
Diseño e implementación de un sistema de supervisión y monitoreo para la apicultura



Tesis de grado

Sergio Daniel Navarro López

Universidad Nacional de Colombia - Sede Bogotá

Diciembre 2012

Documento maquetado con TEXIS v.1.0+.

Diseño e implementación de un sistema de supervisión y monitoreo para la apicultura

*Trabajo de grado presentado como requisito parcial para optar
por el título de:
Ingeniero Electrónico*

Director: Giovanni Aldemar Baquero Rozo

Universidad Nacional de Colombia - Sede Bogotá

Diciembre 2012

Copyright © Sergio Daniel Navarro Lopez

Agradecimientos

El autor expresa sus agradecimientos a los docentes y compañeros que brindaron sus valiosos aportes y sugerencias durante la realización del proyecto. De igual manera, a la Universidad Nacional de Colombia en su conjunto por darme la posibilidad de concluir con éxito este trabajo, además de proveer los conocimientos y herramientas necesarias, cumpliendo con su compromiso de formar profesionales de alta calidad. Por último, a mi familia que me ha apoyado incondicionalmente durante la realización de este trabajo y a Saida Benhoura Miquel, quien me ha apoyado académica y personalmente en la culminación de este trabajo.

Resumen

En este artículo se describe el diseño y la implementación de un Sistema de Supervisión y Monitoreo para la Apicultura (SSMA), que tiene como función recibir la información capturada por los sensores en el Sistema de Medición y Trasmisión de Información Inalámbrica (SMTIA), organizarla, activar el envío de mensajes y mostrarla al usuario final. Para ello, por una parte se han utilizado servicios de red, bases de datos embebidas SQLite, aplicaciones Web basadas en Python y una interfaz hombre máquina que pueda ser accesible desde otros nodos. Por otra parte, se utiliza un tipo de conexión inalámbrica IEEE 802.11 y acceso a red GSM. Finalmente, se pretende instalar el SSMA en el centro de acopio del apíario, ubicado en la finca de la Universidad Nacional de Colombia - Sede Bogotá, Marengo.

Índice

Agradecimientos	v
Resumen	vii
1. Introducción	1
1.1. Objetivos	2
1.1.1. Objetivo General	2
1.2. Objetivos específicos	2
1.3. Alcances y limitaciones	3
2. Estado del arte y definición de problema	5
2.1. Sistemas embebidos	6
2.2. Comunicaciones inalámbricas	6
2.3. Sistemas de supervisión y monitoreo	7
2.3.1. Supervisión	7
2.3.2. Monitoreo	8
2.3.3. Características de un sistema de supervisión y monitoreo.	8
2.4. Definición del problema	8
3. Metodología de diseño e implementación	11
3.1. Definición de especificaciones	11
3.2. Selección de tecnologías	13
3.2.1. Selección de Hardware.	14
3.3. Implementación	17
3.3.1. Conexión inalámbrica.	17
3.3.2. Conexión a la red GSM.	20

3.3.3. Compilación de un kernel libre para la aplicación	20
3.3.4. Sistema de archivos	21
3.3.5. Base de datos	25
3.3.6. Aplicación Web	25
3.3.7. Mensajes SMS	26
3.3.8. Mensajes de correo electrónico	27
3.3.9. Utilidades terminal	27
3.3.10. HMI	27
3.3.11. Base de datos para el manejo de la información.	29
3.3.12. Servicios que permitan la recepción de datos provenientes de la red de sensores.	32
3.3.13. Aplicaciones que permiten enviar mensajes de texto o correo electrónico.	35
3.3.14. Configuración de la red de datos inalámbrica que conecta el apíario con el centro de mando.	39
3.3.15. Interfaz hombre maquina HMI.	39
3.4. Construcción del dispositivo	42
4. Pruebas y resultados	47
4.1. Conexión inalámbrica	49
4.2. Conexión GSM	56
4.3. Inicio del kernel	56
4.4. Base de datos SQLite	57
4.5. Servidor UDP	58
4.6. Cliente UDP	59
4.7. Aplicación Web	59
4.7.1. Webpy	59
4.7.2. HTML5	60
4.8. Interfaz Hombre Maquina	60
4.8.1. Apiario	60
4.8.2. Colmena	61
4.8.3. Config	61
4.8.4. Utilities	61
5. Conclusiones	65
5.1. Conclusiones	65

Bibliografía

67

Lista de acrónimos

69

Índice de figuras

2.1. Diagrama general del SSMA	9
3.1. Sistema de supervisión y monitoreo para la apicultura	13
3.2. Modulo Stamp. Tomado de www.friendlyarm.net	15
3.3. Modulo SDK con entradas y salidas. Tomado de www.friendlyarm.net	16
3.4. Tarjeta de red usb wifi Encore N150 (tomado de http://www.encore-usa.com/mx/product/WU8188CUS2)	18
3.5. Menú de configuración de buildroot.	28
3.6. Arquitectura para la base de datos relacional del SSMA	30
3.7. Estructura de la base de datos relacional que se utilizará para la aplicación	32
3.8. Diagrama de flujo que describe las tareas a ejecutar por el servidor UDP en Chumby	34
3.9. Diagrama de flujo que describe las tareas a ejecutar por el cliente UDP en FriendlyARM	35
3.10. Diagrama de bloques	36
3.11. Topología de red para conectar los SMTIA a SSMA	40
3.12. Estructura de la interfaz hombre máquina	41
3.13. Imagen del montaje de la tarjeta en la caja de paso	43
3.14. Imagen del montaje de la pantalla LCD en la caja de paso	44
3.15. Imagen del montaje de la pantalla LCD en la caja de paso	45
4.1. Topología de red para conectar los SMTIA a SSMA	48
4.2. Topología de red para conectar los SMTIA a SSMA en pruebas de laboratorio	49
4.3. Sistema de Monitoreo y Trasmisión Inalámbrico para la Apicultura.	50

4.4.	SMTIA con estructura que contiene los sensores para la colmena	51
4.5.	Disposicion final del SSMA y SMTIA	52
4.6.	Imagen que muestra el montaje para pruebas en laboratorio .	53
4.7.	Salida del comando <i>sqlite3</i> , consultando la tabla con información almacenada	58
4.8.	Mensajes de la terminal que muestran la información enviada por la aplicación servidor UDP	59
4.9.	Mensajes de la terminal que muestran la información enviada por la aplicación servidor UDP	60
4.10.	Interfaz Hombre Maquina vista desde un navegador Web externo al SSMA	62
4.11.	Interfaz Hombre Maquina vista en la pantalla LCD del SSMA	63

Índice de Tablas

3.1. Selección de Hardware Sistema Embebido	14
3.2. Selección de tecnologías Tarjeta WiFi	17
3.3. Elementos de software requeridos para implementar el SSMA . .	23
3.4. Comparación de sistemas de archivos	23
3.5. Selección de sistema de base de datos	29

Capítulo 1

Introducción

La apicultura es la crianza de abejas, recolección y explotación controlada de los productos generados por esta clase de insectos (6). En la actualidad se logran obtener al menos diez productos directos de colmena: miel, jalea real, polen, pan de abejas, cera, propoleos, embriones de zángano, apitoxina y aire proveniente de la colmena. Igualmente se sigue investigando que otros productos se pueden extraer de la actividad de las abejas.

Con el animo de preservar la explotación controlada y la investigación respecto a nuevos productos, existen investigaciones específicas en los campos de interes que involucran la actividad de las abejas.

Adicionalmente, los sistemas de supervisión y monitoreo se han venido desarrollando ampliamente en diferentes areas, como la actividad industrial, aeronautica, minera, energetica, economica, salud, producción a gran escala y procesos agricolas o agroindustriales, entre otros. En el area enfocada en los procesos agricolas, la mayoria de sistemas de supervisión y monitoreo se enfoca en la observación de variables climatologicas que afectan los cultivos o entornos de crianza de animales.

Con el desarrollo de la red de redes Internet y sus tecnologías relacionadas, además del avance en las escalas de integración de semiconductores y su aplicación en sistemas embebidos, cada vez se encuentran con más frecuencia dispositivos que pueden tener una conexión robusta a redes de datos. Es posible encontrar en el mercado actual, dispositivos de red, controladores industriales, medidores y otro tipo de instrumentos, que permiten realizar sus tareas de forma remota, haciendo disponible una intefaz completa para su visualización y manejo. Ejemplo de estos equipos los podemos encontrar

fácilmente en hogares o instituciones donde se cuenta con una conexión a Internet, la cual se gestiona mediante un dispositivo de esta naturaleza.

En determinado momento se consideró que el computador personal PC era el equipo que más se utilizaba como nodo de la red Internet. Pero gracias al desarrollo de herramientas de software libre y el aporte de la comunidad de personas interesadas en el desarrollo de nuevos sistemas, el concepto de nodo de red ya no se limita a un equipamiento de hardware muy grande o sobredimensionado para una aplicación específica.

Un sistema de supervisión y monitoreo permite hacer una conexión en red y usando el modelo cliente servidor, las posibilidades de adquisición, visualización y análisis de datos de un fenómeno en particular aumentan, ya que permite llegar a lugares o entornos donde la instalación de un gran equipo de computo es muy compleja. En el caso del Sistema de Supervisión y Monitoreo para la Apicultura (SSMA), se requiere implementar el dispositivo en lugares donde el acceso a la red eléctrica presenta dificultades, por lo que el diseño de la aplicación sobre un sistema embebido se convierte en la solución más favorable, teniendo en cuenta costos, tamaño y tiempo de desarrollo.

1.1. Objetivos

1.1.1. Objetivo General

Diseñar e implementar un sistema de supervisión y monitoreo inalámbrico remoto para un apiario, basado en un sistema embebido de arquitectura ARM.

1.2. Objetivos específicos

- Desarrollar una metodología para buscar utilizar tecnologías de bajo costo y gran alcance para la trasmisión de datos inalámbrica entre el apiario y el sistema de supervisión y monitoreo.
- Implementar un sistema de base de datos en un sistema embebido y crear la estructura de datos para almacenar datos generados por la actividad de un apiario.
- Diseñar e implementar una aplicación Web que funcione como interfaz

hombre maquina (HMI) que permita la visualización de las variables en observación y una base de datos embebida que permita el almacenamiento, organización y consulta de estos mismos valores.

1.3. Alcances y limitaciones

- La red de sensores para la captura de datos del apíario ya se encuentra instalada en la finca de la Universidad Nacional de Colombia - Sede Bogotá, Marengo, como resultado del trabajo de grado *Diseño y construcción de un sistema de medición y transmisión de información inalámbrica para la apicultura*, realizada por Jeisson Alejandro Rodriguez.
- El sistema de captura y procesamiento de información entrega una estructura de datos que procesa el sistema embebido basado en ARM para almacenar y enviar la información de forma inalámbrica, a través de canales de comunicación de red.
- En la finca de la Universidad Nacional de Colombia - Sede Bogotá, Marengo, se encuentra disponible una red local de datos con acceso a Internet y cobertura de la red celular GSM/GPRS.

Capítulo 2

Estado del arte y definición de problema

La incorporación de los avances en el desarrollo de productos tecnologicos para la industria, en el campo de la agricultura, han traído beneficios en las actividades de esta ciencia. Sumandole el hecho, que la era de la información ha llegado a muchos campos de investigacion, se ha facilitado la intergracion de nuevos dispositivos con los procesos en cuestión.(1).

Mediante la introducción tecnologica se busca aprovechar y reorganizar la agricultura hacia un modelo mas sostenible, mas eficiente y de bajo costo. En busqueda de este beneficio, se pueden utilizar tecnologia y plataformas implementadas en otras areas y aplicaciones, si tenemos en cuenta los avances de los sistemas de posicionamiento global, comunicaciones inalámbricas y procesamiento de datos entre otros.

Para obtener resultados exitosos en la implementacion de una tecnologia en una area de reciente interes, se hace necesario tener en cuenta un diseño orientado al ciclo de vida del producto, para que no se vean afectados el entorno del area de aplicación, buscando como se dijo anteriormente, la mayor eficiencia, que incluye tambien reducir al maximo las consecuencias o efectos secundarios que pueda tener un desarrollo tecnologico en la introduccion al nuevo campo de interes.

2.1. Sistemas embebidos

Un sistema embebido es un sistema de computo diseñado para una aplicación específica, habitualmente acondicionados para funcionamiento en tiempo real. Generalmente se componen de un procesador o microcontrolador que ya implementa muchas funciones de alto nivel, como interfaces con el exterior o periféricos de uso común, un sistema de memoria RAM, para ejecutar distintas tareas independientemente, memoria de almacenamiento permanente y dependiendo para la aplicación que han sido diseñados, incluyen otro tipo de circuitos, inclusive un procesador auxiliar que también cumple algunas tareas específicas.

Gracias al desarrollo de la electrónica digital y el aumento de las escalas de integración de semiconductores, en el mercado también se ven grandes movimientos respecto a lo que ofrecen los fabricantes de electrónica y ha facilitado la integración de sistemas, logrando así crear toda una corriente de dispositivos que funcionan con bajos niveles de energía, ayudando en la realización de tareas cotidianas o repetitivas de procesos donde se aplican estas tecnologías.

En sistemas de supervisión y monitoreo se observa una amplia utilización de este tipo de sistemas, ya que permite tener en un solo dispositivo distintos tipos de circuitos, como módulo para conversión analógico digital, comunicaciones inalámbricas e interfaces multimedia.

Desarrollar una solución en este tipo de dispositivos permite reducir costos de producción, diseño, implementación, depuración y actualización.

2.2. Comunicaciones inalámbricas

Los sistemas de monitoreo en red que usan protocolos IP, habitualmente son los preferidos en la introducción de una nueva tecnología, ya que su difusión a nivel mundial, facilita las tareas de diseño e implementación. También se utilizan protocolos que usan gran cantidad de dispositivos, facilitando la recolección, análisis y transmisión de datos procedentes del fenómeno en observación.

La red de área local inalámbrica (Wireless Local Area Network) WLAN, es un sistema de comunicación de datos inalámbrico, utilizado habitualmente como alternativa a la red de área local cableada, (Local Area Network LAN).

Esta basada en radiofrecuencia, eliminando la necesidad de cables. De esta forma se puede prestar los mismos servicios de red que se tendrían con una red LAN, como servidores de correo, servidores web y compartición de recursos entre otros (3).

Un red inalámbrica de área local habitualmente se configura mediante los siguientes elementos:

- Puntos de acceso: es el punto mas importante de una red inalámbrica, tiene cobertura omnidireccional, permite establecer conexiones entre los diferentes nodos de la red y permite hacer la interfaz entre la plataforma de radiofrecuencia y la red de datos cableada.
- Tarjeta de red inalámbrica: se instala en los nodos que se quieran conectar a la red. Se encuentran de modo PCI, PCI-E y USB.
- Enrutadores: se encarga de hacer la conexión entre redes.

En algunos casos se pueden encontrar dispositivos que cumplen la función de enrutador y punto de acceso, facilitando la implementación de una red de área local inalámbrica.

2.3. Sistemas de supervisión y monitoreo

2.3.1. Supervisión

Se define como supervisión el conjunto de acciones realizadas con el propósito de asegurar el correcto funcionamiento de un proceso, inclusive cuando se presentan situaciones anormales.

Las tareas de supervisión en procesos industriales inicialmente se realizaban mediante sinópticos del proceso, creando la visualización mediante lámparas, displays, alarmas, mientras que por dispositivos como interruptores, pulsadores y conmutadores se realizaba el accionamiento de algún proceso o máquina específica.

2.3.2. Monitoreo

2.3.3. Características de un sistema de supervisión y monitoreo.

Existen diferentes tipos de arquitecturas para supervisión y monitoreo (10). En el apiario se requiere la medición y transmisión de las variables temperatura y humedad, así como la detección de cambios en los estados lógicos que indican la situación para los sensores de apertura y nivel. Debido a la naturaleza de las variables, el SSMA debe estar en permanente funcionamiento para hacer una revisión constante de los estados de las variables en el apiario.

Es preciso, entonces, que el sistema cumpla con las siguientes características(23):

- Alta confiabilidad: un sistema de supervisión y monitoreo requiere estar en constante funcionamiento por un período largo de tiempo y libre de fallas.
- Alta seguridad: el sistema debe ofrecer resistencia a las condiciones climáticas y del entorno propias de las zonas rurales.
- Tiempo real: el sistema no debe presentar demoras de tiempo desde que cambia una variable en el apiario hasta que se visualiza y se registra en el SSMA.
- Facilidad de uso: para hacer más autónomo el sistema se requiere implementar los mecanismos necesarios para ofrecer una interfaz de usuario sencilla y de fácil acceso a la configuración del equipo.
- Más funciones: el dispositivo no sólo se limita a cambios de estado en alarmas, sino también al registro sistemático y confiable de las variables y estados de los diferentes componentes del apiario.

En la figura 2.1 se muestra un diagrama general de como se requiere este configurado el SSMA.

2.4. Definición del problema

La presente propuesta busca desarrollar un dispositivo que permita supervisar y monitorear algunas variables presentes en el habitat de las abejas, así

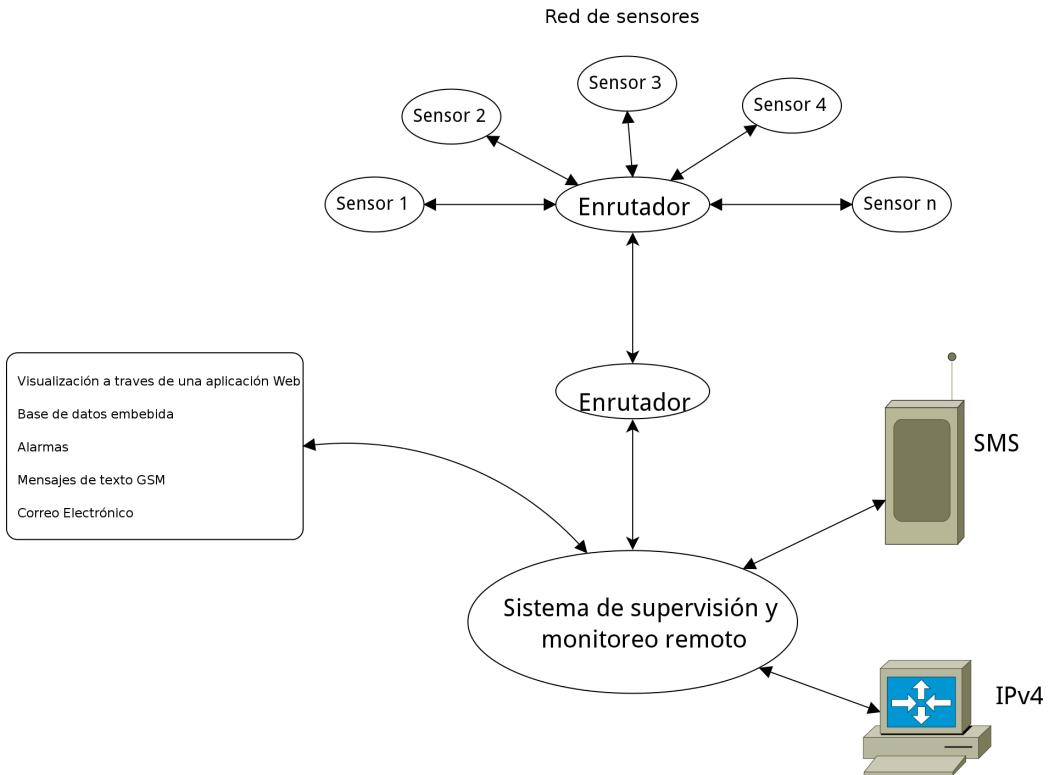


Figura 2.1: Diagrama general del SSMA

como tambien los estados que se han definido previamente respecto al estado de la puerta de la colmena y del tanque de liquido que sirve de alimento. Este dispositivo debe tener la capacidad de conectarse mediante alguna tecnología de red al sistema desarrollado anteriormente como resultado de la tesis *Diseno y construcción de un sistema de medición y trasmisión de información inálambrico para la apicultura*, el cual se encarga de hacer la medición de las variables temperatura y humedad, asi como los estados de la puerta y el nivel del tanque, directamente sobre la colmena. Esta sistema esta adecuado para no afectar los procesos normales del enjambre, permitiendo asi obtener información real y consistente de las variables en medición.

Esta conexión permite que el sistema a desarrollar pueda tomar decisiones en caso de cambio o fluctuación de las variables en medición, ademas de los estados de los sensores de nivel y apertura. Entre estas decisiones se encuentra el envío de mensajes de texto a traves de la red celular GSM y de correo

electrónico a través de la red Internet.

Con la información recolectada de la actividad de las colmenas, también se busca crear un registro histórico del proceso para poder determinar comportamientos o situaciones de vital importancia para la colmena, lo cual permite utilizar herramientas estadísticas para analizar las relaciones entre los cambios de las variables observadas y el comportamiento del enjambre.

Otra de las características del sistema SSMA, se basa en la interpretación de la información que envía el SMTIA, la cual se debe presentar de forma adecuada a las personas que están a cargo del apíario. Se precisa una forma didáctica o de fácil lectura para el usuario final que no tiene un conocimiento muy avanzado en el manejo de sistemas de información.

Capítulo 3

Metodología de diseño e implementación

3.1. Definición de especificaciones

Para lograr con éxito el diseño y la implementación del SSMA, se utilizará una metodología de diseño, que consiste en realizar un análisis de las limitaciones que tiene el entorno donde se va a instalar el sistema, seguido de los recursos disponibles respecto a hardware y software que permitan hacer funcionar el sistema de una forma fácil y confiable. También se tendrán en cuenta los requerimientos del Centro Agropecuario Marengo de la Universidad Nacional de Colombia, quienes determinan el tipo de variables a monitorear y el tipo de usuarios que utilizarán y administrarán en sistema.

Se busca principalmente utilizar herramientas de código abierto o software libre para realizar todo el desarrollo e implementación del Sistema de Supervisión y Monitoreo para la Apicultura.

Inicialmente se analizará la ubicación geográfica del apíario donde se instalará la red de sensores y el SSMA.

El apíario se encuentra ubicado en la finca de la Universidad Nacional de Colombia - Sede Bogotá, Marengo, donde existe una instalación física en la que se requiere instalar el SSMA. Debido a las características del lugar relacionadas con la dificultad de acceso a la red eléctrica, se opta por implementar un sistema que utilice la menor cantidad de recursos a través de un SBC (Single Board Computer).

También se debe tener en cuenta que por la naturaleza del lugar, donde

hay abundante vegetación y fauna silvestre, se debe proporcionar protección para la intemperie, buscando garantizar que el sistema no tenga fallas debido a la acción de insectos, humedad u otro elementos del entorno donde se utiliza la aplicación.

En el apiario ya se encuentran instalados los sistemas encargados de medir y trasmitir la información que se captura de la actividad de las colmenas. Entre estas variables se encuentran la temperatura, humedad, el estado del sensor de nivel del tanque donde se almacena el líquido que beben las abejas y el estado del sensor que indica si hay apertura de la tapa que protege la colmena.

Conocer la temperatura al interior del apiario permite predecir el comportamiento que tendrá el enjambre en un momento específico. La temperatura se debe mantener en un rango definido, generalmente entre los 34 °C y 38 °C. En el momento en que se aumenta este valor por encima del rango establecido, se esperaría que las abejas empiecen a sacudir sus alas para disminuir la temperatura y refrescarse, mientras que con valores menores a los 12 °C, algunas especies de abejas suspenden la producción de huevos, ya que las crias requieren de temperaturas mayores para desarrollarse.

La humedad también cumple un papel importante en la salud de las abejas y está estrechamente relacionado con la variable descrita anteriormente, la temperatura. Mantener un buen nivel de humedad permite a las abejas mantener constante la reproducción y aumentar la población de forma favorable. Se espera una humedad relativa al interior de la colmena del 80 %.

Una apertura innecesaria o sin el acondicionamiento previo de la colmena, puede resultar en graves consecuencias para el enjambre. Por esta razón se necesita que al presentarse un evento de esta naturaleza, se informe inmediatamente al personal responsable del apiario para tomar acciones preventivas o correctivas. El sistema SMTIA, cuenta con los sensores adecuados para informar al SSMA cuando una colmena se encuentre abierta o cerrada para así tomar decisiones en base al estado de la puerta o tapa de la colmena.

Finalmente, también se realizará la supervisión y monitoreo del sensor del nivel del tanque del líquido que sirve de alimento para las abejas, permitiendo una pronta intervención del personal del apiario para evitar inconvenientes derivados de la escasez o falta de agua en la colmena.

Para poder implementar el SSMA, también es necesario contar con una conexión a internet a través de un enrutador de gran alcance, el cual ya se encuentra instalado en el apiario, permitiendo tener un enlace de comunicación

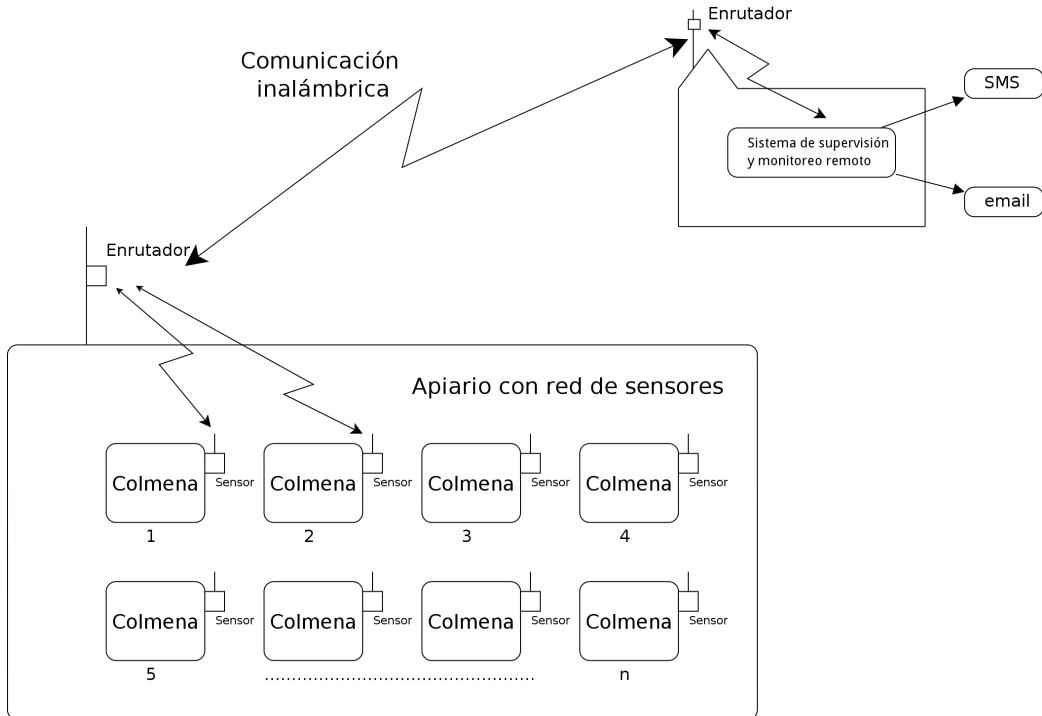


Figura 3.1: Sistema de supervisión y monitoreo para la apicultura

entre la red local de la Universidad Nacional de Colombia Sede Bogotá y el apiario localizado en la finca Marengo.

Ahora haremos un análisis de las características que debería tener el sistema para poder llevar a cabo una efectiva y constante tarea de supervisión y monitoreo.

3.2. Selección de tecnologías

Para cumplir con las características que debe tener un sistema de supervisión y monitoreo para la apicultura, se hará una selección de hardware y software que permita mantener una alta confiabilidad, seguridad, mecanismos para que el sistema trabaje en tiempo real. Finalmente, se busca la interfaz y tecnologías necesarias para que los usuarios que no tienen un amplio conocimiento en el manejo de sistemas, puedan interactuar con el sistema.

correctamente.

3.2.1. Selección de Hardware.

Existen en el mercado distintos tipos de SBC, en los cuales la diferencia la marca el tipo de aplicación para los cuales puedan estar diseñadas, como tambien la capacidad de procesamiento y el tipo de semiconductores que se utilizan.

Fabri-cante	Especificaciones	Soporte
Chum-by HB	Freescale iMX233 454 MHz, RAM 64 MB, 1 GB uSD, I/O pins, microfono, controlador LCD, joystick, serial, usb, A/V,	GCC
Beagle-Board	Texas Instruments ARM Cortex A8 1GHz, RAM 512 MB, uSD, USB Flash, GPIO, serial, stereo in, audio out, ethernet, USB, DVI, LCD	GCC, fuentes Texas Instruments
Friendl-yARM	Samsung S3C6410, RAM 256 MB, 1GB NAND Flash, GPIO, serial, stereo in, audio out, ethernet, USB, LCD, ADC, SPI, IR	GCC, Fuentes FriendlyARM
Android Stamp	Freescale iMX233 454 MHz, RAM 64 MB, uSD, USB, GPIO	GCC, Fuentes Linux en Caja

Tabla 3.1: Selección de Hardware Sistema Embebido

En la tabla 3.1 se hace una comparación de las características de los sistemas embebidos disponibles para el desarrollo de este proyecto. Las tarjetas Chumby HB y FriendlyARM se encuentran disponibles en el Departamento de Ingeniería Electrica y Electronica. Para el caso de la tarjeta Android Stamp, sus archivos de diseño son libres, lo que permite utilizarlos para realizar la construcción física del dispositivo, pero requeriría un trabajo adicional, no previsto en la propuesta de tesis presentada. Varias de estas

En base a la comparación realizada, se elige utilizar la solución ofrecida para FriendlyARM, en su producto Tiny6410.

Se decidió utilizar esta sistema, ya que su forma fisica presenta facilidad para instalar en una caja de paso disponible en el mercado, ya que tiene las perforaciones listas para montar con tornillos estandar y las interfaces de entrada y salida dispuestas de forma sencilla. Adicional a esto cuenta con interfaces de entrada y salida que podran ser utilizadas en implemetación de

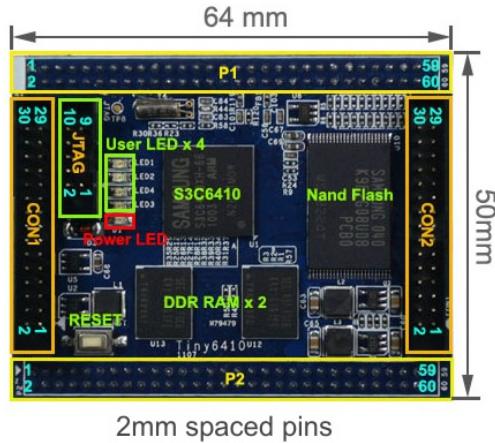


Figura 3.2: Modulo Stamp. Tomado de www.friendlyarm.net

otros modulos para el sistema de supervisión.

Ademas de esto, el paquete que despacha el fabricante, mas especificamente el SDK, viene con pantalla LCD ajustada al dispositivo en cuestion, facilitado tambien la adecuacion de un gabinete o caja para el montaje. La cantidad de memoria RAM tambien fue tenido en cuenta, ya que se van a ejecutar varios servicios de red y manejará gran cantidad de conexiones concurrentes, dependiendo del numero de colmenas y dispositivos que recibiran la informacion proveniente del sistema de supervisión y monitoreo.

Para realizar la implementación del SSMA se utiliza el módulo Stamp Tiny6410, que funciona gracias a un procesador de la empresa Samsung, S3C6410 de arquitectura ARM11 y que presenta las siguientes características:

- CPU: Samsung S3C6410A ARM1176JZF-S VFP-Unit.
- RAM: 256 Mbytes 32 bits bus.
- Flash: 1 Gbyte NAND Flash.
- Serial, SPI,USB,LCD,CMOS Camera interfaces.
- Salidas y entradas analógicas.

En la figura 3.3 se puede observar la disposición física de algunos de sus componentes y sus dimensiones.

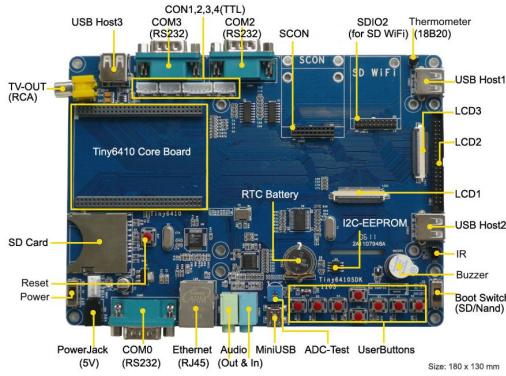


Figura 3.3: Modulo SDK con entradas y salidas. Tomado de www.friendlyarm.net

Para realizar la interfaz con el exterior, Stamp Tiny6410 se puede complementar con el módulo de desarrollo SDK-Board, que cuenta con las siguientes características:

- EEPROM: 1024 Byte (I2C).
- SD-Card Socket.
- Serial ports: 4xDB9 (RS232).
- IR: infrared receiver.
- USB: 3xUSB-A Host 1.1 , 1x miniUSB Slave 2.0.
- LCD: Nec N43 LCD touchscreen.
- Power: 5 V DC.

En la figura 3.3 se pueden observar las características del circuito.

Junto a la tarjeta Stamp y el SDK, se incluyen archivos de desarrollo y manuales de funcionamiento e implementación, los cuales la mayoría están escritos en idioma Mandarin, por lo que el mayor recurso de información de este dispositivo será la red Internet y los sitios especializados en este SBC y sistemas embebidos. Esto es posible gracias a la licencia con que se regula el software que controla el dispositivo.

3.3. Implementación

3.3.1. Conexión inalámbrica.

Uno de los aspectos más importantes para el desarrollo del SSMA es la conectividad que debe tener con redes exteriores y con los SMTIA instalados en el apiario. Por razones de facilidad y compatibilidad se prefiere utilizar un enlace de la norma IEEE 802.11 que permita configurar los dispositivos, interfaces y conexiones fácilmente.

Teniendo en cuenta las capacidades del módulo Stamp y SDK, se elige una tarjeta de red IEEE 802.11 que se conecte vía USB y que tenga soporte para controladores en Linux.

<i>Fabricante</i>	<i>Velocidad</i>	<i>Antena</i>	<i>Chipset</i>	<i>Autenticación</i>	<i>Linux</i>
TP-LINK	150 Mbps	Externa	AR9271	WEP WPA, WPA2	OK
D-LINK	150 Mbps	Interna	RT3070	WEP, WPA, WPA2	OK
D-LINK	150 Mbps	Interna	RT3070	WEP, WPA, WPA2	OK
TP-LINK	150 Mbps	Externa	RT3070	WEP, WPA, WPA2	OK
ENCORE	150 Mbps	Externa	RT9182CU	WEP, WPA, WPA2	OK

Tabla 3.2: Selección de tecnologías Tarjeta WiFi

En la tabla 3.2 se puede observar una comparación de las tarjetas de red disponibles en el mercado para utilizar en el sistema de supervisión y monitoreo.

Inicialmente se utiliza una tarjeta del fabricante TP-Link con chip Ralink RT3070, pero al realizar la compilación de los drivers y hacer pruebas funcionales se observa que no mantiene una conexión estable.

Para solucionar esta falla en la conectividad, se utiliza, entonces, una tarjeta Encore N150, figura 3.4 basada en un chip de Realtek, rt8192cu, que permite compilar sus drivers como módulos, haciendo uso de archivos generados en la compilación del kernel. Al realizar las pruebas funcionales, se observa que ésta sí mantiene la conexión a lo largo del tiempo y permite una configuración rápida de su interfaz.

Para poder utilizar el módulo para la tarjeta inalámbrica realtek, se realizó la construcción de este módulo mediante las herramientas de desarrollo obtenidas en la compilación del sistema de archivos. El código fuente de este controlador se encuentra disponible en el sitio web de Realtek.



Figura 3.4: Tarjeta de red usb wifi Encore N150 (tomado de <http://www.entre-usa.com/mx/product/WU8188CUS2>)

Una vez realizada la compilación de los controladores, se obtienen archivos tipo objeto que proveen al sistema operativo de las rutinas necesarias para establecer una conexión y permitir la transferencia de datos.

Para establecer la conexión se utilizan las siguientes herramientas:

- Ifconfig: permite crear interfaces de red visibles para el kernel.
- Wpa_supplicant: permite realizar la autenticación cuando se trabaja con redes WPA.
- Dhclient: permite obtener una dirección IP automáticamente.

La conectividad inalámbrica se puede configurar al iniciar el sistema mediante un script ejecutado en el proceso de inicialización. El primer archivo de procesamiento en consola, se encarga de inicializar las interfaces de red, mediante el comando ifup:

```
!/bin/sh
#
# Start the network....
```

```
case "$1" in
    start)
        echo "Starting network..."
        /sbin/ifup -a
    ;;
    stop)
        echo -n "Stopping network..."
        /sbin/ifdown -a
    ;;
    restart|reload)
    "$0" stop
    "$0" start
    ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
esac

exit $?
```

Seguido a esta ejecución, se realiza la autenticación ante el enrutador de la red, mediante la colección de herramientas wpa_supplicant:

```
case "$1" in
    start)
        echo "Associating...."
        ifup -a
        /usr/sbin/wpa_supplicant -B -Dwext -iwlan0 -c /etc/wpa_apiario.conf
    ;;
    stop)
        echo -n "Stopping wireless connection..."
        /sbin/ifdown -a
    ;;
    restart|reload)
    "$0" stop
    "$0" start
    ;;
```

```

    *)
echo $"Usage: $0 {start|stop|restart}"
exit 1
esac

exit $?

```

3.3.2. Conexión a la red GSM.

Para poder proporcionar el servicio de envío de mensajes tipo SMS, se elige un módem GPRS de Huawei, que permite acceder a su interfaz a través de protocolos seriales, mediante comandos tipo AT. Se cuenta con librerías para algunos lenguajes de programación que permiten el manejo de estos comandos y de otras funciones del módem. Mediante esta red se hará posible alertar a los responsables o administradores del apíario sobre los eventos que se programarán en el SSMA. Se decidió realizar el sistema de alertas a través de esta red, ya que su cobertura es muy amplia en el territorio nacional y así mismo en zonas rurales donde se podría implementar a futuro esta aplicación.

3.3.3. Compilación de un kernel libre para la aplicación

FriendlyARM proporciona en su sitio Web (<http://www.friendlyarm.net/>), dos imágenes ISO que contienen los archivos de desarrollo para los sistemas Linux, Android y Ubuntu. En estos ficheros se encuentra la versión del kernel de Linux 2.6.38, que viene con la configuración por defecto para la tarjeta.

Inicialmente se hace la compilación de las fuentes con un entorno de desarrollo proporcionado también por el fabricante, pero se encuentran problemas de compatibilidad a la hora de crear un sistema de archivos consistente.

Para obtener consistencia entre los diferentes entornos de desarrollo, se crea, en primera instancia, un sistema de archivos con buildroot que utilice la librería uClib y con la configuración más adecuada respecto a las prestaciones del procesador. De esta forma se crea un entorno de desarrollo, siendo la etapa inicial la creación de ejecutables para hacer la compilación cruzada necesaria.

Para compilar el kernel, se utilizaron las herramientas de desarrollo del sistema de archivos Buildroot. En el directorio de las fuentes del kernel se encuentra el archivo de configuración para el kernel compatible con el kit

Tiny6410, por lo que solo se procedio a ejecutar el comando:

```
linux-2.6.38$ cp config_linux_mini6410 .config
```

Para iniciar el proceso de compilación se ejecuta:

```
make ARCH=arm CROSS_COMPILE=/home/kapak/debian_qt/ \
buildroot-2012.08/output/host/usr/bin/ \
arm-unknown-linux-uclibcgnueabi- -j8
```

En este procedimiento se muestra la ruta tal cual se ejecutó en la maquina de desarrollo anfitrion, variable que varia dependiendo de la ubicacion del compilador cruzado.

Una vez terminadas las tareas de compilación del kernel, se obtiene una imagen *zImage* que se transfiere a la Tiny6410 mediante el software que contiene el kit desarrollo para instalar el kernel en la memoria NAND Flash. Es necesario configurar correctamente los parametros de inicio, los cuales se muestran a continuación:

```
root=/dev/mmcblk0p1 noinitrd console=tty1 console=ttySAC0,115200 \
init=/sbin/init
```

3.3.4. Sistema de archivos

En esta sección, se revisaran las tareas o procesos que se requiere cumpla el SSMA, para así plantear una solución que cumpla con las características descritas anteriormente, teniendo como principal lineamiento el uso de herramientas de software libre.

Para implementar el sistema de supervisión y monitoreo para la apicultura, se requiere un sistema de archivos que contenga las siguientes prestaciones:

- Manejo de interfaces de red.
- Manejo de dispositivos inalámbricos.
- Servicios de red: smtp, http, ssh, ftp.

- Puertos de entrada y salida usb, serial.
- Ethernet.

Para satisfacer cada una de estas necesidades, existen aplicaciones mantenidas por la comunidad involucrada en el software libre, que se utilizan en diversas aplicaciones y arquitecturas, independiente de la aplicación.

En vista de la disponibilidad de herramientas de software para implementar los servicios requeridos, se realizará una comparación entre sistemas de archivos que contiene las aplicaciones listadas en la tabla 3.3.

Se analizó cuáles sistemas de archivos contenían estas prestaciones, de las cuales se aproximaron más a los requerimientos del SSMA las distribuciones listadas en la tabla 3.4.

En razón a la cantidad de recursos que utiliza la distribución, se decide utilizar un sistema de archivos creado con Buildroot.

A continuación se hará una breve descripción de la utilización de los componentes listados en la tabla 3.3 y que tarea va a desarrollar en el SSMA.

3.3.4.1. Red

3.3.4.2. wpa_supplicant

Es una implementación en software libre de un suplicante IEEE 802.11i para Linux, FreeBSD, NetBSD y Microsoft Windows. Aparte de la implementación para el sistema de autenticación WPA2, también se cuenta con capacidades de autenticación en WPA y otros protocolos antiguos. Entre sus funcionalidades se encuentran (21):

- WPA y IEEE 802.11i/RSN/WPA2 completo.
- WPA-PSK y WPA2-PSK
- WPA con EAP.
- Manejo de llaves para CCMP, TKIP, WEP.

<i>Modulo</i>	<i>Funcion</i>	<i>Aplicación</i>
Buildroot	Red	wpa_supplicant
		ip
		ifconfig
		dhclient
		ssh
		wget
		ping
	Base de datos	sqlite3
		Python API
		C/C++ API
	Aplicacion WEB	Python2.7
		Webpy
		Rgraph
		Javascript
	Mensajes SMS	Python2.7
		PySerial
		PyHumod
	Mensajes correo	Python2.7
		smtplib
	Utiilidades terminal	ls
		rm
		mv
		cat
		grep
	HMI	Qt 4.8
		Fancybrowser
		tslib

Tabla 3.3: Elementos de software requeridos para implementar el SSMA

<i>Distribucion</i>	<i>Tamaño</i>	<i>Uso de memoria</i>
DebianEmbedded	256 MB aprox	120 MB aprox
ArchLinuxARM	157 MB aprox	120 MB aprox
Ubuntu	400 MB aprox	180 MB aprox
Buildroot	40 MB aprox	40 MB aprox

Tabla 3.4: Comparación de sistemas de archivos

3.3.4.3. ip

Mediante este comando se puede configurar el enrutamiento, dispositivos y tuneles de red. Se puede encontrar mas información acerca de las opciones de este comando en los manuales de linux o mediante el comando *man ip*.

3.3.4.4. ifconfig

ifconfig es un programa disponible en varias versiones del sistema operativo UNIX, que permite configurar o desplegar numerosos parámetros de las interfaces de redes, como la dirección IP (dinámica o estática), o la máscara de red. Si se llama sin argumentos suele mostrar la configuración vigente de las interfaces de red activas, con detalles como la dirección MAC o el tráfico que ha circulado por las mismas hasta el momento (17).

3.3.4.5. dhclient

El cliente DHCP de Internet Software Consortium, proporciona los mecanismos para configurar una o mas interfaces de red usando el protocolo DHCP y BOOTP (2). Con este cliente el sistema puede tomar una dirección IP asignada automáticamente desde el enrutador que se utilizará para la aplicación.

3.3.4.6. ssh

SSH (Secure SHell), en castellano, intérprete de órdenes segura, es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de sistema de ventanas X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix y Windows) en ejecución (20).

3.3.4.7. wget

GNU Wget es una herramienta libre que permite la descarga de contenidos desde servidores web de una forma simple. Su nombre deriva de World Wide Web (w), y de «obtener» (en inglés get), esto quiere decir: obtener desde la WWW. Actualmente soporta descargas mediante los protocolos HTTP, HTTPS y FTP (16).

3.3.4.8. ping

Ping es una utilidad en redes de computadoras que comprueba el estado de la conexión del host local con uno o varios equipos remotos de una red TCP/IP por medio del envío de paquetes ICMP de solicitud y de respuesta. Mediante esta utilidad puede diagnosticarse el estado, velocidad y calidad de una red determinada (19).

3.3.5. Base de datos

3.3.5.1. sqlite3

Base de datos embebida local para el registro historico de las variables en observación y de los parametros de configuración del SSMA.

3.3.6. Aplicación Web

3.3.6.1. Python

Permite desarrollar aplicaciones de manera rapida, permitiendo compatibilidad entre plataformas. De esta manera se podria cambiar rápidamente la plataforma de hardware sin necesidad de hacer grandes cambios en el software.

3.3.6.2. Webpy

Framework de Python que permite diseñar aplicaciones web, que permiten interactuar con otros procesos o módulos del sistema, manteniendo el paradigma de programación en que esta basado Python. Se utiliza este entorno de desarrollo dada su facilidad de uso y que existe bastante documentación para poder escribir una aplicación desde cero.

3.3.6.3. Rgraph

Es una libreria de graficas para HTML5, que usa la etiqueta *canvas* para dibujar. Soporta diferentes tipos de graficas, creando estas representaciones en un navegador web utilizando el soporte para Javascript, reduciendo la carga al servidor y optimizando el tiempo de renderizado (9).

3.3.6.4. Javascript

Es un lenguaje de programacion interpretado, dialecto del estandar ECMAScript, ampliamente difundido en clientes de navegación web. Es orientado a objetos, basado en prototipos, imperativo y dinamico. Se usa principalmente en el lado del cliente, permitiendo mejoras en interfaz de usuario y paginas web. Tiene una sintaxis similar a C, aunque utiliza convecciones o palabras del lenguaje de programación Java (18).

3.3.7. Mensajes SMS

3.3.7.1. Python

3.3.7.2. PySerial

Este modulo para Python, encapsula el acceso al puerto serial de la maquina en que se ejecuta. Entre sus caracteristicas se encuentra (14):

- Manejo de clases iguales, independiente de la plataforma.
- Acceso a puertos a traves de propiedades de Python.
- Escrito totalmente en Python.
- El puerto se configura para trasmision en binario.
- Manejo de puerto similar al manejo de archivos.

3.3.7.3. PyHumod

Paquete para Python que permite conectar redes IP y enviar mensajes de texto SMS con un modem Huawei(13).

3.3.8. Mensajes de correo electronico

3.3.8.1. Python

3.3.8.2. Smtplib

3.3.9. Utilidades terminal

Las utilidades de terminal las provee el sistema Busybox, que combina pequeñas versiones de las utilidades mas comunes de UNIX en un solo ejecutable. Proporciona reemplazos para la mayoria de utilidades que generalmente se encuentran en GNU fileutils, shellutils, etc. Ha sido escrito buscando resudir el tamaño y poder ejecutarse con bajos recursos, por lo que se considera especial para sistemas embebidos (4).

3.3.10. HMI

3.3.10.1. Qt 4.8

Es una biblioteca multiplataforma usada para desarrollar apliaciones con interfaz grafica. Esta desarrollada como software libre y codigo abierto a traves de Qt Project, donde participa la comunidad como Nokia, Digia y otras empresas. Funciona en todas las plataformas principales. El API para acceder a la biblioteca cuenta con metodos para acceder a bases de datos, gestion de procesos, redes, manipulación de archivos, entre otras tareas.

3.3.10.2. Fancybrowser

Gracias al sistema de archivos creado por Buildroot, se implementaron las librerias compartidas para ejecutar apliaciones QT, lo que nos permite utilizar algunos recursos disponibles en la red para implementar un navegador que permite visualizar la interfaz web en el SSMA. En los ejemplos de Qt 4.8, se encuentra fancybrowser, aplicación con la cual vamos a lograr visualizar la informacion que devuelve el navegador QT.

3.3.10.3. Tslib

Es una capa de abstracción para eventos de paneles touchscreen y una pila de procedimientos para manejar estos eventos. Generalmente se usa en

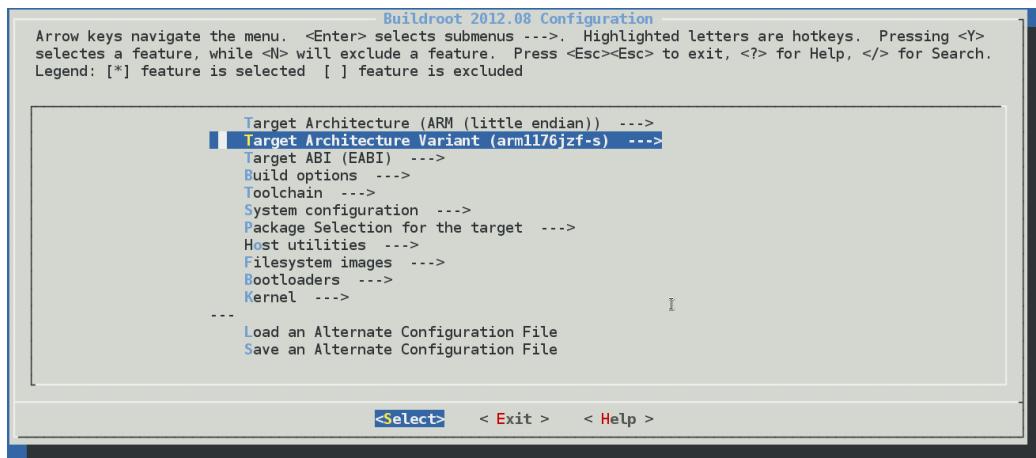


Figura 3.5: Menú de configuración de buildroot.

sistemas embebidos para proporcionar al espacio de usuario toda la funcionalidad del touchscreen (11).

Gracias a la comunidad involucrada en el desarrollo de buildroot, en las aplicaciones y binarios ofrecidos se pueden encontrar soluciones para las necesidades del SSMA.

Se decide hacer uso de la herramienta buildroot, la cual permite configurar a punto el sistema de archivos, evitando crear binarios innecesarios para la operación del SSMA, utilizando de forma más eficiente los recursos. Adicionalmente, los SMTIA usan un sistema de archivos basado en buildroot, por lo que permite obtener mayor compatibilidad respecto a las aplicaciones requeridas por ambos sistemas.

Uno de los aspectos mas importantes para implementar un sistema de archivos correcto para el funcionamiento del sistema, es especificar correctamente la arquitectura y tipo de procesador sobre la cual se va ejecutar la aplicación. En la figura 3.5 se muestra el tipo de sistema que se necesita para realizar la compilación.

Una vez terminadas las tareas de compilación para buildroot, se obtiene un archivo tipo tar, que contiene toda la estructura del sistema de archivos. Para que inicie correctamente esta distribución, se deben configurar algunos scripts y crear nodos del sistema de archivos /dev, que representa los dispositivos disponibles en el sistema embebido.

En este caso se obtuvo el archivo *rootfs.tar.bz2* que mediante el comando

```
tar jxvf rootfs.tar.bz2 -C /mnt/arm/
```

y habiendo montado la tarjeta SD en el directorio */mnt/arm* despliega todos los ficheros necesarios para ejecutar la aplicaciones requeridas correctamente.

3.3.11. Base de datos para el manejo de la información.

Nombre	Tipo	Orientado a	Licencia
CouchDB	NoSQL	Documentos	Apache 2.0
MongoDB	NoSQL	Documentos	GNU AGPL v3.0
MySQL	SQL	Registros	GPL
PostgreSQL	SQL	Registros	PostgreSQL
SQLite3	SQL	Registros	Public Domain

Tabla 3.5: Selección de sistema de base de datos

Una de las tareas principales del SSMA consiste en el registro constante de los sensores de humedad, temperatura, vibración y apertura. Sin embargo, también se requiere generar un mecanismo que registre todos los eventos que ocurren en la colmena(12). Para ello, se utiliza una abstracción de la información, en forma relacional, para hacer posible el uso de una base de datos.

Aunque en un inicio se plantea la posibilidad de usar un servicio de red que soporte sentencias de SQL, finalmente se elige una base de datos embebida local, con el fin de usar la menor cantidad de energía.

En términos de aplicación, las bases de datos embebidas de pueden dividir en dos grupos(23):

- Bases de datos con funciones de manejo de datos distribuidos.
- Bases de datos dedicadas, diseñadas especialmente para aplicaciones en específico.

Las bases de datos embebidas generalmente están asociadas a funciones de sistemas operativos, por lo que deviene innecesario el uso de otras aplicaciones para su funcionamiento.

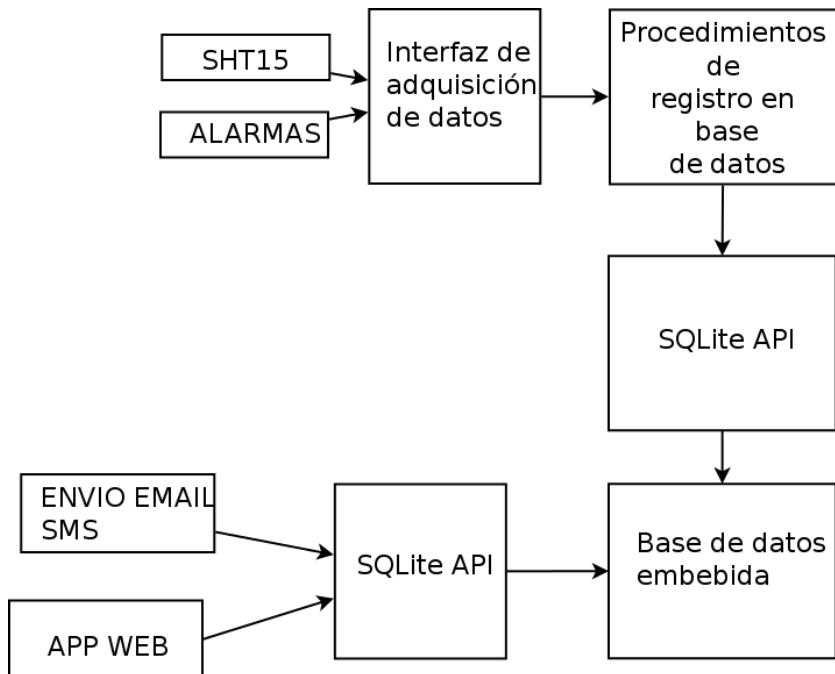


Figura 3.6: Arquitectura para la base de datos relacional del SSMA

Como sistema de base de datos para almacenar la información proveniente del apiario, se utiliza el motor de base de datos SQLite.

SQLite es un motor de base de datos embebido, con una librería que implementa una base de datos SQL autocontenido, sin servidor y sin necesidad de configuración, lo que permite su fácil distribución en redes de sensores. Adicionalmente, no necesita procesos externos y todas sus operaciones se hacen mediante la escritura y la lectura de un archivo en disco. Este motor facilita las tareas de recolección y análisis de datos.

Este sistema cuenta con las siguientes características:

- Pequeño y rápido: binario de máximo 300kB cuando están todas las funciones activadas.
- Buen desempeño en sistemas con pocas cantidades de memoria.
- Código abierto.

Para tener una idea más clara de la forma en que se relacionan los componentes del sistema con la base de datos embebida, podemos analizar la figura 3.6 en la cual encontramos:

- Interfaz de adquisición de datos: mediante aplicaciones hechas en lenguaje C en el SMTIA se hace la lectura del sensor SHT15, que proporciona información de temperatura y humedad, así como los GPIO que representan los sensores de nivel y apertura.
- Procedimientos de registro de datos: aplicación en Python para almacenar registros compuestos por número, identificación de la colmena, hora de muestra, temperatura, humedad, estado del sensor de nivel y estado del sensor de apertura.
- SQLite API: se usa el módulo SQLite, el cual proporciona una interfaz DB 2.0 de fácil implementación.
- Base de datos embebida: archivo en que se almacena la información de forma binaria.

Teniendo en cuenta las variables a observar en el apíario, se necesita implementar la base de datos que contenga los campos adecuados para almacenar de forma organizada la siguiente información:

- Variables de cada colmena, como lo son nombre, marca de tiempo, temperatura, humedad, nivel y apertura.
- Información de colmena, entre las que encontramos nombre de la colmena, numero MAC del SMTIA instalado en cada colmena y la dirección ip de SMTIA con la que se configura cada colmena dentro de la red.
- Informacion del apíario, como nombre, ubicación geografica, numero de colmenas en el apíario, nombre del administrador del apíario, correo electronico y contraseña de acceso.
- Usuarios de alarmas, donde se almacena información de nombre, teléfono al cual enviar alerta por SMS, correo electronico, mensaje a enviar para alertas de nivel y de apertura.

En la figura 3.7 se observa la estructura de la base de datos relacional que consultaran las aplicaciones desarrolladas para el SSMA, representada mediante un mapa mental.

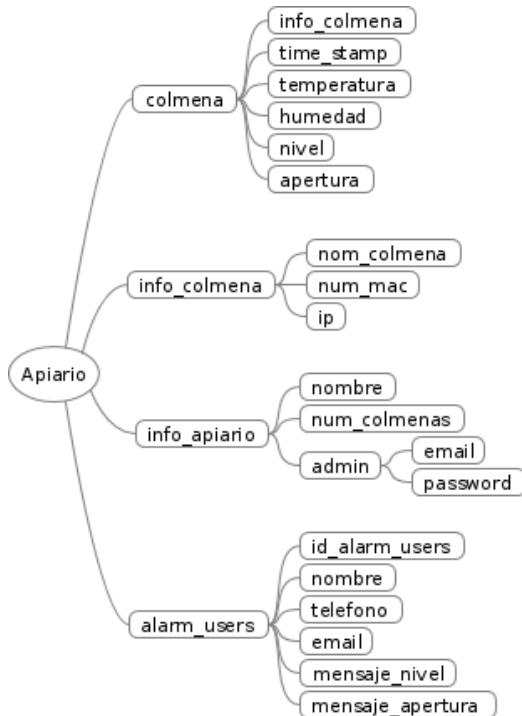


Figura 3.7: Estructura de la base de datos relacional que se utilizará para la aplicación

3.3.12. Servicios que permitan la recepción de datos provenientes de la red de sensores.

3.3.12.1. Servidor UDP

Mediante el protocolo UDP asociado a la capa de transporte del modelo OSI, se implementa un sistema de intercambio de datagramas, estructurado en forma pregunta-respuesta para consultar los valores leídos por los sensores de la colmena. En el SMTIA, mediante lenguaje C, se construye una aplicación tipo servidor que permanece a la espera de la petición de un cliente UDP. Para establecer la comunicación, el cliente envía el comando QUERY, al cual el servidor responde transfiriendo una serie de datagramas con la información solicitada.

Se hace la elección de este protocolo debido a las siguientes características:

- Transferencia de datos muy rápida.
- Flexibilidad.
- Adecuado para transmisión de pequeños datos.

El objetivo del SSMA es concentrar la información de la red de sensores de las colmena. Por ello, es preciso buscar la forma de conectar los dispositivos instalados en cada colmena con el SSMA, con el fin de disponer de un acceso confiable y constante de las variables en medición.

El SMTIA se ejecuta sobre una tarjeta Chumby ARM, que soporta la ejecución de aplicaciones en lenguaje C, permitiendo utilizar una aplicación de bajo de consumo de memoria y desempeño aceptable.

Para realizar esta aplicación se utilizó lenguaje C, compilando las fuentes con las herramientas de desarrollo resultado de la creación del sistema de archivos con Buildroot. Es necesario que esta aplicación ejecute las siguientes tareas(24):

- Crear un mecanismo para realizar una conexión de red con otros dispositivos, en este caso el SMTIA. Para implementar esta tarea se utiliza la librería socket.h que proporciona C.
- Consultar el sensor de temperatura y humedad SHT15 cuando se realice una petición del cliente, para no mantener en constante lectura del sensor.
- Organizar la información obtenida del sensor, teniendo en cuenta el tipo de datos obtenido y el tipo de dato a trasmisir.
- Responder a una petición por parte del cliente cuando se envía un datagrama con la palabra "Query".

Teniendo los requerimientos para este módulo del sistema se realizó un diagrama de flujo para tener una descripción más general de las tareas a ejecutar. Este diagrama se encuentra en la figura 3.8.

3.3.12.2. Cliente UDP

En función de las tareas que se desean realizar, se puede implementar un cliente para que consulte los datos que proporciona el SMTIA, enviando

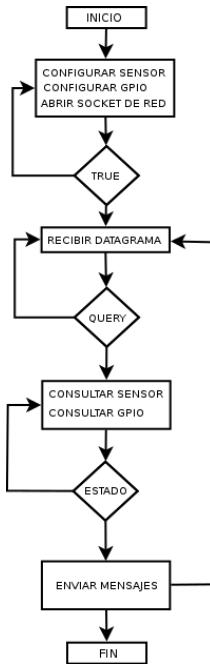


Figura 3.8: Diagrama de flujo que describe las tareas a ejecutar por el servidor UDP en Chumby

la sentencia QUERY y recibiendo datagramas con tamaños fijos adecuados para el diseño. Ya que se establece comunicación con un socket de red, el cliente se puede desarrollar en cualquier lenguaje que soporte este tipo de conexiones.

Para realizar las consultas al servidor instalado en el SMTIA, se puede utilizar el lenguaje interpretado Python, que contiene tambien una libreria especializada en la creacion de sockets y establecimiento de conexiones de red. La libreria para desarrollar este procedimiento se conoce como el módulo socket.

Entre las tareas que necesitamos que cumpla este módulo de la aplicación se encuentran:

- Establecer un socket de red en el SSMA.
- Realizar la conexión al servidor, previo conocimiento de la dirección de red del SMTIA y de su puerto de comunicación.

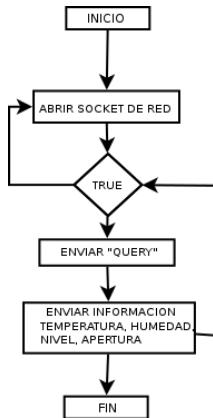


Figura 3.9: Diagrama de flujo que describe las tareas a ejecutar por el cliente UDP en FriendlyARM

- Enviar la palabra “QUERY” para iniciar el envío de datagramas por parte del servidor en el SMTIA.
- Recibir de forma secuencial y ordenada la información preparada por parte del SMTIA acerca de la temperatura, humedad y estado de los sensores que se representan por los valores leídos en los GPIO.

Al igual que para el servidor UDP, se puede describir la operación general del sistema mediante un digrama de flujo en la figura 3.9.

3.3.13. Aplicaciones que permiten enviar mensajes de texto o correo electrónico.

Es requerimiento del sistema contar con un mecanismo que permita enviar mensajes de texto y correo electrónico en el caso de alarmas, las cuales responden a cambios de nivel en el tanque de alimentación de las colmenas y al sensor magnético instalado en la tapa de las mismas.

Teniendo en cuenta el servidor implementado en el sistema de toma de datos, se desarrolla una aplicación que consulta las variables adquiridas, actuando como cliente UDP. Al recibir la información, realiza un análisis de los datagramas y en base a ello determina la alarma a enviar. Con el fin de no enviar mensajes indiscriminadamente, se lleva a cabo la comprobación entre

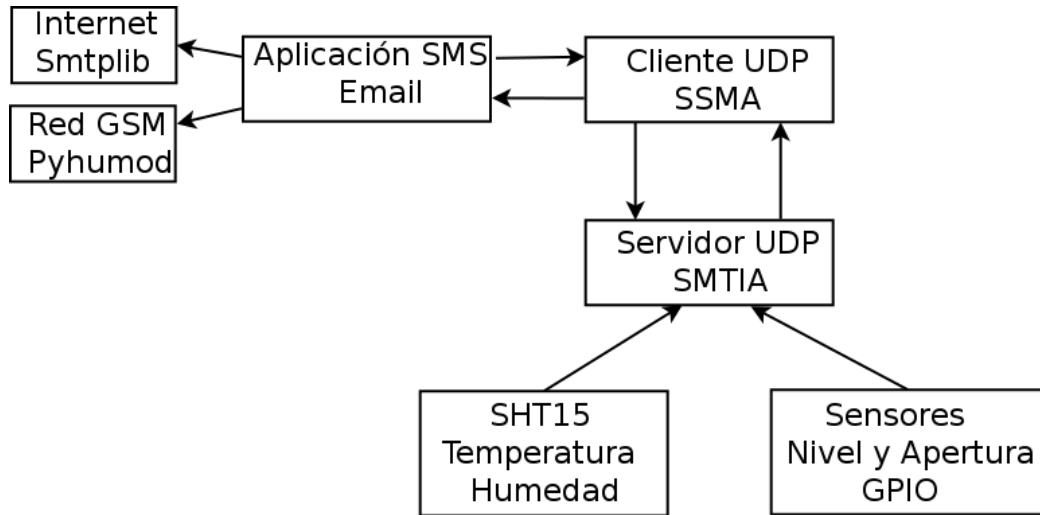


Figura 3.10: Diagrama de bloques

el tiempo actual y el tiempo en que se envió la última alarma, buscando no sobrecargar innecesariamente el sistema.

La información concerniente a los mensajes de alarma se puede configurar desde la interfaz hombre máquina, que los almacena en la base de datos implementada. Cuando se activa una alarma se envía un mensaje a todos los usuarios registrados en la base de datos.

3.3.13.1. SMS

Utilizando Python se facilita el desarrollo de la aplicación de envío de alertas, ya que existen librerías de manejo de hardware para el envío de mensajes de texto a través de la red GSM (15). En la implementación de este mecanismo se utiliza el paquete Pyhumod de Python, que implementa las rutinas adecuadas para el manejo de módems Huawei. De esta librería sólo se utilizan las tareas de inicialización y envío de mensajes de texto, aunque también permite usar otras funciones para establecer conexiones de red.

En la documentación de esta librería se muestran ejemplos para realizar el envío de un mensaje de texto vía SMS, previa configuración e inicialización del modem Huawei GPRS:

```
$ python
```

```
Python 2.5.4 (r254:67916, May 21 2009, 22:07:14)
[GCC 4.1.2 (Gentoo 4.1.2 p1.0.2)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import humod
>>> m=humod.Modem()
>>> m.show_model()
'E270'
>>> m.enable_textmode(True)
>>> m.sms_send('+353?????????', 'hello world')
52
>>> m.connect()
>>> m.disconnect()
```

Esta librería implementa a bajo nivel, mediante la escritura y lectura del puerto serial al cual se asocia el modem GPRS, una serie comandos AT que permiten alistar el modem y configurarlo para enviar un mensaje de texto de forma correcta. Para la aplicación inicialmente se realizaron pruebas escribiendo directamente al puerto serial, verificando su correcto funcionamiento para luego elegir esta librería con el fin de disminuir el tiempo de desarrollo y permitiendo mayor modularidad en la aplicación. Este elección permite que se pueda implementar envío de mensajes de texto SMS en otras aplicaciones con requerimientos similares.

Se puede descargar el código fuente y la documentación de pyhumod en <http://code.google.com/p/pyhumod/>.

3.3.13.2. Email

La aplicación permite, de la misma forma, el envío de correos electrónicos a los usuarios que están registrados en la base de datos. Para implementar este mecanismo se utiliza el módulo `stplib` de Python para manejo del protocolo SMTP.

El contenido del mensaje es el mismo que se configura mediante la interfaz HMI para avisos de alerta.

En la red Internet, se observa que esta librería tiene amplia difusión para el manejo de envío de correos electrónicos. También se muestra un pequeño ejemplo de la implementación de un programa que permite enviar un mensaje a través de la red Internet:

...

```
def send_email_alarm():
    conn = sqlite3.connect('../database/data.sqlite3')
    cur = conn.cursor()
    cur.execute("select * from alarm_users")

    for row in cur:
        name = row[1]
        email = row[3]
        msg = row[4]
        #message = name+ ","+ msg
        subject = "Subject:ALARMA"
        header='To: ' + email + '\n' + 'From: ' + ACCOUNT + '\n' \
        + subject+'\n'
        message=header+name+", "+msg
        print message
        server = SMTP(HOST, PORT)
        server.ehlo()
        server.starttls()
        server.ehlo()
        server.login(ACCOUNT, PASSWORD)
        try:
            server.sendmail(ACCOUNT,email,message)
            server.close()
            print "Successfully sent email"
            sleep(1) #sleeps for 60 seconds
        except smtplib.SMTPException:
            print "Error: unable to send email"
    cur.close()
    conn.close()
    ....
```

3.3.13.3. Servidor de alarmas

Teniendo en cuenta las prestaciones que nos entrega las librerías mencionadas anteriormente, es necesario desarrollar una aplicación que realice las siguientes tareas(5) (22):

- Crear un socket de red en un puerto específico.
- Permanecer a la espera de un datagrama con información de los estados de los sensores que representan los valores de los GPIO del SMTIA.
- En base a estos estados, enviar un mensaje de texto a través de la red GSM y un correo electrónico a través de la red Internet.
- Dependiendo de la marca de tiempo de la última alarma, enviar nuevamente un mensaje, teniendo cuidado de no enviar demasiados mensajes al usuario.

Podemos representar el funcionamiento general de este módulo mediante el diagrama de flujo que se muestra en la figura 3.10.

En este caso se implementó la lectura de la base de datos para consultar los correos electrónicos a los cuales enviar la alarma para los eventos de nivel y apertura.

3.3.14. Configuración de la red de datos inalámbrica que conecta el apíario con el centro de mando.

Para la configuración de la red inalámbrica se elige un esquema de direcciones IP fijas con el fin de garantizar que las colmenas tengan siempre la misma dirección. Mediante un router inalámbrico TP-Link se establece comunicación con el SMTIA. El sistema de archivos creado con buildroot contiene todas las aplicaciones necesarias para establecer y controlar este enlace, como se describió en la sección de conexión inalámbrica.

En la figura 3.11 se describe la topología de red que se siguió para implementar el SSMA en conjuntos con los sensores instalados en cada colmena (SMTIA).

3.3.15. Interfaz hombre maquina HMI.

Con el fin de utilizar un mismo proceso para hacer accesibles los datos al usuario final, se desarrolla una aplicación Web basada en Webpy de Python, interfaz que puede ser consultada de forma local y remota mediante un navegador que tenga soporte para HTML5 y Javascript.

La estructura de la interfaz se muestra en la figura 3.12.

En esta aplicación se implementan las siguientes secciones:

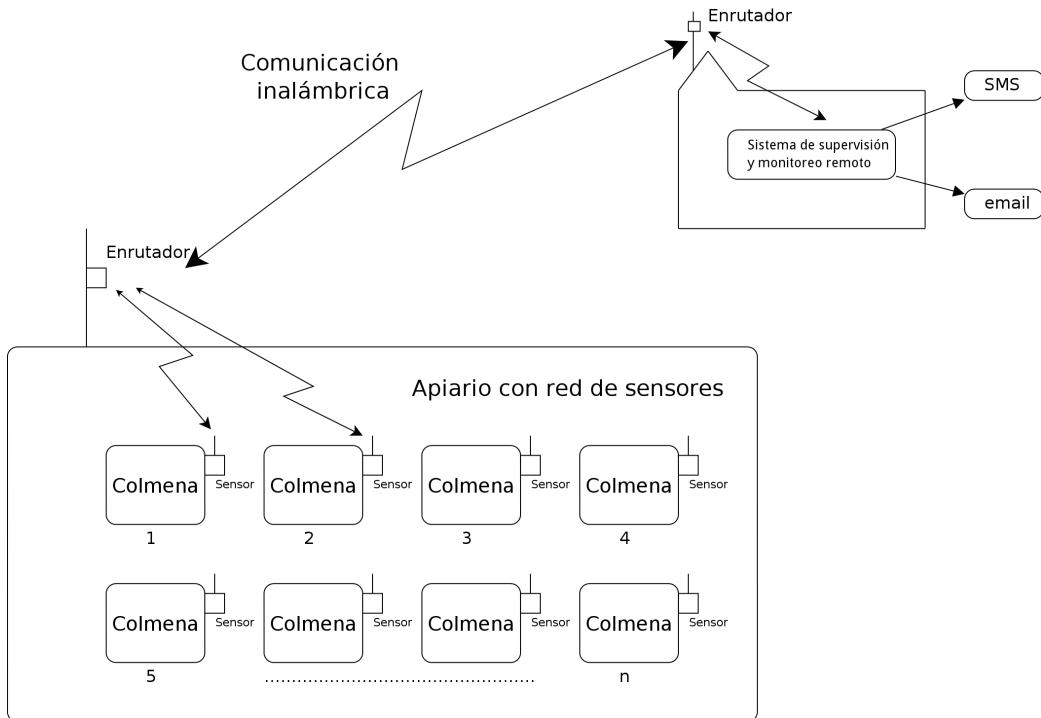


Figura 3.11: Topología de red para conectar los SMTIA a SSMA

- Colmena: visualización de variables y estados de los sensores.
- Config alarm: configuración de los mensajes y números de teléfono para enviar avisos de alarmas.
- Utilities: sección que permite la descarga de la base de datos en un dispositivo extraíble USB.
- Config colmena: sección que permite asignar nombres, direcciones IP y números MAC a las colmenas que hacen parte del apiario.

La aplicación Webpy realiza una consulta al servidor UDP instalado en el SMTIA mediante un cliente UDP descrito anteriormente, realizando un análisis de los datagramas recibidos para generar una página Web renderizada desde esta misma aplicación usando elementos de HTML5, JS y CSS. Por la naturaleza de las variables en observación, los tiempos entre consultas al

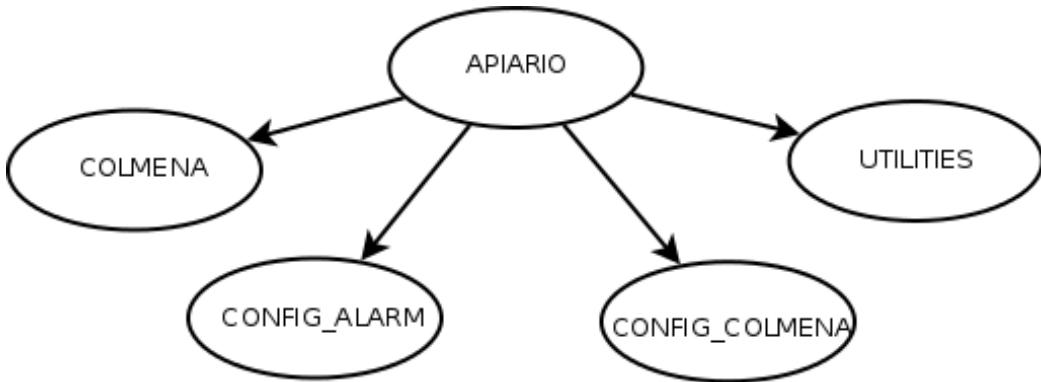


Figura 3.12: Estructura de la interfaz hombre máquina

servidor UDP en el SMTIA pueden ser de 10 a 30 segundos. Mediante esta plataforma también se muestra un formulario para la configuración de las alarmas, el acceso a utilidades del sistema y la futura implementación de otros servicios.

Se decide desarrollar este tipo de arquitectura para poder implementar fácilmente otros módulos sin necesidad de un conocimiento muy detallado de los aspectos internos del SSMA.

Para poder visualizar localmente la aplicación Web, se implementa un navegador standalone basado en QT y WebKit, que permite tener acceso a toda la interfaz hombre máquina.

3.3.15.1. Aplicaciones Web

Para poder proporcionar el valor de las variables en medición al usuario final, se dispondrá de una aplicación Web (7) basada en Python, específicamente en el framework Webpy, que permite realizar aplicaciones que usan también recursos del dispositivo sin limitarse solo a la interfaz o protocolo web. Mediante esta librería de Python, podemos seleccionar la forma en que se van a manejar los datos que arroja el cliente UDP cuando consulta al SMTIA para tener conocimiento de las variables observadas. El programa principal de esta aplicación, utiliza las primitivas del protocolo HTTP para obtener y enviar información al cliente o navegador. Cuando el navegador web en cliente solicita una página, a nivel de protocolo se utiliza una sentencia GET, que permite procesar todo el contenido que se mostrará en la página web, dada

la lectura de las variables que arroja el cliente UDP u otros subprogramas.

Las tareas que realiza la aplicación Web se realiza en la siguiente secuencia:

- Desde el navegador se solicita la pagina Web.
- La aplicación web recibe una petición GET.
- Cuando se recibe esta sentencia, se ejecuta el cliente UDP
- El servidor UDP del SMTIA, recibe la palabra QUERY y envia una secuencia de datagramas al cliente.
- El cliente devuelve a la aplicación webpy, informacion en variables acerca de los datos que obtuvo del SMTIA.
- Webpy se encarga de renderizar el contenido web con esta información.
- Mediante elementos de HTML5 y JS, se dibujan indicadores para representar la información obtenida.
- Cada 5 segundos, se solicita nuevamente a la aplicación web que repita el anterior proceso.

3.4. Construcción del dispositivo

Para montar el SSMA en el apiario, se busco una caja de algun material resistente, preferiblemente en plastico, para hacer resistente el sistema a la acción de la humedad, insectos o material vegetal que pudiera afectar el funcionamiento del SBC.

Finalmente se utilizo una caja TIBOX, con grado de protección IP 56, la cual viene con una bandeja de metal adecuada para fijar la tarjeta mediante postes de plastico. A su vez, cuenta con una tapa desmontable, que permite acceder al interior facilmente y a su vez, cuando se asegura la tapa, presenta resistencia a chorros de agua o filtraciones.

La caja no viene con perforaciones para hacer accesibles algunos conectores, por lo que se procedio con ayuda de un motor tool a realizar los agujeros necesarios, especialmente para puesto USB, antena de la tarjeta de red Wifi, alimentación de 5 VDC e interruptor de encendido y apagado.



Figura 3.13: Imagen del montaje de la tarjeta en la caja de paso



Figura 3.14: Imagen del montaje de la pantalla LCD en la caja de paso



Figura 3.15: Imagen del montaje de la pantalla LCD en la caja de paso

Capítulo 4

Pruebas y resultados

El contenido de esta capítulo se basa en los resultados obtenidos en laboratorio, de los cuales se hará una breve descripción de las pruebas realizadas basadas en graficas obtenidas en la maquina de desarrollo y del sistema construido.

Las pruebas se fueron realizando en la medida que cada modulo era terminado, con fin de ir depurando partes pequeñas del sistema, para finalmente integrar todos los modulos y realizar las pruebas correspondientes de funcionamiento.

Para realizar las pruebas en laboratorio, se utilizara la configuración prevista para el apario. Se utilizara el mismo enrutador que se utilizo para el desarrollo del SMTIIA (8). Como este ultimo sistema ya cuenta con la configuración de red lista para funcionar, solo faltaria configurar el router para trabajar con el ESSID “APIARIO”. La clave utilizada fue *apiariogmun2012*, con autenticación WPA. El SSMA tambien se configura para trabajar con este punto de acceso, teniendo una dirección ip fija.

El diagrama de conexiones de los sistemas instalados en el apario se muestra en la figura 4.1. Como se realizarán pruebas de laboratorio, se utiliza la configuracion de la figura 4.2

En la figura 4.1 se observa la configuracion de los dispositivos involucrados con el sistema de supervisión y monitoreo. Para realizar las pruebas se utilizo el dispositivo resultado de la tesis *Diseño y construcción de un sistema de medición y transmisión de información inalámbrica para la apicultura* y el enrutador utilizado para conectar a la red este dispositivo.

Teniendo en cuenta el trabajo realizado en la asignatura Taller de Proyec-

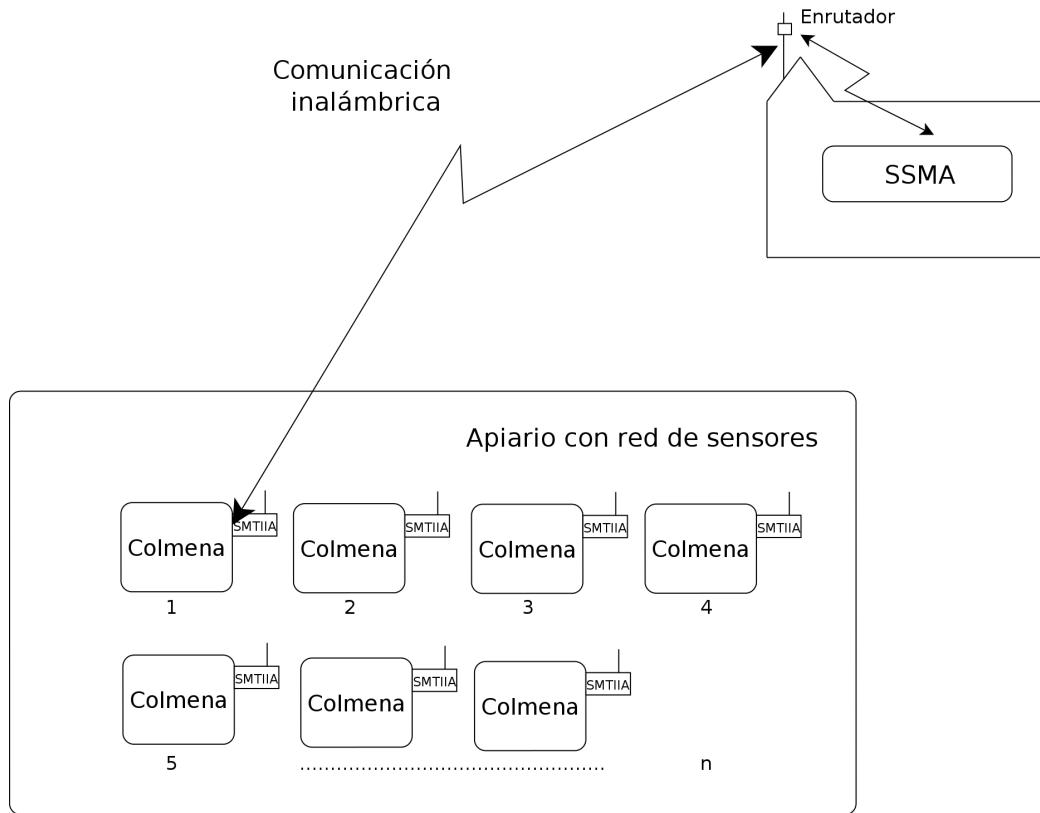


Figura 4.1: Topología de red para conectar los SMTIA a SSMA

tos Interdisciplinario, bajo la tutoría de Giovanni Baquero Rozo, *Diseño de sondas para medición de variables físicas y alarmas en el interior de una colmena apícola*, se utilizó el conjunto mostrado en la imagen 4.4, en el que se conecta la estructura que soporta los sensores de la colmena con el SMTIA. Este sistema toma el valor de temperatura y humedad directamente de los sensores y prepara una serie de datagramas, que se transmiten al sistema de supervisión y monitoreo a través del servidor UDP implementado.

En la figura 4.5 se especifica las tareas que va a realizar cada sistema, indicando la aplicación que finalmente queda instalada y ejecutándose para lograr el objetivo del presente trabajo.

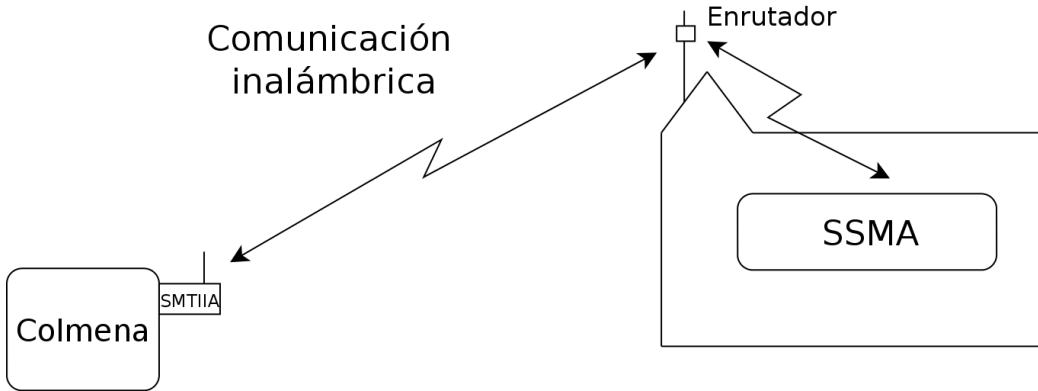


Figura 4.2: Topología de red para conectar los SMTIA a SSMA en pruebas de laboratorio

4.1. Conexión inálambrica

En esta sección se muestra la puesta a punto de la conexión de red, basándose en los mensajes de depuración del kernel de linux. A continuación se muestra la depuración de carga en memoria de los controladores de la tarjeta de red inálambrica:

```
rtw driver version=v3.4.4_4749.20121105
Build at: Nov 16 2012 17:58:46
register rtw_netdev_ops to netdev_ops
CHIP TYPE: RTL8188C_8192C

usb_endpoint_descriptor(0):
bLength=7
bDescriptorType=5
bEndpointAddress=81
wMaxPacketSize=40
bInterval=0
RT_usb_endpoint_is_bulk_in = 1

usb_endpoint_descriptor(1):
one_wire_status: 5
bLength=7
```



Figura 4.3: Sistema de Monitoreo y Trasmisión Inalámbrico para la Apicultura.

```
bDescriptorType=5
bEndpointAddress=2
wMaxPacketSize=40
bInterval=0
RT_usb_endpoint_is_bulk_out = 2

usb_endpoint_descriptor(2):
bLength=7
bDescriptorType=5
bEndpointAddress=3
wMaxPacketSize=40
bInterval=0
RT_usb_endpoint_is_bulk_out = 3
```



Figura 4.4: SMTIA con estructura que contiene los sensores para la colmena

```
usb_endpoint_descriptor(3):  
bLength=7  
bDescriptorType=5  
bEndpointAddress=84  
wMaxPacketSize=40  
bInterval=1  
RT_usb_endpoint_is_int_in = 4, Interval = 1  
nr_endpoint=4, in_num=2, out_num=2  
  
NON USB_SPEED_HIGH  
Chip Version ID: VERSION_NORMAL_TSMC_CHIP_88C.  
RF_Type is 3!!  
EEPROM type is E-FUSE  
====> ReadAdapterInfo8192C  
Boot from EFUSE, Autoload OK !  
EEPROMVID = 0x0bda  
EEPROMPID = 0x8176  
EEPROMCustomerID : 0x00  
EEPROMSubCustomerID: 0x00
```

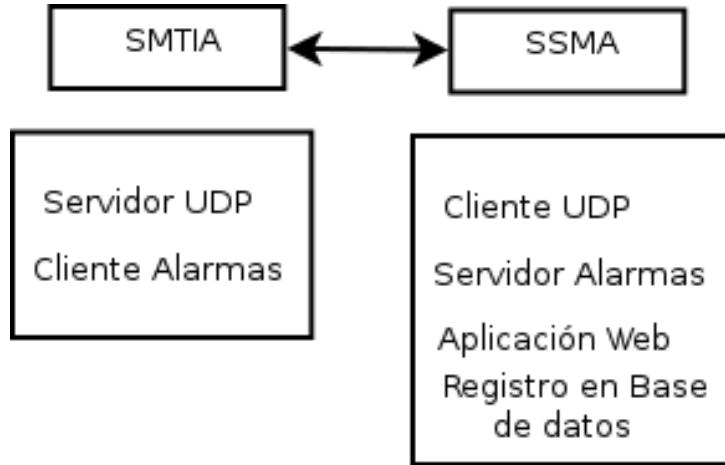


Figura 4.5: Disposición final del SSMA y SMTIA

```

RT_CustomerID: 0x00
    _ReadMACAddress MAC Address from EFUSE = fc:8f:c4:02:e2:ea
    EEPROMRegulatory = 0x0
    _ReadBoardType(0)
    BT Coexistance = disable
    RT_ChannelPlan: 0x00
    _ReadPSSetting...bHWPwrPindetect(0)-bHWPowerdown(0) , \
    bSupportRemoteWakeups(0)
    one_wire_status: 4
    ### PS params=> power_mgnt(1),usbss_enable(0) ####
    ### AntDivCfg(0)
    readAdapterInfo_8192CU(): REPLACEMENT = 1
    <==== ReadAdapterInfo8192C in 2310 ms
    rtw_macaddr_cfg MAC Address = fc:8f:c4:02:e2:ea
    MAC Address from pnetdev->dev_addr= fc:8f:c4:02:e2:ea
    bDriverStopped:1, bSurpriseRemoved:0, bup:0, hw_init_completed:0
    usbcore: registered new interface driver rtl8192cu

```

Ahora se inicia el servicio de autenticación ante el enrutador, mediante la colección de aplicaciones wpa supplicant mediante el comando *wpa supplicant*. Se observa que se carga el firmware que esta embebido en el modulo

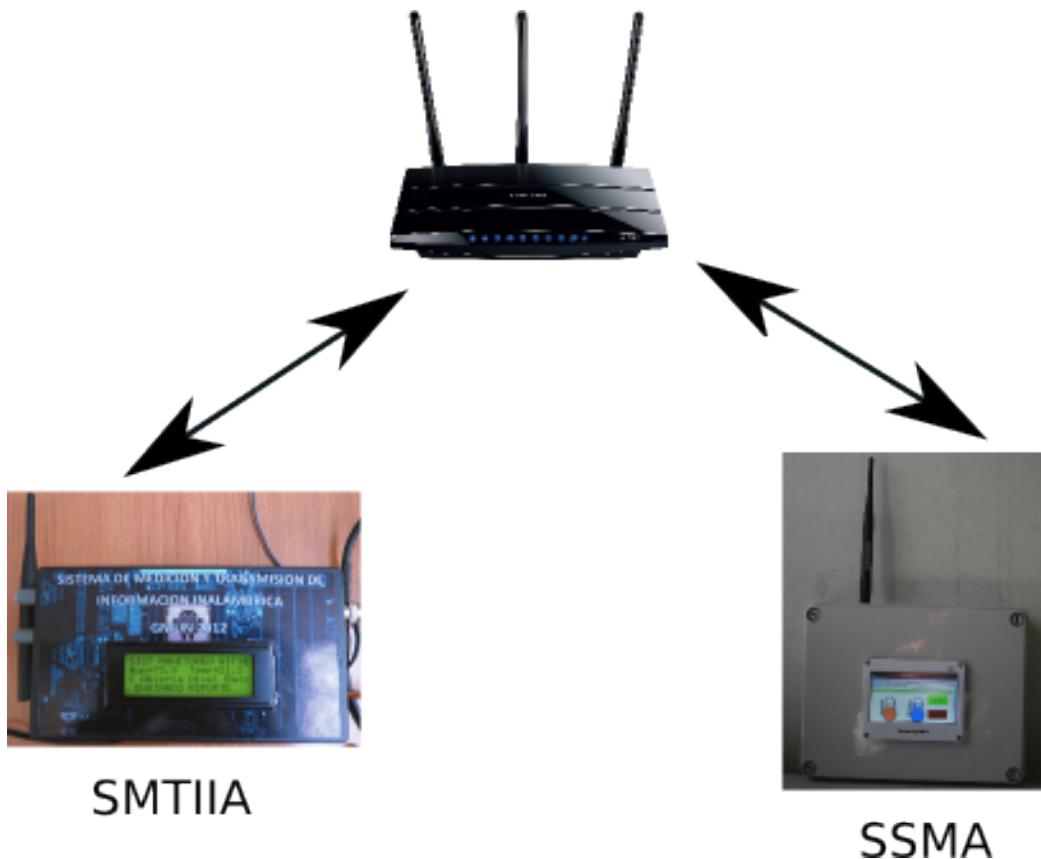


Figura 4.6: Imagen que muestra el montaje para pruebas en laboratorio

controlador y realiza la autenticacion WPA.

```
====> FirmwareDownload91C() fw:Rtl819XFwImageArray_TSMC
FirmwareDownload92C accquire FW from embedded image
fw_ver=v80, fw_subver=0, sig=0x88c0
fw download ok!
Set RF Chip ID to RF_6052 and RF type to 1T1R.
IOL rtl18192c_IOL_exec_cmds_sync complete in 30ms
IOL rtl18192c_IOL_exec_cmds_sync complete in 30ms
IOL rtl18192c_IOL_exec_cmds_sync complete in 70ms
IOL rtl18192c_IOL_exec_cmds_sync complete in 70ms
IOL rtl18192c_IOL_exec_cmds_sync complete in 250ms
```

```
IQK:Start!!!
Path A IQK Success!!
Path A IQK Success!!
IQK: final_candidate is 0
IQK: RegE94=105 RegE9C=13 RegEA4=ff RegEAC=1 RegEB4=0 \
RegEBC=0 RegEC4=0 RegECC=0
    Path A IQ Calibration Success !
pdmpriv->TxPowerTrackControl = 1
MAC Address from REG_MACID = fc:8f:c4:02:e2:ea
rtl8192cu_hal_init in 1790ms
MAC Address = fc:8f:c4:02:e2:ea
-871x_drv - drv_open, bup=1
[rtw_wx_set_pmkid] IW_PMKSA_FLUSH!
set_mode = IW_MODE_INFRA
ioctl[SIOCSIWAP]=>rtw_wx_set_essid
ssid=APIARIO, len=32
Set SSID under fw_state=0x00000008
<=rtw_wx_set_essid, ret 0
: Operation not permitted
[rtw_wx_set_pmkid] IW_PMKSA_FLUSH!
wpa_set_auth_algs, AUTH_ALG_OPEN_SYSTEM
set_mode = IW_MODE_INFRA

wpa_ie(length:24):
0xdd 0x16 0x00 0x50 0xf2 0x01 0x01 0x00
0x00 0x50 0xf2 0x02 0x01 0x00 0x00 0x50
0xf2 0x02 0x01 0x00 0x00 0x50 0xf2 0x02
=>rtw_wx_set_essid
ssid=APIARIO, len=7
Set SSID under fw_state=0x00000808
<=rtw_wx_set_essid, ret 0
survey done event(17)
[by_bssid:0][assoc_ssId:APIARIO][to_roaming:0] new candidate: \
APIARIO(00:1d:68:ec:f5:4d) rssi:-72
rtw_select_and_join_from_scanned_queue: candidate: \
APIARIO(00:1d:68:ec:f5:4d)
link to Broadcom AP
one_wire_status: 4
```

```
link to Broadcom AP
issue_deauth to 00:1d:68:ec:f5:4d
OnAuthClient
network.SupportedRates[0]=82
network.SupportedRates[1]=84
network.SupportedRates[2]=8B
network.SupportedRates[3]=96
network.SupportedRates[4]=24
network.SupportedRates[5]=30
network.SupportedRates[6]=48
network.SupportedRates[7]=6C
network.SupportedRates[8]=0C
network.SupportedRates[9]=12
network.SupportedRates[10]=18
network.SupportedRates[11]=60
bssrate_len = 12
OnAuthClient
OnAssocRsp
report_join_res(1)
rtw_joinbss_update_network
rtw_joinbss_update_stainfo
HW_VAR_BASIC_RATE: BrateCfg(0x15d)
HTOnAssocRsp

~~~~stastakey:unicastkey

~~~~stastakey:groupkey
==> rtw_set_key algorithm(2),keyid(1),key_mask(2)
update raid entry, mask=0x40000fff, arg=0x80
rtl8192c_set_FwJoinBssReport_cmd mstatus(1)
SetFwRsvdPagePkt
Set RSVD page location to Fw.
=>mlmeext_joinbss_event_callback
rtl8192c_dm_RF_Saving(): RF_Save
rtw_set_ps_mode(): Enter 802.11 power save mode...
rtl8192c_dm_RF_Saving(): RF_Normal
rtl8192c_set_FwPwrMode_cmd(): Mode = 1, SmartPS = 2
```

Finalmente, si utilizamos el comando *ifconfig*, se puede observar que la interfaz de red inalámbrica queda correctamente configurada con la dirección IP **192.168.0.7**.

```
wlan0      Link encap:Ethernet HWaddr FC:8F:C4:02:E2:EA
           inet addr:192.168.0.7 Bcast:192.168.0.255 Mask:255.255.255.0
                     UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                     RX packets:315 errors:0 dropped:333 overruns:0 frame:0
                     TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
                     collisions:0 txqueuelen:1000
                     RX bytes:34581 (33.7 KiB) TX bytes:3399 (3.3 KiB)
```

Mediante el establecimiento de esta conexión, se puede ejecutar el cliente y servidor UDP, como también el servidor de alarmas y la aplicación web para el despliegue de la información capturada.

4.2. Conexión GSM

El modem GPRS Huawei E120 que se utilizó para la aplicación, utiliza puertos seriales para su funcionamiento, por lo que no es necesario realizar instalación alguna del módulo controlador. La aplicación escribe directamente sobre el dispositivo *ttyUSB**, para enviar los comandos AT que permiten enviar los mensajes a través de la red GSM.

```
USB Serial support registered for GSM modem (1-port)
option: v0.7.2:USB Driver for GSM modems
option 1-1.2:1.0: GSM modem (1-port) converter detected
usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB0
option 1-1.2:1.1: GSM modem (1-port) converter detected
usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB1
option 1-1.2:1.2: GSM modem (1-port) converter detected
usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB2
option 1-1.2:1.3: GSM modem (1-port) converter detected
usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB3
```

Como se muestra en la salida de terminal anterior, se observa que para poder utilizar el modem GPRS de forma correcta se necesita configurar cuatro puertos seriales. En este punto ya se puede acceder a cada uno de los puertos seriales para configurar la red GSM y enviar los mensajes correspondientes a las alarmas.

4.3. Inicio del kernel

Realizada la compilación del kernel, mediante las fuentes proporcionadas por FriendyARM e instalando adecuadamente la imagen ejecutable en la memoria NAND Flash

se obtiene un salida en consola de la carga que hace el SBC en la memoria RAM de la misma, donde se observa la configuración de algunos dispositivos.

OK

```
U-Boot 1.1.6 (Jul 21 2011 - 12:49:00) for FriendlyARM MINI6410
```

```
CPU:      S3C6410@532MHz
          Fclk = 532MHz, Hclk = 133MHz, Pclk = 66MHz, Serial = CLKUART (SYNC Mode)
Board:    MINI6410
DRAM:    256 MB
Flash:   0 kB
NAND:    2048 MB
In:      serial
Out:     serial
Err:     serial
MAC:    08:90:90:90:90:90
Hit any key to stop autoboot:  0

NAND read: device 0 offset 0x400000, size 0x5000000
.....Boot with zImage

Starting kernel ...

Uncompressing Linux... done, booting the kernel.
Initializing cgroup subsys cpu
Linux version 2.6.38-FriendlyARM (kapak@puputi) (gcc version 4.5.1 (ctng-1.8.1-FA) ) #7 \
PREEMPT Wed Oct 24 18:43:26 COT 22
CPU: ARMv6-compatible processor [410fb766] revision 6 (ARMv7), cr=00c5387f
CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
Machine: MINI6410
.....
```

Como se observa anteriormente, el kernel inicia correctamente, configurando el procesador y el sistema de memoria NAND. No se incluye la depuración completa, ya que incluye demasiada información redundante e innecesaria para los propósitos de este texto.

4.4. Base de datos SQLite

Para almacenar la información obtenida de los sensores y los estados de los sensores de nivel y apertura, obteniendo la información mediante la aplicación cliente UDP, se

```
[root@buildroot database]# sqlite3 colmena.sqlite3 "select * from colmena"
1|API01|1354654526.85378|29.08000|44.32117|NOR|CL0
2|API01|1354654560.97734|29.14999|43.87949|NOR|CL0
3|API01|1354654595.06208|29.08000|44.32117|NOR|CL0
4|API01|1354654630.66778|29.13000|43.71422|NOR|CL0
5|API01|1354654664.64867|29.08000|44.07622|NOR|CL0
6|API01|1354654700.06104|28.86000|44.50215|NOR|CL0
7|API01|1354654735.57461|28.78000|44.53453|NOR|CL0
8|API01|1354654769.55606|28.57000|44.92063|NOR|CL0
9|API01|1354654804.96796|28.50999|45.48631|NOR|CL0
10|API01|1354654840.5861|28.66000|43.70609|NOR|CL0
11|API01|1354654874.6668|28.62000|44.31358|NOR|CL0
12|API01|1354654911.51241|28.66999|44.44127|NOR|CL0
13|API01|1354654945.49294|28.45000|44.70372|NOR|CL0
14|API01|1354654979.47086|28.36000|45.51105|NOR|CL0
15|API01|1354655015.7032|28.17000|45.69493|NOR|CL0
16|API01|1354655049.68272|28.28000|48.00143|NOR|CL0
17|API01|1354655085.09625|28.52000|47.41254|NOR|CL0
18|API01|1354655121.25572|28.98999|45.98831|NOR|CL0
19|API01|1354655155.92037|29.49000|43.05819|NOR|CL0
20|API01|1354655189.90124|29.33000|45.24704|NOR|CL0
21|API01|1354655322.67096|29.24000|45.15559|NOR|CL0
```

Figura 4.7: Salida del comando *sqlite3*, consultando la tabla con información almacenada

obtienen tramas de datos que se van almacenando en una base de datos local embebida, implementada mediante SQLite 3.

En la figura 4.7 se observa la salida en la terminal generada por una consulta a la base de datos mediante el comando *sqlite3*, solicitando toda la información de la tabla *colmena*. Como se observa, cada registro tiene la misma forma de la respuesta del servidor UDP, cuando se envia la sentencia “QUERY”.

4.5. Servidor UDP

Una vez implementado el servidor UDP en la tarjeta que controla el SMTIA, basada en un procesador ARM, se realizaron pruebas para verificar el correcto funcionamiento de la aplicación, observando la depuración generada desde el código fuente y que es posible verificar gracias a los mensajes que se muestran en la terminal.

En la figura 4.8 se observa la información que es organizada y enviada al cliente UDP que se encuentra en el SSMA. La aplicación se utilizará para el sistema de alarmas y la interfaz hombre maquina.

Se puede observar de izquierda a derecha: la marca de tiempo, valor de la temperatura en ese instante, valor de la humedad relativa en ese instante, estado del nivel del tanque y

```
Consultando sensor
+ 1354679276.9 20.94999 76.72259 NOR CL0
```

Figura 4.8: Mensajes de la terminal que muestran la información enviada por la aplicación servidor UDP

estado de la tapa de la colmena. La aplicación envio toda la secuencia de datagramas cada vez que un programa cliente solicitaba esta información mediante la sentencia “QUERY”.

4.6. Cliente UDP

Mediante esta aplicación, desarrollada en Python, se envia un datagrama con la palabra “QUERY”, al cual el servidor responde enviando informacion de los sensores instaladas. Esta aplicación se utilizo en el sistema que controla la interfaz hombre maquina y el envio de mensajes en caso de alarmas.

Es posible desarrollar aplicaciones cliente en cualquier lenguaje de programación, flexibilizando el campo de aplicación del sistema de supervisión y monitoreo.

Se utiliza la dirección IP **192.168.0.7**, solicitando una conexión por el puerto **4001**.

4.7. Aplicación Web

Se realiza la ejecución en consola de la aplicación realizada en Python, mediante las librerías de Webpy, observando la salida que se programo desde el código fuente de la aplicación.

4.7.1. Webpy

En la gráfica 4.9 se observa la salida de consola para la aplicación, creando un servidor web en el puerto *8080*. Seguido a esto se observa la salida del cliente UDP, común para variar aplicaciones y finalmente el renderizado de la página web que contiene los indicadores para temperatura, humedad y estados de los sensores de nivel y apertura.

Se observa los elementos que utiliza el subsistema de renderizado de la página web, donde utiliza imágenes y código de javascript para dibujar los indicadores mencionados anteriormente.

```

-----
Iniciando aplicacion Web SSMA
-----
Consultando sensor
Enviando QUERY
+ 1358821760.98 19.62000 71.99505 NOR CL0
192.168.0.3:35393 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /colmena/" - 200 0K
192.168.0.3:35395 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.common.resizing.js" - 200
192.168.0.3:35395 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.common.context.js" - 200
192.168.0.3:35393 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.common.core.js" - 200
192.168.0.3:35397 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.common.key.js" - 200
192.168.0.3:35396 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.common.effects.js" - 200
192.168.0.3:35394 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.common.annotate.js" - 200
192.168.0.3:35395 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.common.tooltips.js" - 200
192.168.0.3:35398 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.common.zoom.js" - 200
192.168.0.3:35394 - - [21/Jan/2013 19:29:26] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.bar.js" - 200
192.168.0.3:35393 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.bipolar.js" - 200
192.168.0.3:35395 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.cornergauge.js" - 200
192.168.0.3:35397 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.fuel.js" - 200
192.168.0.3:35398 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.funnel.js" - 200
192.168.0.3:35396 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.gauge.js" - 200 192.168.0.3:3539
393 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.gantt.js" - 200

192.168.0.3:35397 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.hprogress.js" - 200
192.168.0.3:35395 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.hbar.js" - 200
192.168.0.3:35398 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/mystyle.css" - 304 Not Modified
192.168.0.3:35394 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.led.js" - 200
192.168.0.3:35393 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.line.js" - 200
192.168.0.3:35398 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.meter.js" - 200
192.168.0.3:35397 - - [21/Jan/2013 19:29:27] "HTTP/1.1 GET /static/javascript/rgraph/RGraph.pie.js" - 200 192.168.0.3:3539

```

Figura 4.9: Mensajes de la terminal que muestran la información enviada por la aplicación servidor UDP

4.7.2. HTML5

Gracias al conjunto de herramientas Rgraph (<http://www.rgraph.net/>), que consiste en código fuente Javascript y gracias a las etiquetas Canvas de la implementación HTML5, se visualizan los indicadores cada vez que el navegador cliente requiere nuevamente la página.

Mediante estas nuevas características en la tecnologías Web, es posible hacer un interfaz amigable para el usuario final, sin necesidad de sacrificar una presentación de buen diseño, en vista de lograr una implementación técnica del sistema en cuestión.

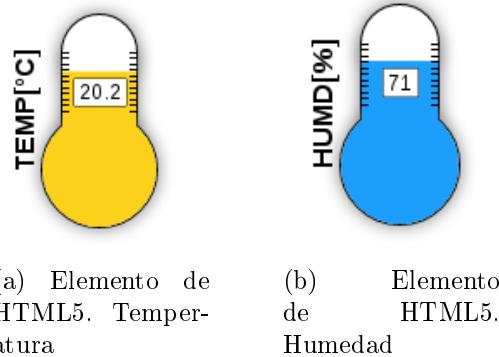
La imagen ?? muestra los elementos dibujados mediante la etiqueta canvas, que es modificada por los script de Rgraph, teniendo como argumentos los valores de humedad y temperatura respectivamente.

4.8. Interfaz Hombre Maquina

En las figuras, se muestran las diferentes interfaces que permiten que la aplicación web cumpla el objetivo propuesto adecuadamente.

4.8.1. Apario

Figura 4.10a Interfaz de bienvenida al SSMA. Se puede implementar otro tipo de información.



4.8.2. Colmena

Figura 4.10b. Interfaz donde se muestran las variables observadas en al colmena. Para la temperatura se observa el valor de esta dimension con un decimal de precisión. Para la humedad se muestra como proporción, el valor relativo de esta variable. Con colores rojo y verde, indicando estado de alerta y normal respectivamente, se muestra una representación de los estados de los sensores de nive y apertura.

Esta pagina se refresca cada 10 segundos a traves de una funcion de javascript que solicita a la aplicacion web nueva informacion.

4.8.3. Config

Figura 4.10c. Mediante este formulario se puede cambiar la informacion para el envio de mensajes de texto y SMS en caso de alarmas, especificando el mensaje para cada uno de los eventos.

4.8.4. Utilities

Figura 4.10d. En este parte de la aplicacion se realiza la descarga de la base de datos en un dispositivo extraible en el SSMA.

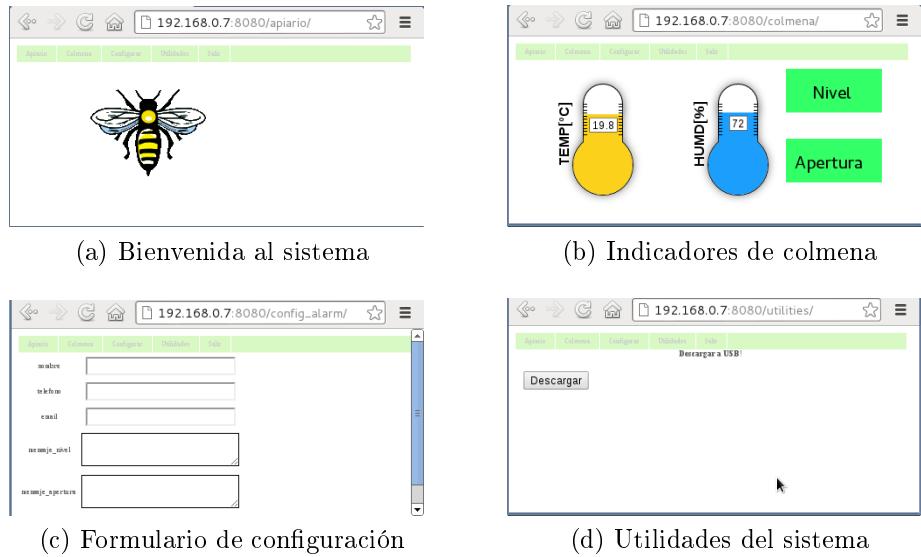


Figura 4.10: Interfaz Hombre Maquina vista desde un navegador Web externo al SSMA

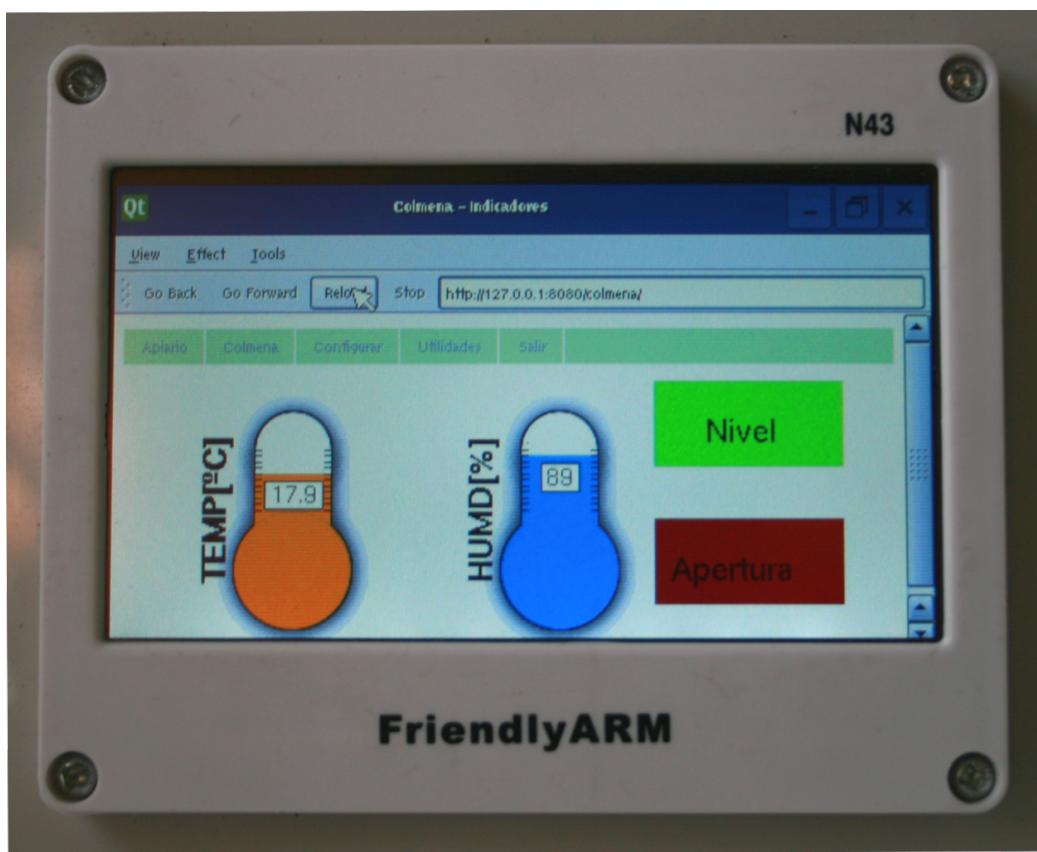


Figura 4.11: Interfaz Hombre Maquina vista en la pantalla LCD del SSMA

Capítulo 5

Conclusiones

5.1. Conclusiones

- El diseño e implementación de un sistema de supervisión y monitoreo remoto, que trabaje de forma autónoma y confiable, es posible ya que no requiere un diseño muy complejo y la oferta de sistemas embebidos o SBC en el mercado es amplia.
- El sistema provee una interfaz que permite visualizar de forma gráfica y conceptual, las variables que mide el sistema de monitoreo y transmisión de datos inalámbrico SMTIA, previamente instalado en el apíario.
- Debido a la arquitectura del sistema, es posible la futura implementación de otros módulos que permitan monitorear otro tipo de variables y hacerlas accesibles en sitio o de forma remota.
- Mediante estos sistemas de supervisión y monitoreo, se facilitan algunas tareas comunes en la investigación y utilización de los productos de las abejas.
- Los sistemas embebidos permiten crear mecanismos para acceder a información de procesos que de formas tradicionales no sería posible pues se afectaría el entorno o los procesos típicos, en este caso de las colmenas.
- Gracias a la difusión del software libre, se pueden crear plantillas que permiten aplicar esta misma solución a otros procesos.

Bibliografía

- [1] B Boydell A Whelan. The impact of precision agriculture, 1997.
- [2] About.com. dhclient. http://linux.about.com/library/cmd/blcmdl8_dhclient.htm, 2012.
- [3] Exequio Leyva Perez Alexander Garcia Fernandez, Joaquin Pina Amargós. Ingenieria industrial: Estado del arte de las redes inalámbricas, 2007.
- [4] Erik Andersen. Busybox: The swiss army knife of embedded linux. <http://www.busybox.net/about.html>, 2012.
- [5] Mohamad Redzuan Bin Ahmad Azhar Fakharuddin. *Design of embedded automated monitoring system; an intelligent application to reduce peak load demand*. University Malaysia Pahang, Makmur, Malaysia, 2010.
- [6] Apicultores Colombianos. Que es apicultura: descripción, 2012.
- [7] Yuhang Wang Guang Dong, Wei He. *Database Design on Embedded Home Gateway and Web Server Implementation*. Changchun University of Science and Technology, Department of Control Science and Engineering, Changchun, China, 2010.
- [8] Jeisson Alejandro Gutiérrez. *Diseño y construcción de uns sistema de medición y transmisión de información inalámbrico para la apicultura*. Universidad Nacional de Colombia, Bogotá, Colombia, 2012.
- [9] Richard Heyes. Rgraph - free html5/javascript charts and graphs. <http://www.rgraph.net/index.html>, 2012.
- [10] Hong-Taek Ju. *Embedded Web Server Architecture for Web-based Element and Network Management*. Pohang University of Science and Technology, Corea, 2001.
- [11] Christopher Larson. Tslib: About. <http://tslib.berlios.de/>, 2012.
- [12] Linying Jiang Kun Yue Liu Yang, Heming Pang. *The Research of Embedded Linux and SQLite Database Application in the Intelligent Monitoring System*. Software College Northeastern University, Shenyang, China, 2010.

- [13] pyhumod. Pyhumod. <http://code.google.com/p/pyhumod/>, 2012.
- [14] PySerial. Pyserial: Overview and features. <http://pyserial.sourceforge.net/pyserial.html#overview>, 2012.
- [15] Wang Yu Sun Jianhua, Yang Xu. *Solar Wireless Intelligent Online Hydrological Monitoring System*. College of Arts and Sciences of Beijing Union University, Beijing, China, Beijing, China, 2010.
- [16] Wikipedia. Gnu wget. http://es.wikipedia.org/wiki/GNU_Wget, 2012.
- [17] Wikipedia. ifconfig. <http://es.wikipedia.org/wiki/Ifconfig>, 2012.
- [18] Wikipedia. Javascript. <http://es.wikipedia.org/wiki/JavaScript>, 2012.
- [19] Wikipedia. Ping. <http://es.wikipedia.org/wiki/Ping>, 2012.
- [20] Wikipedia. Ssh. http://es.wikipedia.org/wiki/Secure_Shell, 2012.
- [21] Wikipedia. wpa_supplicant. http://en.wikipedia.org/wiki/Wpa_supplicant, 2012.
- [22] JianWei Li XiaoLi Wang. *CDMA-based Application of Wireless Intelligent Monitoring Systems*. JiLin Institute of Architecture And Civil Engineering, Changchun, China, 2010.
- [23] Jing Li. Yong Xu. *Remote Monitoring Systems Based on Embedded Database*. Guilin University of electronic technology, Guilin, China, 2009.
- [24] Ruchun Cui Zhen Zhu. *Remote Intelligent Monitoring System Based on Embedded Internet Technology*. Foshan University, Guangdong, China, 2007.

Lista de acrónimos