Ingeniería Informática

Proyecto Final de Carrera

"Plataforma de telecontrol sobre Linux embebido en arquitecturas AVR32"

Especificación de requisitos del sistema
V2.0

Tutor del proyecto:
Moisés Friginal Santos Olmo
mfriginal@usj.es

Trabajo elaborado por:
José Antonio Esparza Isasa
alu.00033@usj.es

4º Ingeniería Informática

29/Septiembre/2009

UNIVERSIDAD
SAN JORGE

Control de versiones

Fecha	Versión	Descripción	Autor
23 noviembre 2008	1.0	Creación del documento. Se añaden requisitos generales del sistema y requisitos referentes a la arquitectura de comunicaciones.	José Antonio Esparza
29 septiembre 2009	2.3	Actualización del documento conforme a los requisitos registrados en la hoja de control de requisitos software	José Antonio Esparza

Tabla de contenido

1. Introducción 4
1.1. Propósito 4
1.2. Alcance 4
1.3. Definiciones, Acrónimos y abreviaturas 4
1.4. Referencias 4
1.5. Visión general 5
2. Descripción general 6
3. Requisitos específicos 7
3.1. Requisitos funcionales 7
3.1.1. Gestión de concurrencia en el acceso a recursos 7
3.1.2. Gestión de mapeados de memoria 8
3.1.3. Comunicación I2C 9
3.1.4. Comunicación serie 10
3.1.5. Comunicación vía TCP 13
3.2. Mantenimiento 16
3.3. Restricciones de diseño 16
3.3.1. Librería C del sistema embebido 16
3.3.2. Cumplir con las tramas ya en uso para el servidor socket 16
3.4. Documentación disponible y ayuda 16
3.5. Componentes adquiridos 16
3.6. Interfaces 16
3.6.1. Interfaces de usuario 16
3.6.2. Interfaces hardware 17
3.6.3. Interfaces software 17
3.6.4. Interfaces de comunicación 17
3.7. Estándares aplicables 17
3.7.1. Comunicaciones 17
3.7.2. Documentación 17
4. Información adicional 18
4.1. Libros: 18
4.2. Especificaciones: 18
4.3. Artículos: 18
4.4. Trabajos académicos: 18
4.5. Páginas web: 18
4.6. Tutoriales y documentación específica 19

1. Introducción

1.1. Propósito

El propósito de este documento es recopilar los requisitos existentes en el proyecto "Plataforma de telecontrol sobre Linux embebido en arquitecturas AVR32".

Se presentarán tanto requisitos funcionales como no funcionales, además de otras restricciones de diseño y factores para comprender los requerimientos del sistema a diseñar en este proyecto.

1.2. Alcance

El proyecto "Plataforma de telecontrol sobre Linux embebido en arquitecturas AVR32" presenta un sistema de telecontrol, con conectividad a internet, capaz de gestionar diversos dispositivos electrónicos a través de buses muy extendidos, como I2C o serie. Además de una infraestructura básica para las comunicaciones, el proyecto comprende la implementación de un servidor socket que cumpla con los formatos utilizados por MACRAUT para sus aplicaciones de telecontrol.

Para una información más detallada sobre el propósito del proyecto se puede consultar el documento "Visión del proyecto".

1.3. Definiciones, Acrónimos y abreviaturas

AVR32: Procesador de 32 bits de la empresa Atmel en el que está basada la solución a desarrollar en este proyecto.

HAL: Hardware Abstraction Layer. Acrónimo utilizado para referirse a la capa de abstracción que permite a los programadores interactuar con el hardware sin la necesidad de conocer cuestiones complejas de bajo nivel.

I+D + i: Investigación, Desarrollo e innovación. Departamento responsable del desarrollo del presente proyecto.

LINX: subsistema de PE-COM responsable de tomar datos de sistemas de menor peso en la red y de comunicarlos a las estaciones centrales.

PE-COM: Power Electronics Communication. Nombre que le da MACRAUT al sistema de telecontrol desarrollado por la misma empresa.

PLC: Programmable Logic Controller. Dispositivo equivalente a un autómata programable, permite almacenar una lógica que es responsable del control de un proceso industrial.

SCADA: Supervisory Control and Data Adquisition. Acrónimo utilizado para referirse a los sistemas encargados del control de un proceso industrial y la toma de datos relevantes del mismo.

1.4. Referencias

Referencias a otros documentos utilizados en el proyecto. Para referencias a otros documentos externos al proyecto consultar el punto 4 referente a la información adicional.

- Esparza Isasa, José Antonio. "Plataforma de telecontrol sobre Linux embebido en arquitecturas AVR32" Visión del proyecto. Versión 2.0

- Esparza Isasa, José Antonio. Memoria de prácticas en MACRAUT ingenieros. Verano 2008
- Esparza Isasa, José Antonio. Hoja Excel de control de requisitos.
- MACRAUT Ingenieros, Protocolo de comunicaciones Nexus, borrador 1 26/04/07

1.5. Visión general

En las siguientes páginas se describe a grandes rasgos el proyecto y se analizan pormenorizadamente los requisitos específicos del sistema, que se clasifican en funcionales y no funcionales.

Dentro del conjunto de requisitos no funcionales se profundizará especialmente en la usabilidad: qué facilidades ofrece el sistema para trabajar con él, como se integra con los equipos ya existentes. En el apartado de mantenimiento se analizará la complejidad del mismo y las posibilidades de gestión remota. Las restricciones de diseño reflejarán el material disponible y su precio.

Documentación disponible y ayuda.

Las posibilidades de expansión del sistema y de la red trat: el producto a desarrollar no es un sistema aislado, ¿cómo se integra en la red? ¿Es escalable? Los interfaces de comunicación: ¿Cómo se puede comunicar con otros equipos?

También se tratan otros temas de menor relevancia en este caso, como estándares y licencias.

2. Descripción general

El problema a solucionar con este proyecto consiste en la modernización de los sistemas LINX de la empresa MACRAUT utilizados en sus dispositivos de telecontrol. La modernización de estos equipos afectará al personal de mantenimiento y a los usuarios finales. Las principales consecuencias de la adopción de este sistema serán una mayor estabilidad de la red de telecontrol y mejorar la escalabilidad. Una solución de éxito sería la creación de una plataforma de telecontrol sobre un sistema Linux embebido en los micros de la empresa Atmel AVR32. Esta solución destaca por las siguientes razones:

- Relación características/precio. Se utiliza un microprocesador económico y potente, que corre el sistema operativo Linux. Linux esta bajo licencia GNU/GPL y por lo tanto se puede utilizar gratuitamente.

- Posibilidad de crear la plataforma hardware a medida, optimizando el consumo. Gracias a esta posibilidad se puede obtener una solución altamente competitiva a la ofrecida por un autómata PLC. Existe suficiente documentación y dispositivos electrónicos económicos que permitirán diseñar una plataforma específica y que consiga bajar el consumo eléctrico y el precio total del equipo a niveles muy inferiores a los de un autómata.

- Amplias posibilidades de comunicación. El sistema es capaz de utilizar múltiples interfaces de comunicación, entre los que destacan: I2C, SPI, Ethernet, USB, Serie, etc.

- Utilización de un sistema estable como Linux, ampliamente probado y depurado. El hecho de que el código fuente de este sistema operativo y de multitud de software desarrollado para el mismo esté disponible en internet ha posibilitado a gran cantidad de profesionales y usuarios su testeo y depuración.

- Posibilidades de expansión del sistema. Entre otras características, figuran: buses de alto rendimiento, interfaz LCD, interfaz para cámaras CMOS, soporte para tarjetas de memoria SD, Compact Flash, Smart Media entre otras, etc.

MACRAUT ingenieros, que cuenta con redes de telecontrol a lo largo del territorio nacional y realiza numerosos despliegues del sistema PE-COM para la telegestión, se plantea un nuevo sistema de telecontrol que realice las tareas que en estos momentos realiza el equipo LINX, con mayor estabilidad, buena relación calidad/precio, utilizando un sistema software de gran calidad y sobre arquitecturas AVR32 que permitirán la creación hardware de una plataforma específica para el sistema.

3. Requisitos específicos

3.1. Requisitos funcionales

Los requisitos funcionales presentados en este punto quedan reflejados para su control en la hoja Excel asociada el proyecto denominada "control de requisitos".

3.1.1. Gestión de concurrencia en el acceso a recursos

Identificador	1
Nombre	El sistema debe ser capaz de gestionar adecuadamente los bloqueos sobre los recursos mediante la libraría existente
Descripción	Al tratarse de un sistema multitarea con múltiples recursos un control de bloqueos adecuado resulta fundamental. Para gestionar los bloqueos se utilizarán semáforos IPV5.
Comentarios	Librería ya desarrollada, necesaria una revisión de los casos problemáticos. Para mayor detalle consultar la documentación creada en el documento "Memoria de prácticas".

Identificador	1.1
Nombre	Gestión de bloqueos de lectura
Descripción	En el caso de que se vayan a realizar operaciones de lectura sobre un recurso, será necesario obtener un bloqueo de de este tipo para que los datos contenidos no sean alterados mientras se realiza la operación.
Comentarios	Librería de gestión de concurrencia ya desarrollada, pendiente de tests más exhaustivos. Para mayor detalle consultar la documentación creada en el documento "Memoria de prácticas".

Identificador	1.2
Nombre	Gestión de bloqueos de escritura
Descripción	En el caso de realizar operaciones de escritura, será necesario obtener bloqueos de escritura, para evitar que otro proceso intente leer datos que están siendo modificados.
Comentarios	Librería de gestión de concurrencia ya desarrollada, pendiente de tests más exhaustivos. Para mayor detalle consultar la documentación creada en el documento "Memoria de prácticas".

Identificador	1.3
Nombre	Revisión de las interfaces ofrecidas
Descripción	Se ofrecen una serie de interfaces que pueden ser utilizadas desde otros programas para hacer uso de los bloqueos. Será necesario revisar cómo se han definido y la manera de utilizarlas.
Comentarios	

Identificador	1.4
Nombre	Revisión de la documentación generada
Descripción	La librería tiene que quedar correctamente documentada de cara a facilitar desarrollos futuros.
Comentarios	Revisar la documentación que se adjuntó en la memoria de prácticas y completarla en el caso de que proceda

3.1.2. Gestión de mapeados de memoria

Identificador	2.0
Nombre	El sistema debe ser capaz de gestionar adecuadamente los mapeados de memoria mediante la librería ya existente
Descripción	El sistema trabajará con datos contenidos en ficheros que serán mapeados en la memoria principal del sistema cada vez que se vayan a utilizar. Es necesario llevar un control de la carga y actualización de estos ficheros binarios desde el sistema empotrado. Será necesario que el acceso a los mapeados esté regulado por las rutinas de control de concurrencia previamente desarrolladas.
Comentarios	Librería desarrollada durante prácticas , necesaria una revisión de los casos problemáticos.

Identificador	2.1
Nombre	Carga de mapeados
Descripción	El sistema deberá ser capaz de cargar en memoria los mapeados de las estaciones almacenados en el mismo. Estos mapeados estarán almacenados en ficheros binarios.
Comentarios	

Identificador	2.2
Nombre	Borrado de mapeados de la memoria
Descripción	El sistema deberá ser capaz de borrar mapeados que hayan sido cargados en memoria previamente
Comentarios	

Identificador	2.3
Nombre	Revisión de las interfaces ofrecidas
Descripción	Se ofrecen una serie de interfaces que pueden ser utilizadas desde otros programas para hacer uso de los mapeados de memoria. Será necesario revisar cómo se han definido y la manera de utilizarlas.
Comentarios	

Identificador	2.4
Nombre	Revisión del guardado de mapeados de memoria en ficheros binarios
Descripción	El sistema deberá ser capaz de poder guardar el contenido del mapeado de memoria de una estación en un fichero binario.
Comentarios	

Identificador	2.5
Nombre	Revisión de la documentación generada
Descripción	La librería tiene que quedar correctamente documentada de cara a facilitar desarrollos futuros.
Comentarios	Revisar la documentación que se adjuntó en la memoria de prácticas y completarla en el caso de que proceda

3.1.3. Comunicación I2C

Identificador	3.1
Nombre	El sistema debe ser capaz de poder mantener una comunicación mediante el bus i2c
Descripción	El sistema comunicará con las tarjetas de entrada salida HIO de manera local mediante este bus.
Comentarios	

Identificador	3.1.1
Nombre	Recompilación del kernel para soporte al bus I2C
Descripción	Es necesario llevar a cabo una compilación del kernel para integar los módulos relativos al funcionamiento del bus i2c.
Comentarios	Por defecto la versión del kernel que se suministra con la placa no viene con los módulos i2c disponibles.

Identificador	3.2
Nombre	El sistema debe poder escribir los datos relativos a un mapeado en una placa esclavo
Descripción	La placa secundaria debe actuar como esclava escritora mientras que la maestra (placa STK100) como lectora. El objetivo será escribir los datos de un mapeado en un dispositivo esclavo.
Comentarios	

Identificador	3.2.1
Nombre	Preparación de una placa secundaria para probar la comunicación
Descripción	Será necesario contar con un equipo que realice funciones sencillas como respuesta de datos ante una petición o escritura de datos recibidos.
Comentarios	

Identificador	3.2.2
Nombre	Prueba de comunicación simple entre STK1000 maestro y placa secundaria esclava. Escritura de datos en el esclavo
Descripción	El equipo secundario debe recibir datos a través de I2C y guardarlos en memoria. Para ello se empezará a transmitir datos sencillos como enteros o cadenas de caracteres de pequeño tamaño
Comentarios	

Identificador	3.2.3
Nombre	Transmisión de un mapeado completo a la placa esclavo
Descripción	Una vez controlada la transmisión de datos será necesario implementar una función que transmita a través del bus I2C un mapeado completa al equipo secundario.
Comentarios	

Identificador	3.3
Nombre	El sistema debe poder recibir los datos relativos a un mapeado en una placa esclavo
Descripción	La placa secundaria debe actuar como esclava lectora, mientras que la maestra (placa STK100) como escritora. El objetivo es recibir los datos de un mapeado existentes en un dispositivo esclavo.

Comentarios	

Identificador	3.3.1
Nombre	Preparación de una placa secundaria para probar la comunicación
Descripción	Será necesario contar con un equipo que realice funciones sencillas como respuesta de datos ante una petición o escritura de datos recibidos
Comentarios	

Identificador	3.3.2
Nombre	Prueba de comunicación simple entre STK1000 maestro y placa secundaria esclava. Lectura de datos desde el maestro
Descripción	El equipo secundario debe enviar datos a la placa principal (STK1000) a través de i2c. La placa principal debe ser capaz de recibir correctamente estos datos.
Comentarios	

Identificador	3.3.3
Nombre	Recepción de un mapeado completo desde la placa maestro
Descripción	El equipo secundario debe enviar el mapeado de una estación completo a la placa principal (STK1000) a través de i2c. La placa principal debe ser capaz de guardar el mapeado completo en memoria.
Comentarios	

Identificador	3.4
Nombre	La comunicación vía I2C tiene que poder integrarse con las tareas de gestión de mapeados y control de bloqueos
Descripción	
Comentarios	

Identificador	3.4.1
Nombre	Prueba de integración con las librerías de gestión de mapeados
Descripción	Los mapeados serán accedidos mediante la librería desarrollada para la gestión de lo mismos.
Comentarios	Ver requisito 2.0

Identificador	3.4.2
Nombre	Prueba de integración con las librerías de control de bloqueos
Descripción	Antes de que los datos contenidos en los mapeados sean actualizados o accedidos para su lectura, será necesario obtener un bloqueo que regule estas transacciones, con el objeto de evitar problemas de concurrencia.
Comentarios	Ver requisito 1.0

3.1.4. Comunicación serie

Identificador	4.1
Nombre	El sistema debe ser capaz de poder mantener una comunicación mediate el puerto serie
Descripción	El sistema se conectará a un modem radio transparente a través del cual se mantendra un enlace punto a punto con una seguda estación. Este modem radio se conecta a través de RS232, por lo tanto es necesario acceder a este puerto desde un programa en C.
Comentarios	Al permitir la comunicación con un puerto serie el abanico de dispositivos que puede manejar el sistema en el futuro es bastante amplio. Por ejemplo: modems GSM, GPRS …

Identificador	4.1.1
Nombre	Revisión de la configuración actual del puerto serie en la placa NGW100
Descripción	Será necesario conocer la configuración por defecto que la placa trae de fábrica y modificarla en el caso de que proceda para que soporte la comunicación por el puerto serie.
Comentarios	

Identificador	4.1.2
Nombre	Revisión de la configuración actual del puerto serie en la placa STK1000
Descripción	Será necesario conocer la configuración por defecto que la placa trae de fábrica y modificarla en el caso de que proceda para que soporte la comunicación por el puerto serie.
Comentarios	

Identificador	4.1.3
Nombre	Prueba de escritura en el puerto serie
Descripción	El sistema deberá ser capaz de escribir datos simples en el puerto serie.
Comentarios	

Identificador	4.1.4
Nombre	Prueba de lectura básica en el puerto serie
Descripción	El sistema deberá ser capaz de leer datos simples recibidos a través del puerto serie.
Comentarios	

Identificador	4.2
Nombre	Se utilizará el protocolo MODBUS para la transmisión de datos
Descripción	El sistema actuará como maestro MODBUS para comunicar con diversos esclavos
Comentarios	

Identificador	4.2.1
Nombre	Creación de peticiones de escritura
Descripción	El sistema deberá ser capaz de crear una trama que contenga una petición de escritura, conteniendo una dirección de destino de la petición además de los datos a escribir
Comentarios	

Identificador	4.2.2
Nombre	Recepción de respuesta a peticiones de escritura
Descripción	El sistema deberá recibir la trama de respuesta por parte del esclavo a una petición de escritura
Comentarios	

Identificador	4.2.3
Nombre	Creación de peticiones de lectura
Descripción	El sistema deberá ser capaz de crear una trama que contenga una petición

	de lectura. Contendrá una dirección de destino de la petición y los parámetros necesarios para identificar los registros a recuperar
Comentarios	

Identificador	4.2.4
Nombre	Recepción de respuesta a peticiones de lectura
Descripción	El sistema recibirá los datos contenidos en los registros indicados en la petición.
Comentarios	

Identificador	4.2.5
Nombre	Verificación de la integridad de los paquetes recibidos
Descripción	Será necesario revisar la integridad de los datos recibidos para detectar posibles errores en la transmisión.
Comentarios	

Identificador	4.3
Nombre	Petición de lectura de un mapeado completo
Descripción	El sistema deberá crear una serie de peticiones modbus de lectura para la recepción de los datos de un mapeado
Comentarios	

Identificador	4.4
Nombre	Petición de escritura de datos sobre un mapeado de memoria
Descripción	El sistema deberá crear una serie de peticiones modbus de escritura para la escritura de los datos en un mapeado
Comentarios	

Identificador	4.5
Nombre	La comunicación vía RS232 tiene que poder integrarse con las tareas de gestión de mapeados y control de bloqueos

Identificador	4.5.1
Nombre	Prueba de integración con las librerías de gestión de mapeados
Descripción	Los mapeados a los que acceden las rutinas serán accedidos mediante la librería desarrollada para la gestión de lo mismos.
Comentarios	Ver requisito 2.0

Identificador	4.5.2
Nombre	Prueba de integración con las librerías de control de bloqueos
Descripción	Antes de que los datos contenidos en los mapeados sean actualizados o accedidos para su lectura, será necesario obtener un bloqueo que regule estas transacciones, con el objeto de evitar problemas de concurrencia.
Comentarios	Ver requisito 1.0

3.1.5. Comunicación vía TCP

Identificador	5.1
Nombre	El sistema deberá ser capaz de mantener una comunicación a través del una conexión socket

Identificador	5.1.1
Nombre	Implementación de un servidor socket con funcionalidades básicas bajo linux utilizando C
Descripción	La creación de un servidor socket sencillo para el intercambio de datos simples permitirá revisar la implementación de este tipo de servicios y ayudará en el diseño y programación del servidor final.
Comentarios	Revisar Beej's Guide to Network Programming. http://beej.us/guide/bgnet/ [link válido Septiembre 2009]

Identificador	5.1.2
Nombre	Implementación de un cliente socket con funcionalidades básicas bajo linux utilizando C
Descripción	La creación de un cliente socket sencillo para el intercambio de datos simples permitirá revisar la implementación de este tipo de aplicaciones y ayudará en el diseño y programación del servidor final.
Comentarios	Revisar Beej's Guide to Network Programming. http://beej.us/guide/bgnet/ [link válido Septiembre 2009]

Identificador	5.2
Nombre	El servidor socket deberá integrarse con el cliente PE-VIEW
Descripción	Será necesario cumplir los protocolos previamente establecidos para este tipo de comunicaciones.
Comentarios	Consultar Espeficicación del protocolo en el documento interno de MACRAUT: "Protocolo de comunicaciones NEXUS Borrador 1"

Identificador	5.2.1
Nombre	Estudio del protocolo NEXUS utilizado por MACRAUT para la comunicación sobre TCP/IP
Descripción	
Comentarios	Consultar Espeficicación del protocolo en el documento interno de MACRAUT: "Protocolo de comunicaciones NEXUS Borrador 1"

Identificador	5.2.2
Nombre	Establecimiento de conexión con el cliente PE-VIEW
Descripción	Tras la creación de una estructura básica para el servidor basada en la implementación de un servidor sencillo (requisito 5.1.1), se procederá a crear un servicio capaz de conectar con el cliente PE-VIEW
Comentarios	

Identificador	5.2.3
Nombre	Recepción de peticiones de lectura
Descripción	El servidor socket deberá poder recibir una trama de petición de lectura a través del socket.
Comentarios	

Identificador	5.2.4
Nombre	Creación de las respuestas a las peticiones de lectura
Descripción	El servidor socket deberá poder elaborar una respuesta a una petición de lectura y enviarla al cliente.
Comentarios	

Identificador	5.2.5
Nombre	Recepción de peticiones de escritura
Descripción	El servidor socket deberá poder recibir una trama de petición de escritura a través del socket
Comentarios	

Identificador	5.2.4
Nombre	Creación de las respuestas a las peticiones de escritura
Descripción	El servidor socket deberá poder elaborar una respuesta a una petición de escritura y enviarla al cliente que haya realizado la petición.
Comentarios	

Identificador	5.3
Nombre	La comunicación vía socket tiene que poder integrarse con las tareas de gestión de mapeados y control de bloqueos

Identificador	5.3.1
Nombre	Prueba de integración con las librerías de gestión de mapeados
Descripción	Los mapeados a los que acceden las rutinas serán accedidos mediante la librería desarrollada para la gestión de lo mismos.
Comentarios	Ver requisito 2.0

Identificador	5.3.2
Nombre	Prueba de integración con las librerías de control de bloqueos
Descripción	Antes de que los datos contenidos en los mapeados sean actualizados o accedidos para su lectura, será necesario obtener un bloqueo que regule estas transacciones, con el objeto de evitar problemas de concurrencia.
Comentarios	Ver requisito 1.0

Identificador	5.4
Nombre	El servidor socket deberá ser capaz de atender peticiones de varios clientes de manera concurrente

Identificador	5.4.1
Nombre	Integrar una estructura que permita la creación de varios procesos para atender las peticiones de distintos clientes
Descripción	
Comentarios	Será necesaria la utilización de las llamadas proporcionadas por el sistema para la gestión de los procesos

Identificador	5.4.2
Nombre	Comprobar el funcionamiento mediante la petición simultánea de datos desde varios equipos
Descripción	
Comentarios	Con el objetivo de simular varias máquinas físicas realizando peticiones, pueden utilizarse máquinas virtuales sobre un mismo equipo que hagan uso de diferentes adaptadores de red virtuales.

3.2. Mantenimiento

El sistema facilitará la gestión remota permitiendo el inicio de sesión mediante telnet. El equipo se diseñará modularmente para que reemplazar componentes defectuosos sea una tarea sencilla de realizar.

3.3. Restricciones de diseño

3.3.1. Librería C del sistema embebido

Librería C disponible en el sistema embebido es µ-Clibc [15]. Se trata de una librería optimizada para sistemas embebidos teniendo en cuenta parámetros como rendimiento y espacio. La librería disponible en los sistemas Linux de escritorio es glibc [16], así que hay que tener especial cuidado en el desarrollo de prototipos sobre sistemas de sobremesa si no se tienen en cuenta estas restricciones.

3.3.2. Cumplir con las tramas ya en uso para el servidor socket

El servidor socket a desarrollar tiene que cumplir con las especificaciones ya definidas por MACRAUT para la gestión de los equipos ya existentes.

3.4. Documentación disponible y ayuda

Atmel ofrece especificaciones sobre el micro que integra la placa de desarrollo sobre la que se creará este proyecto, así como especificaciones hardware de la placa. [12]

Si bien contamos con amplia documentación en el punto anterior, la configuración y la interfaz hardware-software esta carentemente documentada por el fabricante. Para este punto, así como para la resolución rápida de problemas relativos a la puesta en funcionamiento y administración del sistema es mejor recurrir a comunidades de usuarios. [17]

Sobre el núcleo del sistema operativo linux, módulos disponibles, compilación la mejor fuente es la página oficial del proyecto [13]

Para temas de programación es mejor recurrir a páginas especializadas en el lenguaje que se esté utilizando. En este proyecto se está utilizando C así que una de las mejores referencias es el libro de sus creadores: El lenguaje de programación C [4].

Para una introducción al desarrollo y la configuración básica de la placa es recomendable comenzar con los "hands-on official training" de Atmel [18]

3.5. Componentes adquiridos

Los componentes adquiridos para ser utilizados en este proyecto se reducen a las placas de desarrollo STK1000 aportada por MCRAUT y la placa NGW100 aportada por el alumno.

En el futuro se prevé contar con un programador JTAG y un equipo secundario que utilice el bus I2C.

3.6. Interfaces

3.6.1. Interfaces de usuario

El sistema no proporcionará ningún tipo de interfaz de usuario en esta versión. En futuras versiones se puede considerar el añadir una pantalla que presente datos que ayuden al

diagnóstico, mantenimiento y monitorización, dada la capacidad de los equipos para manejar tanto pantallas LCD pequeñas como monitores a través del puerto VGA

Dada la naturaleza del sistema, no procede la interacción con un usuario diferente de un equipo electrónico.

3.6.2. Interfaces hardware

Diversos pulsadores y LED's están disponibles en la placa STK100 además de una pantalla LCD. No se utilizarán en este proyecto.

3.6.3. Interfaces software

Conforme se desarrollen las librerías, estas tienen que presentar un conjunto de funciones que hagan del desarrollo de nuevas funcionalidades una tarea sencilla. Para ello se tendrán que crear interfaces claramente definidas y bien documentadas.

3.6.4. Interfaces de comunicación

- Interfaz para actuar de maestro I2C: permite la comunicación con equipos esclavos conectados al mismo bus.

- Puerto serie para conexión con modem radio con conector dsub9

- Puerto serie secundario para manejar otro tipo de dispositivos. Útil para poder trabajar con módems GSM y GPRS

- Interfaz de red Ethernet con conector RJ45: resulta especialmente interesante ya que permite la conexión de adaptadores wi-fi que dotan al equipo de conectividad inalámbrica.

3.7. Estándares aplicables

3.7.1. Comunicaciones

- I2C: El equipo debe cumplir con el estándar definido en la especificación I2C
- MODBUS serial specification: El equipo debe cumplir con el estándar definido para la transmisión de paquetes modbus a través de una línea serie.

3.7.2. Documentación

- UML: la documentación generada para el modelado del sistema debe satisfacer la versión de UML definida hasta la fecha.
- IEEE Std 830 para la creación de documentos SRS

4. Información adicional

4.1. Libros:

[1] Miles, Russ. "Learning UML 2.0" O'Reilly. Abril 2006. ISBN 0596009828

[2] Fowler, Martin. "UML Distrilled Second Edition. A Brief Guide to the Standard Object Modeling Language." Addison-Wesley Object Technology. 2000 ISBN 020165783

[3] Pérez García, Miguel Ángel. "Instrumentación electrónica". 2004. ISBN 8497321669

[4] Kernighan Ritchie "El lenguaje de programación C".

4.2. Especificaciones:

[5] Object Management Group. "Unified Modeling Language: Superstructure" version 2.0 formal 05/07/04 August 2005

[6] Philips Semiconductors. "The I2C bus specification" version 2.1 January 2000.

4.3. Artículos:

[7] Friginal Santos-Olmo, Moisés. "Documento de especificación de requisitos para un proyecto de pequeño tamaño".

4.4. Trabajos académicos:

[8] Esparza Isasa, José Antonio. "Memoria de prácticas en MACRAUT Ingenieros" Verano 2008.

[9] Esparza Isasa, José Antonio. "Propuesta de proyecto: Plataforma de telecontrol sobre Linux embebido en arquitecturas AVR32".

4.5. Páginas web:

[10] macraut.com: web oficial de la empresa para la que se está desarrollando el producto.

[11] Ficha Z-HIO: información sobre módulos Z-HIO utilizados por el sistema PECOM.

[12] AVR 32-bit MCU AP7 application processors: web en la que se puede encontrar documentación técnica sobre el micro utilizado por el sistema.

[13] Kernel.org: página oficial del proyecto centrado en el desarrollo del núcleo de Linux, base del sistema operativo que se utiliza en el proyecto.

[14] AVR freaks NGW100 hardware reference: descripción del hardware utilizado en la placa de desarrollo NGW100, una de las utilizadas en este proyecto.

[15] µ-Clibc: librería de C optimizada para su uso en sistemas embebidos, debido especialmente a su consumo de memoria, tamaño y gestión de recursos

[16] glibC: librería de C utilizada ampliamente en los sistemas Linux.

[17] AVR freaks: comunidad de usuarios de productos de Atmel.

[18] STK1000: web oficial del fabricante Atmel sobre la placa de desarrollo utilizada en el proyecto.

4.6. Tutoriales y documentación específica

[19] Atmel Hands-On official training.