css_templates.asp

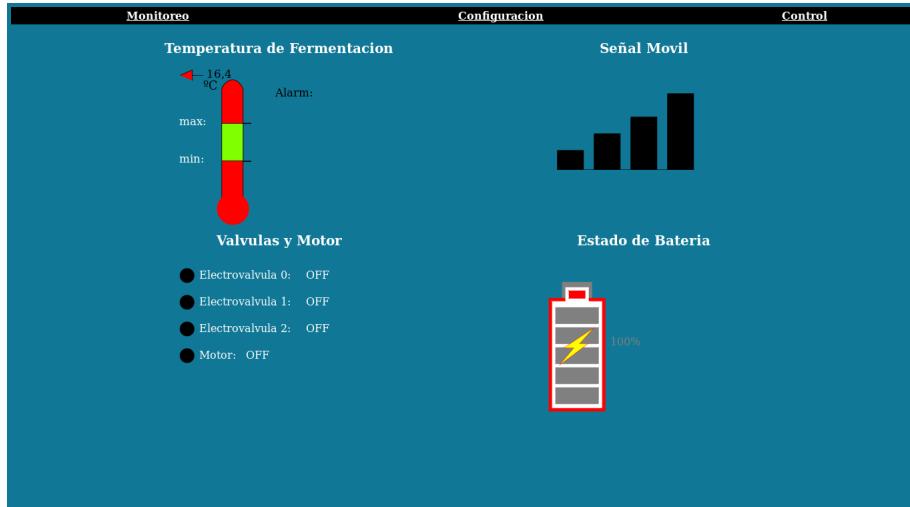


FIGURA 3.14: Primera versión de la web de monitoreo.

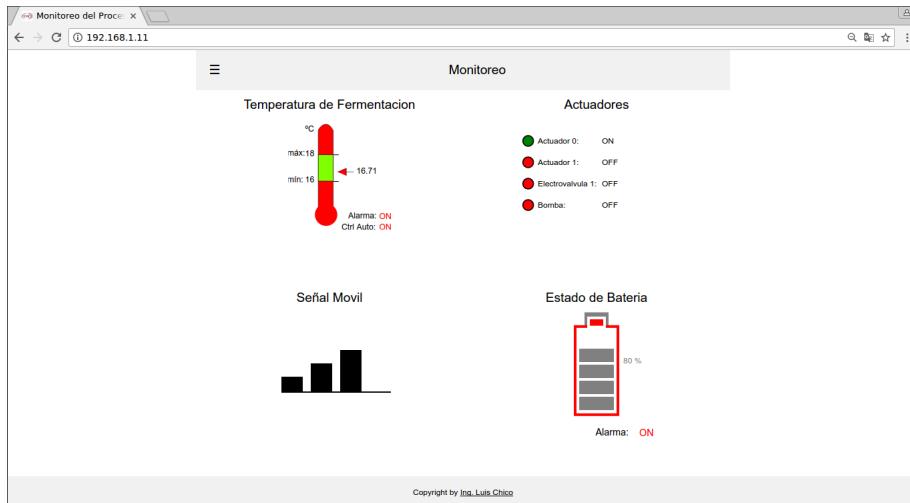


FIGURA 3.15: Versión actualizada haciendo uso del template.

Configuración del servidor

Al iniciar el servidor debemos tener en cuenta que este sera el encargado de gestionar las peticiones del usuario. Es decir una vez que el usuario ingresa la dirección del servidor, este le transmite la información de la web. Para realizar la comunicación entre cliente y servidor, se utilizo la API de lwIP que utiliza una serie de callbacks para controlar los eventos debido a la comunicación de la red. Es por ello que para optimizar los recursos se utilizo AJAX, permitiendo que el cliente sea el encargado de realizar las consultas al server en forma periódica. La información se solicita mediante directivas del tipo SSI. Es decir, solo se transmite un archivo ajax.shtml con los parámetros que se van a actualizar. Luego el navegador, mediante una serie de funciones en JavaScript es el encargado de actualizar los cambios realizados. Podemos ver el procedimiento mencionado en la siguiente figura³ 3.16.

³<http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-Trabajo-Final-CESE-Patricio-Bos-2016.pdf>, figura 3.9.

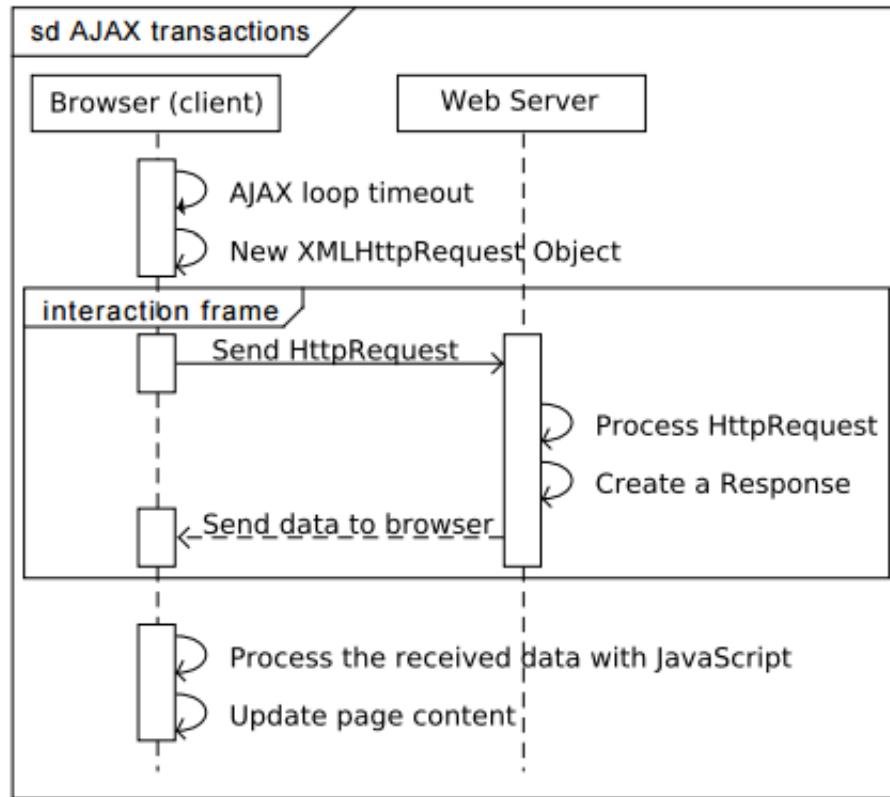


FIGURA 3.16: Diagrama de secuencia AJAX.

La configuración establecida por default es la que se muestra en la siguiente Tabla 3.1.

Parámetro	Valor
IP	192.168.1.11
Máscara de red	255.255.255.0
Puerta de enlace	192.168.0.1
Puerto	80
DHCP	Deshabilitado

TABLA 3.1: Configuración por default del servidor

Capítulo 4

Ensayos y Resultados

En este capítulo se expone cuales fueron las pruebas realizadas para determinar que el sistema funciona en forma correcta.

4.1. Banco de pruebas

Para poder simular el comportamiento de un sensor de temperatura y la descarga de una batería. Se implemento una placa básica la cual ya fue vista en el capítulo 3.2, que para facilidad del lector volvemos a mostrarla en la siguiente figura 4.1. La cual consta de dos potenciómetros que permiten simular la variación de estos dispositivos. Sobre la misma se conecto un conversor RS232 a TTL, para lograr la integración con el módem.



FIGURA 4.1: Placa de simulación de temperatura y estado de la batería con el modulo SIM800l integrado.

Todo esto se integra la Plataforma de la CIAA-NXP, véase figura 4.2, para verificar el correcto funcionamiento del firmware desarrollado.

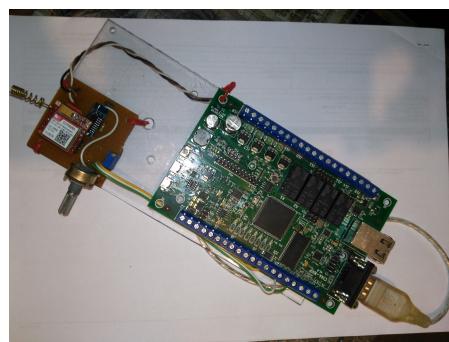


FIGURA 4.2: Placa básica integrada con la CIAA-NXP.

4.2. Configuraciones para la PC

Para la realización de los ensayos, comenzamos configurando la PC en forma adecuada. Para ello en la siguiente figura 4.3 vamos a ver como configurar en Linux la interfaz de red. Es importante destacar que dichos comandos son ejecutados con el permiso se súper usuario, *root*.

```
root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
              UP LOOPBACK RUNNING MTU:65536 Metric:1
              RX packets:26202 errors:0 dropped:0 overruns:0 frame:0
              TX packets:26202 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1
              RX bytes:1987773 (1.9 MB)  TX bytes:1987773 (1.9 MB)

root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# ifconfig eth0 192.168.1.2/24
root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
192.168.1.0     *           255.255.255.0   U     0      0    0 eth0
root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# ifconfig
eth0      Link encap:Ethernet HWaddr a4:ba:db:a:b5:30
        inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::a4ba:dbff:fedab5:30/64 Scope:Link
              UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
              RX packets:3969 errors:0 dropped:0 overruns:0 frame:0
              TX packets:6035 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:1460102 (1.4 MB)  TX bytes:765043 (765.0 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
              UP LOOPBACK RUNNING MTU:65536 Metric:1
              RX packets:26262 errors:0 dropped:0 overruns:0 frame:0
              TX packets:26262 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1
              RX bytes:1992384 (1.9 MB)  TX bytes:1992384 (1.9 MB)

root@joseph:/home/lchico/Documents/Tesis_Especializacion/Memoria/Informe/Chapters# 
```

FIGURA 4.3: Configuración de una red estática para la interfaz Ethernet.

Una vez que tenemos la configuración de red adecuada vamos a proceder a dirigirnos a un navegador web. Y como vemos en la imagen 4.4 que fue vista en el capítulo anterior y que para facilitar al lector se ingreso nuevamente, donde el numero de IP correspondiente al dispositivo de control es: 192.168.1.11

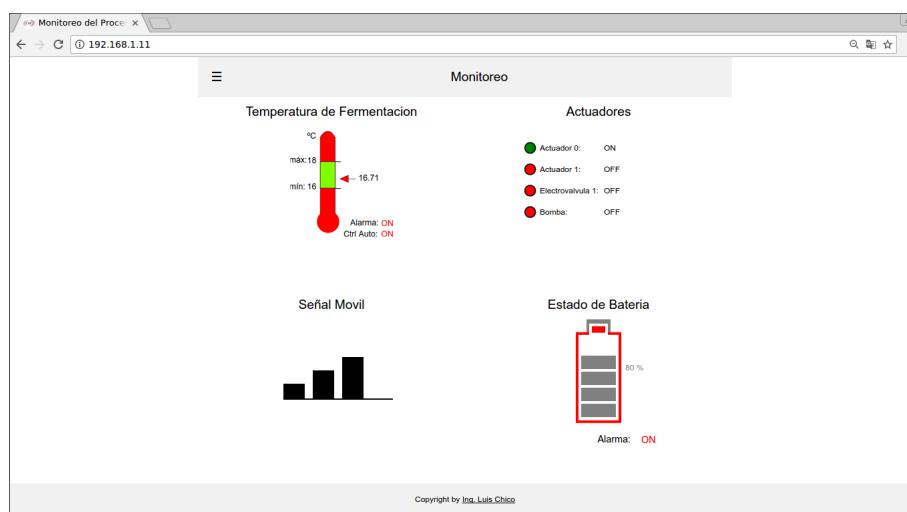


FIGURA 4.4: Pantalla principal de la web de monitoreo.

En esta pantalla vamos a tener toda la información correspondiente al sistema. Arriba a la izquierda podemos ver el estado de la temperatura, con sus rangos

máximos y mínimos seteados. Y debajo de dicho termómetro podemos ver si esta activada: la alarma y el control automático.

Luego arriba y a la derecha podemos ver los estados de los actuadores, la electroválvula y la bomba. Estos le permiten al cliente controlar en forma remota dos actuadores y el sistema correspondiente a la bomba de refrigeración con la electroválvula asociada. El sensor de temperatura permitirá informarle al sistema de control automático cuando debe actuar acorde a los rangos establecidos, esto solo en el caso de estar activado dicho control automático.

Abajo a la izquierda tenemos el nivel de señal correspondiente al módem GSM, mediante el cual podremos saber si la cobertura de señal de la red de telefonía móvil esta en condiciones de operar y permitirnos el servicio de alertas mediante mensajes SMS.

Finalmente abajo a la derecha tenemos el estado de la batería, el cual va a permitir al sistema continuar en funcionamiento en caso de un corte de energía. El principal uso de este será mantener la posibilidad de enviar un mensaje SMS notificando el corte de energía.

4.3. Verificación del sistema web embebido

De lo explicado en la sección anterior, se procederá a navegar por el menú, véase figura 4.5, las diferentes configuraciones para constatar de que todo el sistema esta funcionando en forma correcta. Al presionar donde dice menú, obtendremos las opciones de configuración. Donde:



FIGURA 4.5: Opciones del al presionar sobre el menú desplegable.

1. **Monitoreo:** Nos dirige a la pantalla principal, donde podemos ver el estado del sistema. figura 4.4.
2. **Actuadores:** Podemos setear en forma manual el estado de los mismos. figura 4.6.
3. **Batería:** Aquí se podrá setear la alarma debido al nivel de descarga de batería. Esta enviara un SMS indicando que se alcanzó dicho estado. figura 4.7.

- 4. Módem GPRS:** En este menú podremos configurar 2 personas a quienes serán enviadas las alertas debido al accionar de alguna alarma, figura 4.8.
- 5. Temperatura:** Aquí se configura el rango de temperatura, la alarma correspondiente y si deseamos activar el control automático de la misma. De estar activado, actuará en forma automática sobre la bomba y la electroválvula para mantener la temperatura dentro de dicho rango. figura 4.9.
- 6. Network:** Aquí se configura los parámetros de la red, IP, mascara de red y puerta de enlace. figura 4.10.
- 7. Cerrar Menú:** Esta opción cierra el menú, permitiendo quitar las opciones de menú.

Estado de los contactores	Actuadores
Actuador0	ON <input checked="" type="radio"/> OFF <input type="radio"/>
Actuador1	ON <input checked="" type="radio"/> OFF <input type="radio"/>
Electrovalvula1	ON <input checked="" type="radio"/> OFF <input type="radio"/>
Bomba	ON <input checked="" type="radio"/> OFF <input type="radio"/>

Copyright by Ing. Luis Chico

FIGURA 4.6: Activar/Desactivar actuadores.

Monitoreo del Proceso de Fermentación del Vino - Google Chrome
192.168.1.11

Bateria

Estado de Batería

Valor

Mínimo Máximo

Alarma OFF

Submit

De estar activada la alarma recibirá un SMS

0 %

Alarma: ON

Copyright by Ing. Luis Chico

FIGURA 4.7: Configurar nivel de descarga para accionar la alerta por batería baja.

Monitoreo del Proceso de Fermentación del Vino - Google Chrome
192.168.0.11

Configurar Modem GPRS

Señal Movil

Configurar telefonos para reportar alarmas SMS

	Nombre	Nro de Celular	Estado
Opcion_1:	<input type="text"/> Test_f	<input type="text"/> 64958758	<input checked="" type="radio"/> Activo <input type="radio"/> Inactivo
Opcion_2:	<input type="text"/> Person_2	<input type="text"/> Nro Cel2	<input checked="" type="radio"/> Activo <input type="radio"/> Inactivo

Submit

Copyright by Ing. Luis Chico

FIGURA 4.8: Configuración de los celulares de quienes van a recibir las alarmas por SMS.



FIGURA 4.9: Configuración del rango de temperatura y activación del estado de control automático y alarma.

FIGURA 4.10: Formulario de configuración de IP, mascara y puerta de enlace.

4.4. Pruebas funcionales del hardware

Una vez recorrido todas las opciones que permite la interfaz web y verificar que el comportamiento haya sido el esperado. Se procede interactuar con el sistema, por medio de alteraciones generadas a través de los potenciómetros que representaran la temperatura y el estado de la batería.

Estando activado el control automático y al simular un incremento de temperatura, se comprobó que al exceder la temperatura máxima inicio en forma automática la bomba que permite la circulación del refrigerante y la activación de la electroválvula que interviene sobre el tanque que estamos monitoreando. En la web se podrá ver como en la figura 4.11. y se recibirá la alerta correspondiente mediante un SMS como se puede observar en la figura 4.12.

Se verificó que éste se mantiene activado hasta llegar a la temperatura mínima, en este instante procede a desconectar la bomba y a cerrar la electroválvula. En la web se verá como en la figura 4.13. y dado su respectiva alerta mediante un SMS como vemos en la figura 4.12

En el caso de la batería se analiza las variaciones en la carga. Si esta disminuye más de un valor que es determinado, se enviará SMS como el que observamos en figura 4.12.

Para verificar los actuadores nos dirigimos a la web en la sección de configuración de actuadores y seteamos para cada uno de ellos los diferentes estados, encendiendo y apagando. Luego se probó en bloques de varios actuadores. En la siguiente figura 4.14 se ve un ejemplo de los casos probados.

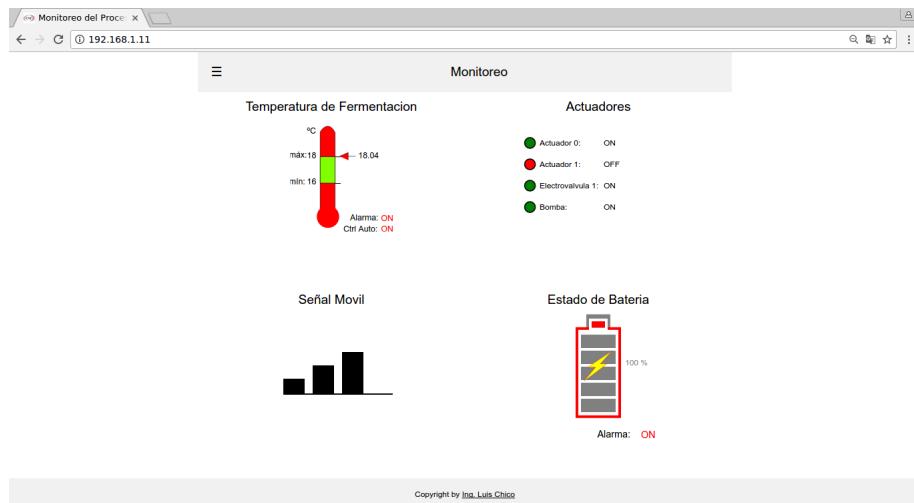


FIGURA 4.11: Activación por control automático debido a temperatura elevada.

De esta forma podemos concluir que el funcionamiento del sistema es el correcto.

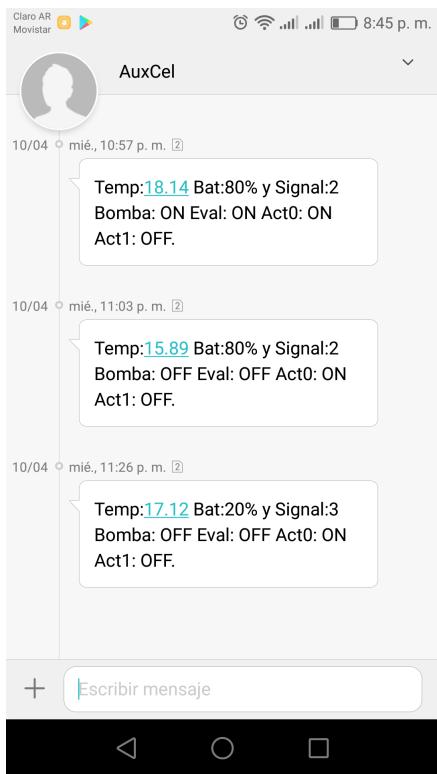


FIGURA 4.12: SMS recibidos debido a temperatura máxima.
El segundo SMS recibido es por temperatura mínima.
Y el último recibido debido a batería baja.

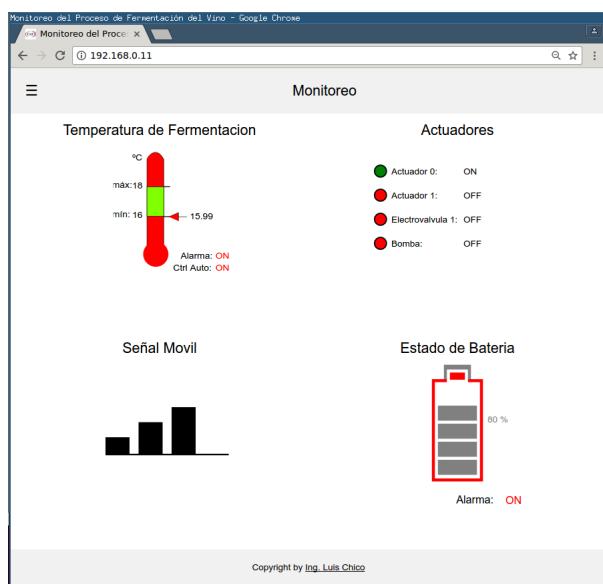


FIGURA 4.13: Temperatura en nivel mínimo, se desactiva la bomba y la electroválvula.



FIGURA 4.14: Activación manual de actuador 1 y 2, electroválvula y bomba inactivo.

Capítulo 5

Conclusiones

5.1. Conclusiones generales

Al finalizar esta primera etapa del prototipo, podemos concluir que se logró:

- Mediante una página web accedemos al estado del sistema.
- Permite navegación mediante pc, celulares, tablets compatibles con navegadores como Chrome y Firefox.
- Permite realizar un control manual y automático del sistema. Mediante el uso de un sensor de temperatura y salidas para actuadores.
- Alertar mediante SMS cuando: la temperatura excede el rango permitido, y/o para un determinado nivel de batería.
- Implementar mediante el stack TCP/IP de lwIP mediante el RTOS freeRTOS.
- Uso del lenguaje C, HTML, CSS y JavaScript.
- Hacer uso de un sistema de control de versiones, GitHub.

Es decir, ya podemos controlar la temperatura de un tanque y recibir alertas en los principales estados críticos del proceso. Tenemos incorporado una interfaz de monitoreo que a su vez nos permite actuar en forma remota.

5.2. Próximos pasos

En esta primera etapa se dió una primera interacción con todo lo que involucra controlar el proceso de fermentación. Ahora estamos en condiciones de perfeccionar este producto extendiendo sus funciones. Para ello podríamos optimizar los siguientes puntos:

- Ampliar la capacidad de control a 4 tanques.
- Mejorar la interfaz web para dichos cambios, y realizar un entorno más amigable.
- Ampliar las funcionalidades de los SMS, no sólo para reportes de alertas sino también para consultas y posibles controles.
- Agregar retroalimentación de sensores y actuadores por hardware para un control más robusto.

- Adicionar una opción de control por intervalos de tiempos.

Bibliografía

- [1] *LM1117 800-mA Low-Dropout Linear Regulator*. Capítulo 8 Figura 16. Texas Instruments. 2016. URL: <http://www.ti.com/lit/ds/symlink/lm1117.pdf>.
- [2] *LM2576xx Series SIMPLE SWITCHER® 3-A Step-Down Voltage Regulator*. Texas Instruments. 2016. URL: <http://www.ti.com/lit/ds/symlink/lm2576.pdf>.
- [3] *LM317 3-Terminal Adjustable Regulator*. Texas Instruments. 2016. URL: <http://www.ti.com/lit/ds/symlink/lm317.pdf>.
- [4] *LM35 Precision Centigrade Temperature Sensors*. Texas Instruments. 2016. URL: <http://www.ti.com/lit/ds/symlink/lm35.pdf>.