

Sistemas Informáticos

Curso 2008 - 2009

Proyecto FonHome

Autores: Miguel Ángel Díaz Calvo
Daniel Dionne González
David Martín Moreno

Profesor director: Juan Carlos Fabero Jiménez

Facultad de Informática
Universidad Complutense de Madrid





fon home



PRESENTACIÓN

Este proyecto consiste en el diseño e implementación de una aplicación capaz de gestionar dispositivos domóticos a través de un router denominado “La Fonera”, en el marco del Software Libre.

Dos son los principales objetivos del proyecto, con los que pretendemos innovar en el campo que nos ocupa. El primero es crear un sistema domótico muy económico (tanto por coste material como por mantenimiento), y el segundo es que este sistema domótico no esté basado en un PC que tenga que estar encendido permanentemente.

Objetivos secundarios son que la aplicación sea independiente de la tecnología domótica utilizada (X10, ZigBee, RF), y que se pueda acceder al sistema de forma remota, o de forma local a través de un dispositivo móvil con WiFi.

La aplicación se centra sobre todo en la gestión y manejo de dispositivos X10, para ello presenta una interfaz amigable con la que el usuario podrá configurar sus dispositivos X10 sin ningún problema.

El principal problema al que nos hemos enfrentado ha sido trabajar sobre una plataforma que está actualmente en desarrollo. La Fonera cambió nueve veces de software (y una de hardware) en el transcurso del proyecto. Las últimas actualizaciones afectaron sobre todo al espacio disponible para el proyecto.

ABSTRACT

The purpose of this project is to design and implement an application that is capable of managing domotic devices using a router called “La Fonera”, in the Open Software paradigm.

The project has two main goals, which try to innovate in the domotics field. The first one is to build a low cost domotic system (using cheap materials and reducing maintenance costs), and the second is that the system is not based on a computer that has to be running constantly.

Secondary objectives are that the application is independent from the used domotic technology (X10, ZigBee, RF), and that the system can be accessed remotely, or locally from a WiFi enabled mobile device.

The designed application is mainly focused on X10 device control and management. For this purpose we designed a friendly interface that lets the user manage X10 devices with ease.

The main problem in the development of this project was to base our work on a platform that is under development. La Fonera had nine software updates and one hardware update during the development of this project. The latest updates affected in particular the available space for the application.



ÍNDICE

Contenido

- Presentación 3
- Abstract 3
- Índice 5
- Introducción 9
- Aspectos destacados 10
- Arquitectura 10
 - Hardware 11
 - La Fonera 11
 - Hardware X10 12
 - Conexión La Fonera – X10 14
 - Software 16
 - Problemas con la librería de C++ 17
- Diseño 18
 - Plugins 21
 - Plugins de eventos 21
 - Plugins de acciones 21
 - Plugins de interfaz 22
- Zonas 22
- Eventos, condiciones y acciones 22
 - Eventos 22
 - Condiciones 22
 - Acciones 23
- Diseño del diagrama de estados del módulo FH 24
- Base de datos 24
 - Introducción 24
 - SQLite 24
 - Formato de las tablas de SQLite* 25
 - Cambio de SQLite a UCI 26
 - UCI 26

<i>Comandos UCI</i>	26
<i>Formato de las tablas de UCI</i>	28
Ejemplo de archivos UCI	29
Aplicación para La Fonera	29
Características	29
Estructura	30
Comunicación módulo FH – aplicación	32
Diseño	32
Mensajes X10.....	33
Mensajes de configuración.....	33
Desarrollo	34
Planificación.....	34
Casos de uso – Trabajo futuro.....	35
Videoportero en el móvil.....	35
Control por email.....	35
Control desde otras fuentes	36
Gestión inteligente de temperatura.....	36
Gestión inteligente de iluminación.....	36
Ejemplos de ejecución	37
Modos de ejecución para depuración	37
Modo depuración	38
Modo depuración de base de datos.....	38
Modo sin monitor	39
Modo ruta de Heyu	39
Modo por defecto.....	39
Promoción del proyecto	40
Conclusiones.....	43
Instalación	44
Instrucciones para instalar la aplicación.....	44
Prototipo CGI	44
Aplicación para La Fonera.....	45
Manual.....	46
Índice	46



Introducción	46
Panel de control.....	46
FonHome	47
Control de dispositivos	47
Gestionar dispositivos	49
Programar acciones.....	50
Registro de eventos.....	54
Copia de seguridad	54
Configuración	55
FonHome móvil	57
Anexos	59
Anexo 1. Protocolo X10	59
Anexo 2. Manipulando La Fonera.....	62
Cómo cargar/descargar archivos a/desde La Fonera	62
Cómo compilar para La Fonera.....	63
Anexo 3. Enlaces.....	65
Agradecimientos.....	67
Autorización a la UCM	69
Palabras clave	71



INTRODUCCIÓN

La idea original del proyecto fue crear una aplicación capaz de controlar dispositivos domóticos que fuera accesible para teléfonos móviles. Al principio estuvimos investigando sobre la plataforma Android, para ver las posibilidades que nos brindaba. Después de hablar con Juan Carlos, nuestro tutor, decidimos ampliar el proyecto, y surgieron dos ideas. La primera, desarrollar el proyecto sobre una “Fonera”, un router WiFi 802.11b/g con Linux 2.4, con un firmware basado en OpenWRT. La segunda era hacer el desarrollo sobre una FPGA proporcionada por la facultad. El acceso desde un dispositivo móvil quedó en segundo plano.

Finalmente se optó por la opción de La Fonera por dos razones: primero por las dificultades para conseguir una FPGA prestada, y segundo por todas las posibilidades que nos proporcionaba La Fonera. En este momento, se cambiaron los objetivos del proyecto, aunque el desarrollo sería muy similar, con el fin de aprovechar las virtudes de La Fonera y hacer el proyecto más atractivo comercialmente y llamar la atención de la comunidad FON.

Los objetivos, influenciados por la política de FON, pasaron a ser los siguientes:

1. Crear un sistema domótico muy económico (tanto por coste material como por mantenimiento)
2. Que este sistema domótico no esté basado en un PC que tenga que estar encendido permanentemente

Nos dimos cuenta de que íbamos por buen camino en este sentido cuando nos pusimos en contacto con FON, y después de hablar con Inouk Bourgon recibimos tres routers (totalmente gratis) para el desarrollo del proyecto.

Más tarde, después de abrir la página web www.fonhome.com, recibimos un email de un empresario de Teruel, que estaba interesado en implementar nuestro proyecto en una granja ecológica para controlar de forma remota la iluminación y las puertas de unos galpones donde guardan ocas. Tuvimos una reunión con este señor, que nos ofreció un puesto de trabajo y alojamiento en Teruel para implantar este sistema en un futuro.

Otra decisión que tuvimos que tomar al principio fue elegir una tecnología domótica apropiada, que a poder ser estuviera extendida y al alcance de todos. Investigamos durante un tiempo y limitamos las opciones a dos: utilizar X10 o ZigBee. Volvimos a evaluar cuáles eran nuestros intereses a la hora de desarrollar la aplicación y decidimos usar X10, debido a que estaba mucho más extendido y ser más barato que ZigBee. De todas maneras, se tuvo en mente la posible futura ampliación del proyecto, haciendo un diseño modular que permitiese la posterior incorporación de distintos protocolos de comunicación. Para este propósito se decidió hacer el proyecto sobre C++.

Después de esta elección lo que hicimos fue comprar dispositivos X10 que pudiéramos usar en casa y que nos sirvieran para el desarrollo de la aplicación.

En principio no pensamos que trabajar sobre una plataforma en desarrollo como es La Fonera fuera a traernos problemas. Los constantes cambios en el firmware (sobre todo en el espacio libre para plugins cada vez más reducido) e incluso el cambio hardware de La Fonera+ a La Fonera 2.0, repercutieron constantemente tanto en el diseño como en la implementación del proyecto. Hasta el último momento tuvimos que hacer cambios. Los más importantes fueron los siguientes:

- Cambiar SQLite a UCI por falta de espacio, desechando parte del diseño original por falta de funcionalidad en UCI
- Desechar el circuito implementado para conectar por puerto serie al salir La Fonera 2.0
- Cambiar *libstdc++* por *uclibc++* (no se llegó a conseguir por problemas en la compilación cruzada)
- Cambiar la implementación del interfaz de CGI a Lua por cambios en el firmware

Actualmente la aplicación ha cumplido los siguientes objetivos:

- Control de dispositivos
- Configuración de dispositivos
- Programación de acciones, alarmas
- Registro de eventos
- Copia de seguridad
- Configuración horaria
- Acceso a la aplicación vía móvil

Para trabajar se abrieron un grupo de desarrollo en <http://groups.google.com/group/si0809> y un repositorio de código en <http://code.google.com/p/si0809/>. Ambos están abiertos al público.

ASPECTOS DESTACADOS

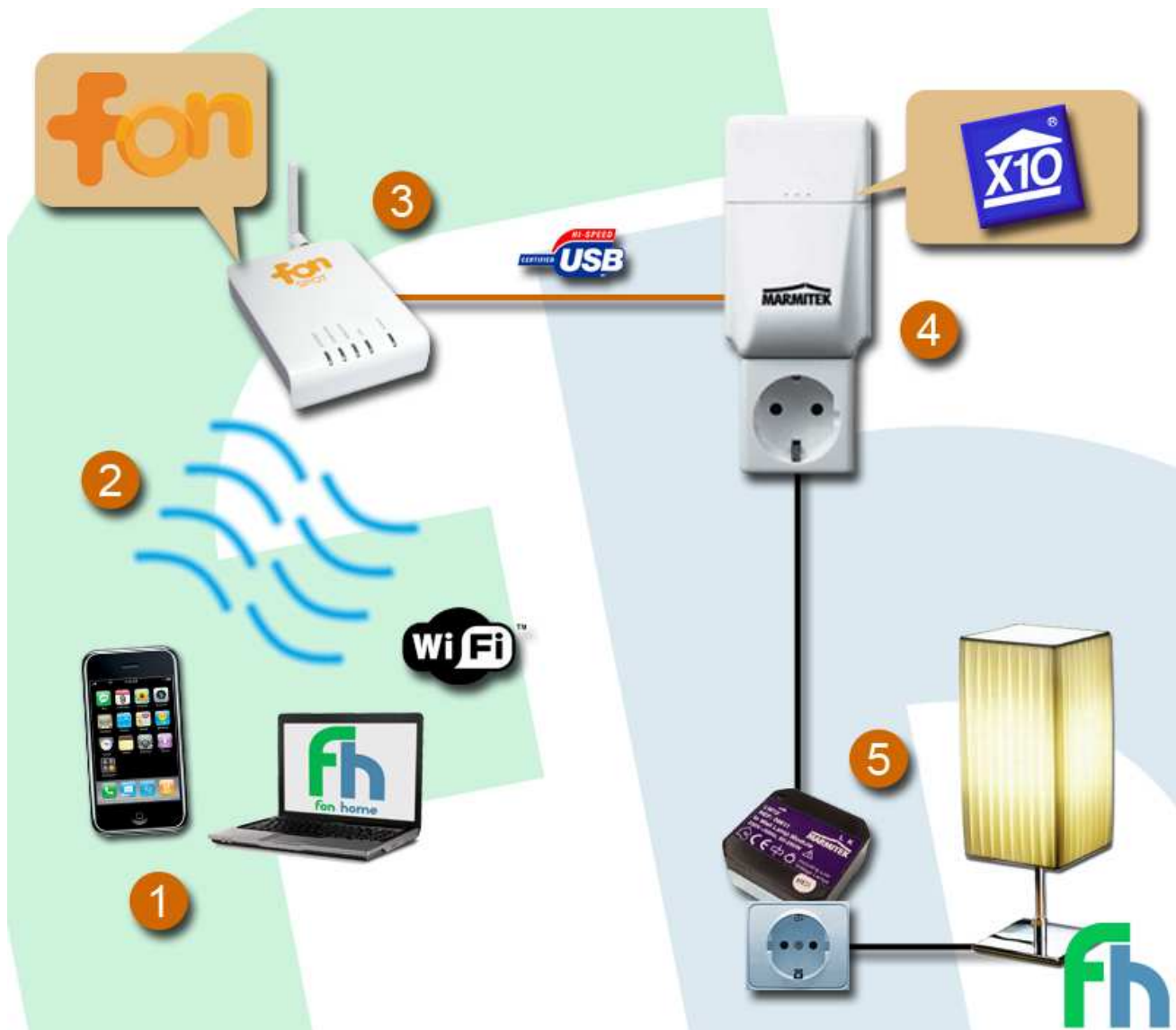
- Sistema muy económico: **alrededor de 80€ frente a otros sistemas comerciales de 400€ - 3000€** (se puede ver un cuadro comparativo en www.fonhome.com, en el apartado “¿Qué es FonHome?”)
- Ahorro importante en el consumo eléctrico: al no necesitar un PC funcionando, **este consumo se reduce a los 5W de La Fonera** (fuente: http://wiki.fon.com/wiki/Energy_consumption)
- Diseño modular que permite cambiar la tecnología domótica
- 100% software libre
- Completamente integrado en La Fonera: utiliza el interfaz LuCi y el sistema de archivos UCI.
- Innumerables posibilidades de expansión: gracias a las funcionalidades que ofrece La Fonera (conexión a Internet, USB, WiFi, interfaz web...), las posibilidades de expansión son casi las mismas que si se trabajase sobre un PC.

ARQUITECTURA

El usuario se comunica mediante un teléfono móvil o un ordenador (1) que tenga tecnología WiFi (2) con La Fonera (3), que es el cerebro del sistema.

La Fonera traduce la señal, y la envía por puerto USB al controlador CM11 (4). A partir de aquí el sistema es como cualquier sistema domótico con tecnología X10. El controlador envía y recibe señales en este formato a los módulos que se encargan de encender y apagar los distintos dispositivos que estén conectados al sistema.

Además La Fonera puede enviar mensajes al controlador sin interacción del usuario a partir de la programación que éste haya grabado previamente y responder a señales recibidas del controlador.



Hardware





La Fonera

La Fonera+:

Cuando empezamos el proyecto La Fonera que mejor se adaptaba a nuestras necesidades era La Fonera+. No dispone de USB. Tiene un puerto COM o RS-232 interno con el que nos podíamos comunicar con el controlador domótico. El problema de ese puerto es que tiene niveles de voltaje distintos a los del estándar RS-232 (niveles TTL frente a $\pm 15V$) por lo que para poder utilizarlo hay que añadir un circuito que convierta el puerto de La Fonera+ en un RS-232 puro.

La Fonera 2.0:

Al salir un modelo de La Fonera con un puerto USB nos hemos evitado tener que utilizar el circuito conversor. Además la nueva La Fonera 2.0 está enfocada a que cualquier desarrollador pueda programar aplicaciones para ella, por lo que nuestra aplicación puede tener mayor difusión por no tener que modificar el hardware de La Fonera para poder utilizarla. Además FON ha creado una comunidad para desarrolladores llamada Fonosfera en la que aportan documentación muy útil para el desarrollo para La Fonera 2.0.

	La Fonera 2100	La Fonera 2200	La Fonera + (2201)	La Fonera 2.0 (2202)
Hardware				
Dimensiones	93.5 x 25.5 x 70	93.5 x 25.5 x 70	93.5 x 25.5 x 110	93.5 x 25.5 x 110
Procesador	MIPS 4KEc V6.4 - 183.50 MHz	MIPS 4KEc V6.4 - 183.50 MHz	MIPS 4KEc (Revisión 00019064)	-
Wireless	Atheros AR531X_COBRA	Atheros AR531X_COBRA	Atheros AR2315	Atheros AR2315A-001
Memoria Flash	8 MB	8 MB	8MB	8MB
SDRAM	16 MB	16 MB	16MB	32MB
Corriente	5V 2A	7.5V 1A	7.5V 1A	5V 2A
Ethernet/USB	1/0	1/0	2/0	2/1
LEDS	3	3	4	5
Fotos internas				
Firmware				
Versión Busybox	1.1.3	1.1.3	-	-
Kernel	2.4.32	2.4.32	2.6.19.2	2.6.26.2
Versión IPKG	0.99.162	0.99.162	-	0.1.5 (OPKG)

Fuente: http://www.tufonera.com/index.php/Comparacion_de_foneras

Hardware X10

Estos son los componentes hardware que hemos usado, aparte de La Fonera.

Controlador X10:

Controlador para PC CM11



El módulo bidireccional CM11 es un interfaz universal de domótica que permite a cualquier producto compatible transmitir y recibir datos desde y hacia los dispositivos X10.

Se conecta por puerto serie o por USB utilizando el conversor basado en el chip PL2303 que viene incluido.

Viene además con software y drivers para Windows.

Módulos X10:

Módulo de lámpara LM12 (con dimmer)



Función de encendido/apagado y regulación de intensidad para controlar bombillas resistivas (incandescentes) de 40 W a 300 W.

No es compatible con bombillas de bajo consumo, fluorescentes, y otros elementos capacitivos.

Módulo de aparato AM12 (sin dimmer)



Función encendido / apagado para controlar luces fluorescentes, aire acondicionado, ventiladores, máquinas de café, etc.

Puede manejar luces incandescentes y fluorescentes de hasta 500 W de potencia, así como motores eléctricos de 250 W o lámparas halógenas con regulación.

Además tiene la ventaja de poder ser conectado y desconectado de forma local. Es decir, que aunque el módulo este apagado, si se apaga y luego se enciende el aparato, éste se enciende ya que el módulo pasa a estado de conexión automáticamente (importante en caso de que el sistema domótico falle).

Interfaz bidireccional XM10



Igual que el controlador CM11, pero mucho más simple. No tiene implementado el protocolo X10, tan sólo lo filtra (tiene un LED que se enciende cuando detecta un mensaje X10) y cambia el de nivel de corriente alterna (220V) a puerto serie y viceversa.

Éste es el módulo que compramos primero, pero no llegamos a usarlo. Se habla de esto en el siguiente apartado.

Sensor de presencia exterior MS13



Tiene un sensor de movimiento por infrarrojos y un sensor lumínico. Utiliza dos códigos de aparato consecutivos, uno para cada sensor, y transmite la información por radio utilizando un protocolo del fabricante.

Se alimenta con dos pilas AAA, y está diseñado para poder ponerse a la intemperie.

Receptor de radiofrecuencia TM13



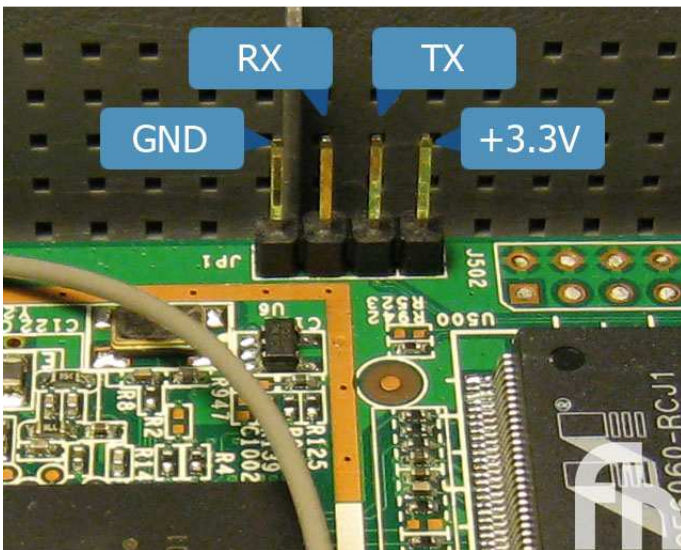
Receptor de radiofrecuencia. Utiliza un protocolo de radio diseñado por el fabricante. Traduce las señales de radio al protocolo X10 y las envía por la red eléctrica.

Además es módulo de aparato, como el AM12.

Se puede consultar más información sobre los aparatos usados en la siguiente dirección:

<http://groups.google.com/group/si0809/web/manuales-mdulos-x10>

Conexión La Fonera – X10



Antes de salir La Fonera 2.0, la conexión del módulo X10 a La Fonera no era algo trivial. Para empezar, La Fonera+ no tenía un puerto externo disponible para conectar nada. Había que abrirla, y utilizar cuatro pines de la placa como puerto de comunicación serie (los cuatro que se ven en la foto de la izquierda).

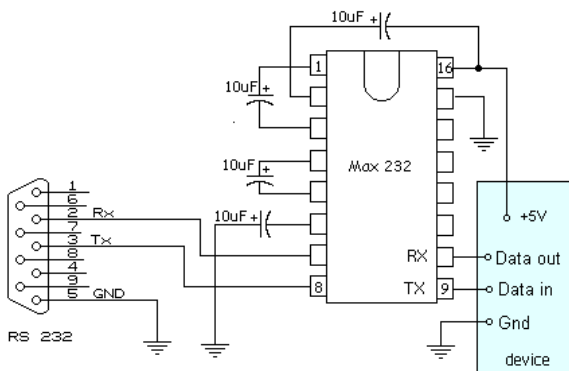
Estos pines funcionan a 0-5V (TTL), mientras que el rango del módulo es $\pm 12V$ (serie).



Circuito MAX232 con clavija DB9

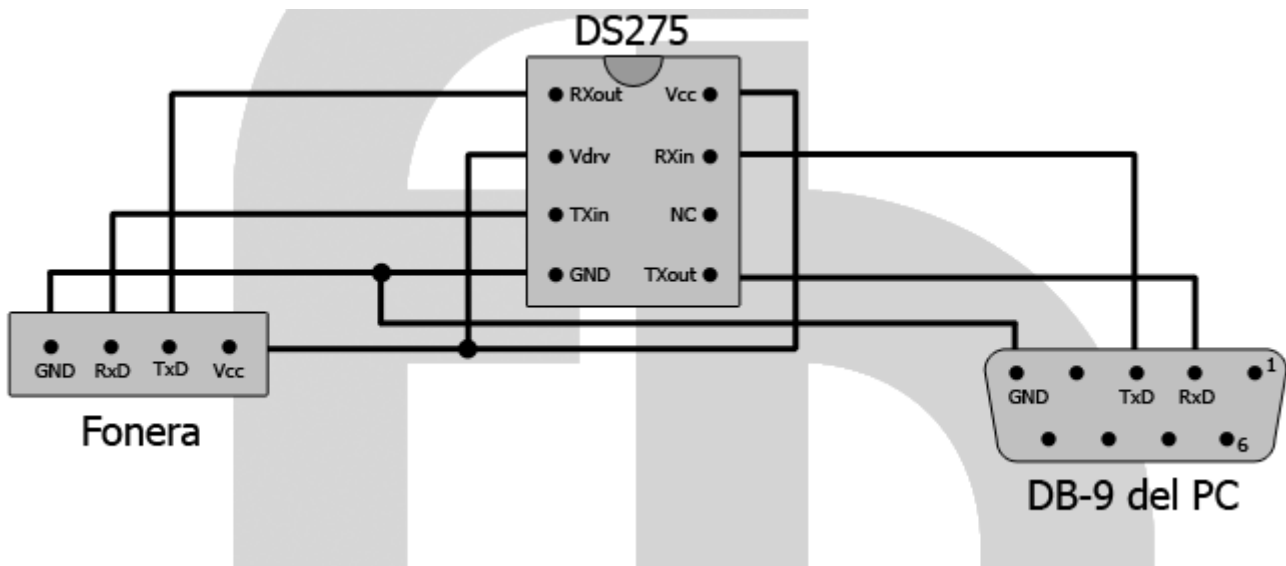
El primer módulo que compramos fue el XM10 visto en el apartado anterior. Fue un error comprarlo, debido a una descripción del producto un poco incompleta por parte del proveedor, pues en realidad queríamos comprar el CM11. Dado que este módulo no tiene implementado el protocolo X10, nuestro tutor nos propuso diseñar un pequeño circuito cambiador de nivel entre La Fonera y este módulo, e implementar nosotros el protocolo X10 por software.

Después de estudiar este inconveniente, y con motivo de intentar aprovechar el módulo que habíamos comprado por equivocación, nos dimos cuenta de que hacer este circuito era complicado y nos iba a llevar mucho tiempo; además, se desviaba de los objetivos originales del proyecto y tenía solución fácil: comprar el CM11.

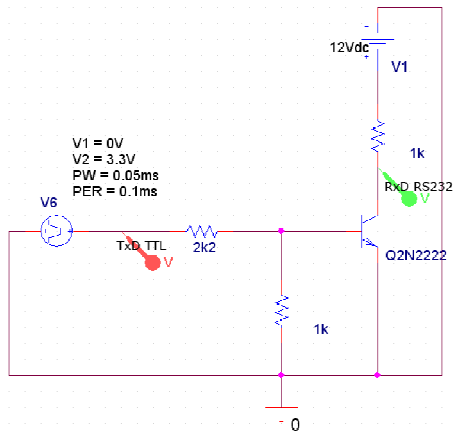


El circuito resultó ser bastante más que un cambiador de nivel. La dificultad radicaba en las diferencias entre el protocolo X10 y el serie. Por un lado, X10 tiene mensajes de 10 bits, frente a los 8 o 16 del serie. Por otro, el X10 se gobierna por los ciclos de la red eléctrica, y el XM10 proporciona estos bits junto a una señal de reloj que hay que usar para ir guardando la información en un registro (dos si la comunicación es bidireccional).

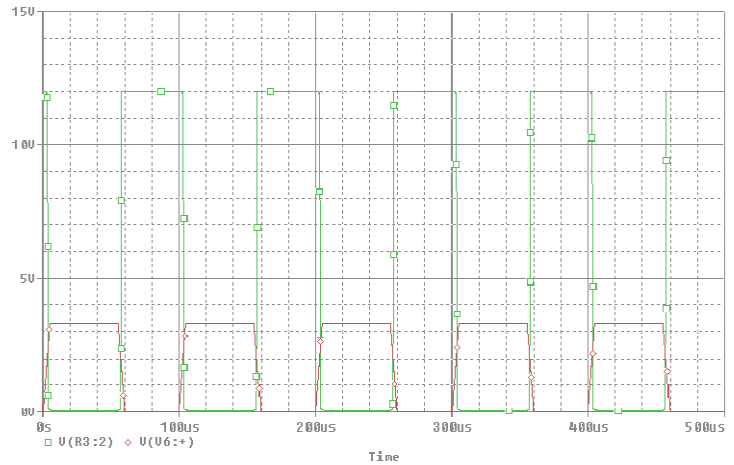
Una vez comprado el CM11 volvimos a la idea del circuito cambiador de nivel. Se vieron varias alternativas. Una de ellas fue utilizar el chip MAX232, como el de las fotos de la izquierda, o un chip DS275:



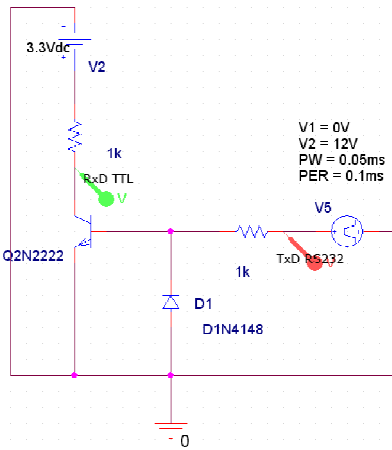
Antes de probar las soluciones anteriores (el DS275 es muy difícil de encontrar, y el circuito necesario para montar el MAX232 era un poco grande), se probó a implementar el siguiente circuito, con idea de que se pudiera introducir en La Fonera, dejándola cerrada sin que se note la modificación.



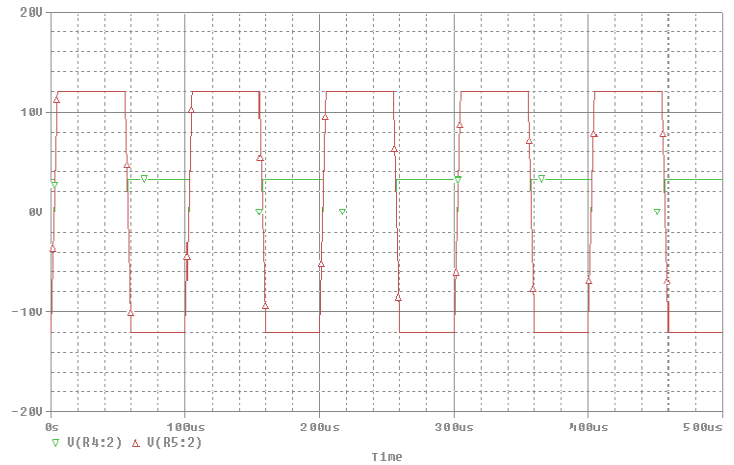
Circuito conversor TTL a RS232



Simulación del circuito en PSPice

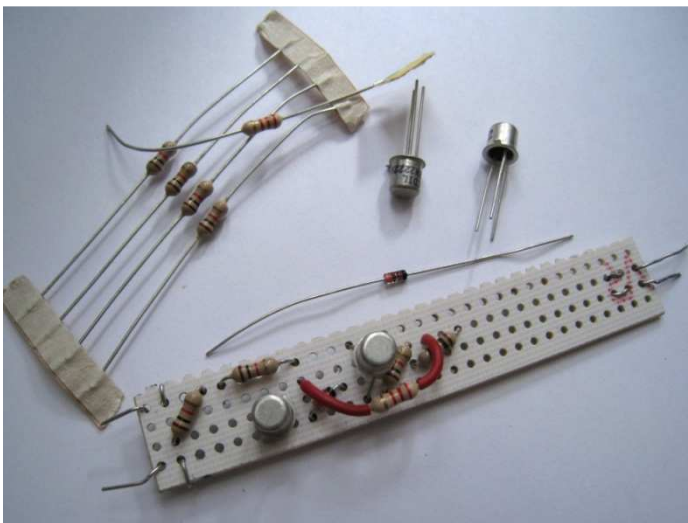


Circuito conversor RS232 a TTL



Simulación del circuito en PSPice

Implementación de ambos circuitos implementados sobre una placa para integrar en La Fonera:



Se diseñó la placa con idea de añadir un conector que permitiese enchufar el circuito directamente en La Fonera, de forma que quedase todo dentro de la carcasa.

Finalmente nunca llegamos a utilizar este circuito, pues recibimos las nuevas La Fonera 2.0 de FON, con puerto USB.

Más información sobre este circuito en la siguiente dirección:

<http://groups.google.com/group/si0809/web/circuito-simple>



Software

El lenguaje de programación que usamos para implementar la aplicación para La Fonera fue C++ utilizando el paradigma de Programación Orientada a Objetos.

Inicialmente, para implementar la interfaz web en La Fonera utilizábamos HTML con CGI (Common Gateway Interface), pero en siguientes versiones de firmware ya no estaba disponible, por lo que tuvimos que migrar toda la interfaz web a HTML con Lua (<http://www.lua.org/>).

Empezamos diseñando la aplicación para interactuar con una base de datos en SQL pero, por problemas de falta de espacio en La Fonera, tuvimos que cambiar a archivos de configuración UCI.

El entorno de desarrollo que hemos utilizado ha sido eclipse (<http://www.eclipse.org/>) para Linux con los siguientes plugins:

- Eclipse C/C++ Development Tooling - CDT (<http://www.eclipse.org/cdt/>): desarrollo en C/C++.
- Subclipse (<http://subclipse.tigris.org/>): cliente SVN.
- Web Standard Tools (<http://www.eclipse.org/webtools/wst/main.php>): herramientas para el desarrollo de webs.

Para intercambiar archivos entre el ordenador y La Fonera utilizamos el programa WinSCP. Para comunicar por SSH desde Windows utilizamos el programa PuTTY (ver Anexo 2. Manipulando La Fonera).

El compilador utilizado ha sido GCC y el SDK de OpenWRT (ver Anexo 2. Manipulando La Fonera).

En la implementación se utilizó la herramienta Heyu (<http://www.heyu.org>) como controlador del CM11, y el driver del conversor PL2303 de USB a RS232.

Problemas con la librería de C++

Las primeras versiones del firmware de La Fonera contenían la librería de C++ estándar, libstdc++. Según fue avanzando el desarrollo de La Fonera se sustituyó esta librería por otra más ligera, libuClibc++.

Esta nueva librería no está debidamente documentada. Hemos intentado compilar la aplicación con esta nueva librería pero no lo hemos conseguido por problemas con el SDK de OpenWRT. Preguntamos tanto en el canal IRC de la Fonosfera como en la lista de desarrolladores si nos podían ayudar pero nadie nos supo ayudar.

Como el espacio disponible en La Fonera para aplicaciones desarrolladas por terceros se ha visto muy reducido, la pesada librería estándar de C++ no se puede incluir en una aplicación, ya que no se dispone de espacio suficiente para poder alojarla.

Para poder seguir desarrollando la aplicación con la librería estándar de C++ mientras se sigue buscando una solución al problema de compilación con uClibc++ se nos han ocurrido dos posibles alternativas:

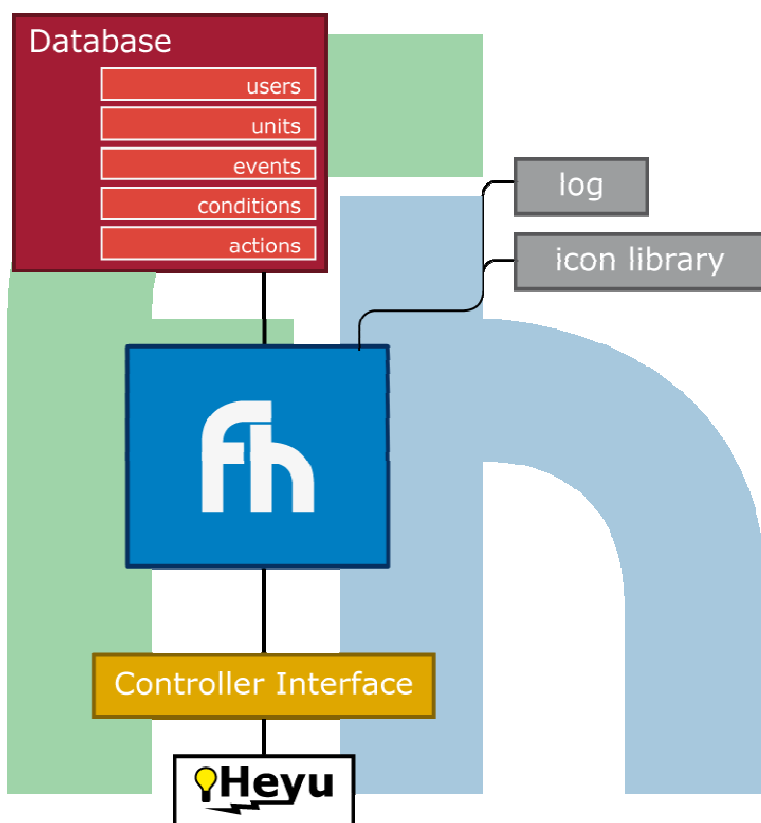
- Incluir la librería comprimida en la aplicación y descomprimirla en cada inicio en el directorio /tmp, ya que éste se borra al apagarse La Fonera.
- Utilizar una memoria USB para ampliar la memoria disponible de La Fonera.

La segunda forma es la más lógica, ocupa menos memoria en La Fonera y no tiene que descomprimirla al inicio.

FON utilizaba C++ para dar soporte a la aplicación de descarga de Torrent. Han tenido problemas de estabilidad con esta aplicación y en las versiones de firmware 2.2.6.0 RC1 y RC2 han cambiado la aplicación por otra que no está implementada con C++, por lo que tampoco incluyen la librería uClibc++. Si esto hubiese pasado antes probablemente hubiésemos desarrollado la aplicación en C para ahorrarnos todos estos problemas, cosa que no descartamos hacer en un futuro.

DISEÑO

Diseño original



Éste es el diseño original que se hizo al comienzo del proyecto.

El centro del sistema se denomina FH (FonHome), y es el encargado de comunicar las distintas partes del sistema y tomar las decisiones.

La base de datos guarda información sobre los usuarios, los dispositivos, los eventos, las condiciones y las acciones (explicadas en la página 22).

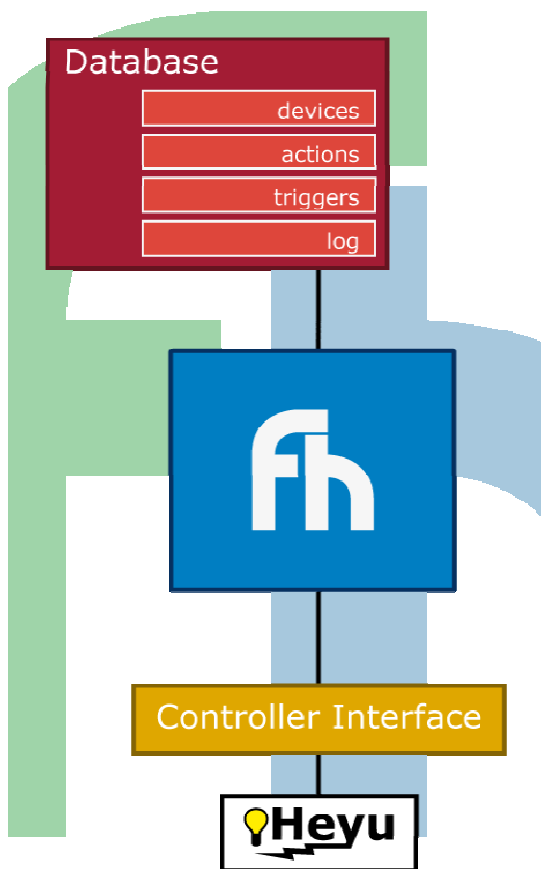
Heyu es el controlador al que se le mandan las instrucciones, y CM11 es el hardware.

El log almacena toda la información posible sobre el funcionamiento del sistema.

El icon library contiene un repositorio de iconos que se pueden asignar a los distintos dispositivos, con el fin de hacer la aplicación más intuitiva. No se implementó por falta de espacio en La Fonera.

El Controller Interface permite abstraer la capa de hardware, de forma que se pueda intercambiar la tecnología utilizada, por ejemplo de X10 a ZigBee.

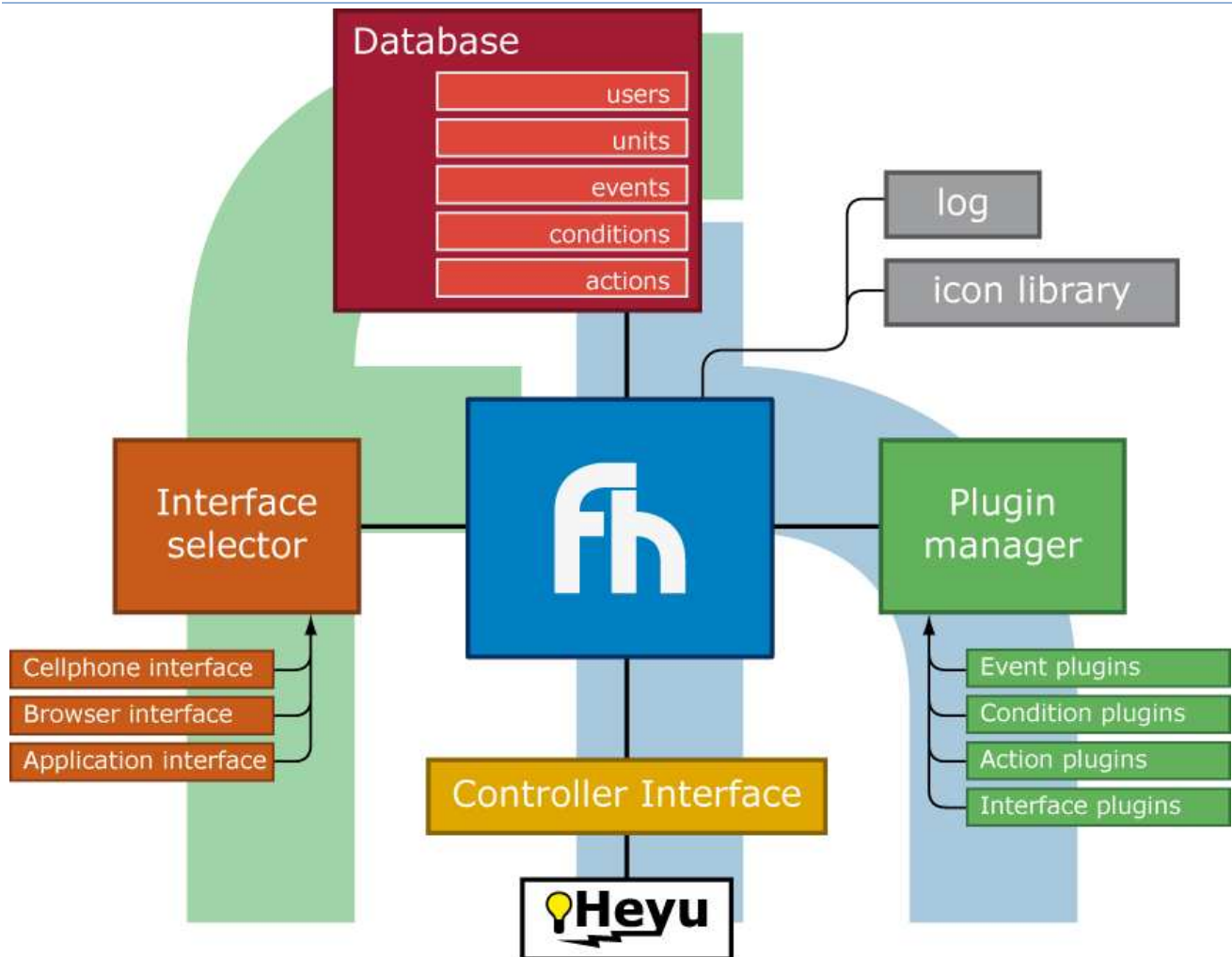
Diseño actual



Debido a la integración del sistema en el interfaz de La Fonera, la parte de interfaces no se llevó a cabo de la manera pensada originalmente. Se diseñó una interfaz completamente integrada en LuCI, el sistema que utiliza FON, y otra interfaz preparada para dispositivos móviles, utilizando el Universal iPhone UI kit.

La base de datos sufrió modificaciones, principalmente debido a la restricción de espacio en La Fonera. Esta limitación nos obligó a prescindir de una base de datos basada en SQL, así que se empezó a usar el sistema UCI basado en archivos de texto, que ofrece una funcionalidad mucho más limitada. Dado que hacer una base de datos compleja no era el objetivo del proyecto, se prescindió de la recursividad en la base de datos, recursividad imprescindible en el diseño original (véase más sobre el diseño original de la base de datos en el apartado de Eventos, condiciones y acciones en la página 22)

Como se ha comentado, el repositorio de iconos se eliminó.



Este sería un posible diseño completo del sistema.

Los módulos marrones de la izquierda representan los distintos interfaces, distintas formas de acceder a la aplicación, y el selector de interfaz se encargaba de seleccionar el apropiado según cómo se ha accedido.

El gestor de plugins se encarga la administración de plugins, y se los suministra al FH.

La BD almacena toda la información de configuración del sistema.

Los plugins de eventos sirven para añadir eventos como “pulsado el botón de la cocina”, “hace 26°C”, “email entrante” o “es medianoche”. Los eventos disparan acciones y/o otros eventos.

Los plugins de acciones hacen cosas como “encender la luz”, “mandar un sms de alerta” o “activar modo vacaciones”.

Las condiciones se introducen entre los eventos y las acciones, siguiendo el esquema evento -> condición -> acción, y permiten cosas como “pulsado botón baño” -> “si es de madrugada” -> “encender luz baño al 20%” (en otro caso, al 100%)

Los plugins de interfaz permiten hacer consultas que requieren un interfaz aparte. Por ejemplo, permiten consultar el consumo eléctrico o la temperatura (dibujado en una gráfica), o cuántas veces se ha producido



un evento, como puede ser cuántas veces han llamado al teléfono o al timbre (interesante si hemos estado fuera de casa). Todo esto sale a partir del log.

Los plugins se interpretan en los módulos de interfaz (los verdes). Por ejemplo, un plugin de evento que sea un temporizador (una alarma) tendría un interfaz que sería un par de campos de texto numéricos separados por ":". Un plugin de evento que sea un detector de botón pulsado tendría un interfaz con dos selectores, uno de casa y uno de unidad. Esta definición se puede formalizar y enviar al módulo de interfaz. Éste construirá el interfaz apropiado utilizando los componentes específicos para el sistema que esté diseñado, a partir de las referencias formalizadas.

Por ejemplo, el evento temporizador tendría un interfaz `<campoTexto,0-23>":"<campoTexto,0-59>`, y el interfaz para navegador convertirá eso a un formulario con los campos de texto definidos por etiquetas HTML, mientras que el interfaz para una aplicación instanciará los widgets de campo de texto correspondientes.

Abstrayendo esta capa, el temporizador podría tener un interfaz que sea simplemente `<reloj>`. El interfaz de aplicación pintaría un reloj al que se le pudieran arrastrar las manillas con el ratón, mientras que el interfaz para un dispositivo móvil lo traduciría a dos campos de texto.

Plugins

Para separar partes diferentes de la implementación y facilitar la adición de nuevos módulos, se podrían implementar plugins. Se distinguen tres tipos de plugins, como se describe a continuación. La idea es que los plugins se puedan añadir en cualquier momento, y que la aplicación venga con unos plugins básicos.

Plugins de eventos

Un plugin de evento añade un evento nuevo a la aplicación. El evento básico que viene con el sistema sería el comando X10 de entrada. Otros plugins que generen eventos serían:

- Temporizador: genera un evento cada cierto tiempo
- Alarma: genera un evento en un momento determinado
- Calendario: parecido al de alarma, pero más completo, genera eventos en determinadas fechas o días de la semana
- Tarifador: descarga las tarifas de la compañía eléctrica de internet (si estuvieran disponibles) y genera un evento en el momento del cambio de tarifa (diurna o nocturna)
- Cliente de correo: consulta una dirección de correo, interpreta instrucciones predeterminadas, y genera los eventos apropiados

Cada plugin añade una entrada a la lista de eventos seleccionables, y tiene su propia interfaz de configuración.

Plugins de acciones

Un plugin de acción añade una acción nueva a la aplicación. La aplicación básica que viene con el sistema sería el comando X10 de salida. Otros plugins que generen acciones serían:

- Activador de cámara WiFi
- Cliente de correo: envía un email a una dirección de correo tras ocurrir el evento asociado
- Alarma: conecta por teléfono con el servicio de seguridad

Cada plugin añade una entrada a la lista de acciones seleccionables, y tiene su propia interfaz de configuración.

Plugins de interfaz

Un plugin de interfaz añade una interfaz a la aplicación. Recoge datos a partir de los eventos para mostrárselo al usuario, pero no produce ninguna acción. Ejemplos de plugins de interfaz:

- Temperatura: muestra al usuario un gráfico con la temperatura en el interior y el exterior de la casa durante el día, la semana, el mes, o el año.
- Consumo: muestra el consumo eléctrico en función del tiempo que cada dispositivo está conectado

Zonas

- Casa
 - Piso
 - Habitación/pasillo
 - Elementos
 - Subdivisión
 - Elementos

La división de una habitación en subáreas atiende al ejemplo de un loft (o cualquier ejemplo de habitación que cumple diferentes funciones). Si por ejemplo la casa tiene en una misma habitación cocina y comedor, puede ser interesante encender las luces de la zona de la cocina, sin encender las del comedor, pero puede no tener sentido establecer diferentes temperaturas para subdivisiones de una misma habitación.

Idea: Con la estructura apropiada, podría existir una instrucción del tipo “**encender luces de camino al dormitorio principal**”, lo que encendería las luces del pasillo, la escalera, el distribuidor del piso de arriba, y del dormitorio.

Eventos, condiciones y acciones

Idea original

Eventos

Un evento es un suceso provocado por una determinada situación. Tiene asociado un disparador, que es el suceso que provoca el evento, y una lista de acciones, condiciones y eventos, que se ejecutan al activarse el disparador.

El disparador en general será un comando X10, como por ejemplo una pulsación de un botón o un cambio de temperatura, pero también puede ser un comando enviado desde el dispositivo móvil o el interfaz web.

Se puede pensar en un evento como en una interrupción, y la lista de acciones y eventos como en una rutina de tratamiento de interrupción.

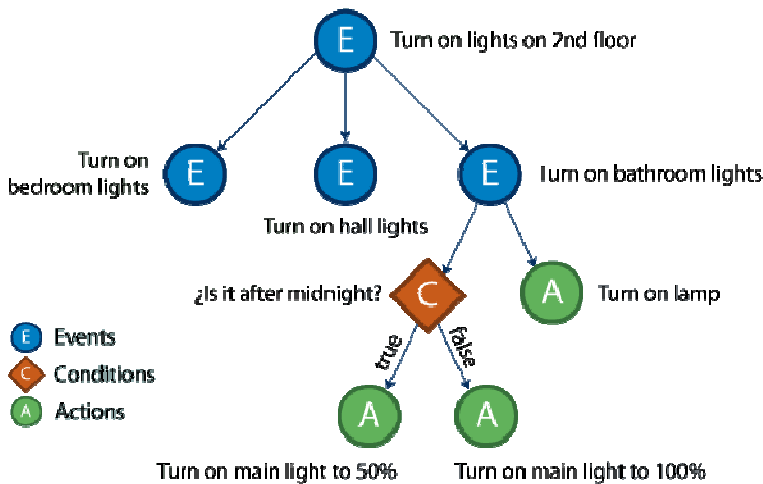
Condiciones

Las condiciones son eventos condicionados, están gobernadas por una cláusula que puede ser verdadera o falsa. Si la cláusula es verdadera, se ejecuta una lista de acciones, condiciones y eventos, y si es falsa, se ejecuta una lista distinta.



Acciones

Una acción es una modificación que se hace sobre el sistema. La acción es disparada por un evento, y ejecuta una instrucción. La instrucción en general será un comando X10, pero en casos particulares puede ser otro tipo de acción, como enviar un email, o activar una cámara WIFI.



La recursividad de los eventos (un evento puede disparar un evento) permite agrupar fácilmente las acciones. Por ejemplo, encender las luces del baño del piso de arriba podría implicar las acciones “encender luz espejo baño” y “encender luz techo baño”. Pero, por ejemplo, encender las luces del piso de arriba implica “encender luz espejo baño”, “encender luz techo baño”, “encender luz techo pasillo”, “encender lámpara 1 dormitorio”, “encender lámpara 2 dormitorio”. Sin embargo, la estructura recursiva simplifica el ejemplo según el gráfico de la izquierda.

En el gráfico, los eventos son los círculos azules marcados con E, y las acciones son los verdes marcados con A. Las condiciones son los rombos naranjas marcados con C. Activar el evento “encender luces baño 2” sólo encendería las luces del baño, mientras que “encender luces piso 2” encendería todas las luces, sin tener que especificar de nuevo las acciones que se necesitan para encender todas las luces de dicho piso (incluyendo las del baño).

En la condición del ejemplo se comprueba si ha anochecido. Si se lanza el evento “encender las luces del baño”, en caso de ser de noche, se enciende la luz a mitad de potencia. En caso contrario, se enciende a máxima potencia.

Puesto que las condiciones también son recursivas, se pueden construir condiciones complejas, como “si hace más de 25°C” Y “si la calefacción está puesta” ENTONCES “bajar la calefacción” SI_NO “encender el aire acondicionado”.

Para no recargar el gráfico, se han omitido las acciones (o eventos) necesarios para ejecutar los eventos “encender luz dormitorio” y “encender luz pasillo”.

Esta estructura facilita también la adición de nuevos elementos. Si, por ejemplo, se añade una nueva lámpara en el baño, basta con asignarle una acción, y añadirla al evento “encender luz baño 2”. No es necesario modificar “encender luces piso 2”

Diseño actual

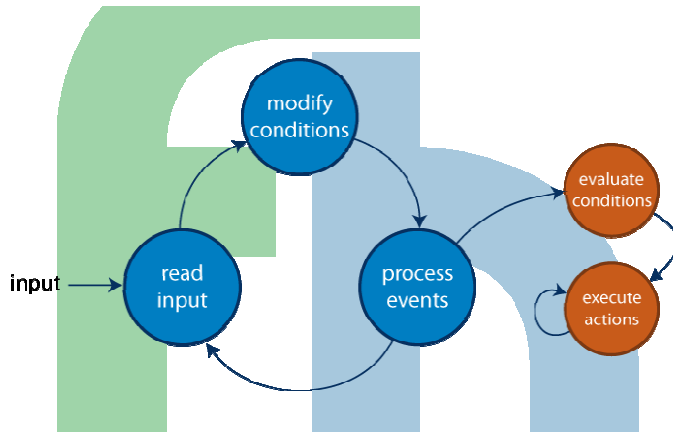
Este diseño se mantuvo durante un tiempo, en el que se utilizaba SQLite como soporte para la base de datos. La falta de espacio en las últimas revisiones del firmware de La Fonera no obligó descartar SQLite y utilizar el sistema de archivos de configuración de La Fonera, UCI. Este sistema no ofrece la funcionalidad de SQLite (por ejemplo, no permite operaciones en cascada ni mantiene la integridad referencial). Dado que es mucho trabajo implementar un sistema de almacenamiento de datos con estas características, y no era el principal objetivo del proyecto, se limitó el diseño, quitando la recursividad. Al quitar la recursividad, algunas partes

del diseño dejaron de tener sentido, por lo que finalmente la base de datos actual consta de dispositivos, acciones y disparadores. Los eventos desaparecieron, y las condiciones se introdujeron en las acciones.

Como trabajo futuro se podría implementar un paquete que dé mayor funcionalidad al sistema de archivos UCI.

Diseño del diagrama de estados del módulo FH

El módulo FH del diagrama general se comporta como un demonio, que itera entre los estados pintados en color azul en el dibujo:



1. Se lee la entrada (idea: debería haber una cola en la entrada, por si vienen varios eventos al mismo tiempo).
2. Se modifica la tabla de condiciones según la entrada.

3. Se procesa cada evento. El procesado consiste en lo siguiente: Para cada evento:
 - a. Se comprueban las condiciones del evento. Si se cumplen las condiciones del evento, entonces se añade la acción correspondiente a una cola de ejecución.
 - b. Cuando termina de evaluar todos los eventos, se ejecutan todas las acciones que se hayan encolado. (En la implementación las acciones no se encolan, sino que se lanza un hilo para ejecutar cada una).

Base de datos

Introducción

Desde el primer momento que empezamos a pensar en este proyecto, sabíamos que necesitábamos algún modo de almacenar nuestros datos de manera persistente, para ello, nada mejor que usar una base de datos.

Después de investigar durante un tiempo, decidimos que la base de datos que más se adaptaba a nuestras necesidades era SQLite, debido a su pequeño tamaño y a sus múltiples posibilidades.

SQLite

Elegimos la versión SQLite3 v3.4.2-1, ya que era la última versión disponible que encontramos para La Fonera, mas información en <http://ipkg.nslu2-linux.org/feeds/openwrt/fon-ng/mips/>, aparte de esto también era necesario bajarse la librería correspondiente a SQLite3, la cual era "libsqlite3 v3.4.2-1".

Una vez instalados estos dos paquetes, ya pudimos proceder a la implementación de las tablas que íbamos a utilizar.



Formato de las tablas de SQLite

Las tablas que en principio íbamos a utilizar en la base de datos fueron las siguientes:

1. Usuarios
2. Dispositivos
3. Eventos
4. Acciones
5. Iconos

Dentro de estas tablas teníamos los siguientes campos:

TABLA USUARIOS

- ID clave principal
- Nombre
- Password
- Permisos

TABLA DISPOSITIVOS

- ID clave principal
- Código de casa
- Código de unidad
- Nombre

TABLA EVENTOS

- ID clave principal
- Nombre
- Nombre_accion_ref

TABLA ACCIONES

- ID clave principal
- Nombre
- Nombre_evento_ref
- Permiso

TABLA ICONOS

- ID clave principal
- Tipo
- Nombre
- Imagen

TABLA ESTADO

- ID dispositivo clave principal
- Nombre
- Estado o parámetros

Una vez establecidas estas tablas, pasamos a su implementación.

Cambio de SQLite a UCI

A medida que íbamos avanzando en el proyecto, tuvimos un gran problema, no teníamos suficiente espacio disponible en La Fonera como para poder instalar la aplicación que estábamos desarrollando.

Probamos varias opciones para disminuir el tamaño de nuestra aplicación y tuvimos que optar por utilizar otro sistema diferente a SQLite ya que, aunque el tamaño de éste era pequeño, La Fonera dispone de muy poco espacio libre para la instalación de aplicaciones.

Hemos optado por utilizar el mismo sistema que usa La Fonera para guardar archivos de configuración. Se basa en archivos de texto planos. LuCI tiene herramientas para su manipulación tanto desde scripts Lua como desde consola de comandos por lo que nos ahorramos espacio al no tener que añadir soporte extra para su uso.

UCI

Las posibilidades que UCI nos ofrecía no eran muy amplias pero suficientes como para poder crear una pequeña base de datos en La Fonera.

Comandos UCI

Las opciones que UCI permite son:

Uso: uci [<opciones>] <comando> [<argumentos>]

Comandos:

```
batch
export      [<config>]
import      [<config>]
changes     [<config>]
commit      [<config>]
add         <config> <section-type>
add_list    <config>.<section>.<option>=<string>
show        [<config>[.<section>[.<option>]]]
get         <config>.<section>[.<option>]
set         <config>.<section>[.<option>]=<value>
delete      <config>[.<section>[.<option>]]
rename      <config>.<section>[.<option>]=<name>
revert      <config>[.<section>[.<option>]]
```

Opciones:

```
-c <path>  asigna el directorio de búsqueda de archivos de configuración (por defecto: /etc/config)
```



- d <str> asigna el delimitador para listas de valores para uci show.
- f <file> usa <file> como entrada en vez de stdin.
- m al importar, mezcla los datos en un paquete existente.
- n nombrar secciones sin nombre al exportar (por defecto)
- N no nombrar secciones sin nombre.
- p <path> añade un directorio de búsqueda de cambios en archivos de configuración.
- P <path> añade un directorio de búsqueda de cambios en archivos de configuración por defecto.
- q modo silencioso (no muestra mensajes de error)
- s forzar modo estricto (parar en errores del analizador sintáctico, por defecto)
- S deshabilitar modo estricto
- X no usar sintaxis extendida con 'show'

De todas estas opciones y comandos a nosotros nos interesaban las siguientes:

- get: devuelve un dato.
- set: modifica un dato.
- delete: borra un campo.
- commit: Hace visibles los cambios realizados sobre un archivo.

Con estas 4 opciones conseguimos crear archivos con un formato muy sencillo de la siguiente manera:

En el shell de La Fonera ponemos lo siguiente (recordar que el directorio por defecto que utiliza UCI es /etc/config/nombre_archivo):

Get-> Devuelve el dato que nosotros le pidamos.

```
uci get fh_nombre_tabla.id.option
```

Por ejemplo:

```
uci get fh_devices.1.name
```

Set-> Modifica el dato que nosotros le digamos.

```
uci set fh_nombre_tabla.id.option=Nuevo_dato
```

Por ejemplo:

```
uci set fh_devices.1.name=prueba
```

Delete->Elimina el campo que nosotros le digamos.

```
uci delete fh_nombre_tabla.id.option
```

Por ejemplo:

```
uci delete fh_devices.1.name
```

Commit-> Hace visible los cambios realizados.

```
uci commit fh_nombre_tabla
```

Por ejemplo:

```
uci commit fh_devices
```

Cuando queremos insertar mediante la opción **set** un dato que tiene espacios, debemos meter este dato entre “ ”, ya que si se hace UCI cogerá el dato hasta el primer espacio, no incluyendo por tanto todo lo que queremos guardar. Ejemplo:

```
uci set fh_devices.1.name="ejemplo de prueba"
uci commit fh-devices
```

Formato de las tablas de UCI

El formato definitivo de las tablas que al final utilizamos fue el siguiente:

Tablas:

1. Dispositivos
2. Disparadores
3. Acciones
4. Iconos
5. Tipos de estados
6. Tipos de disparadores
7. Registro

Dentro de estas tablas tenemos lo siguiente:

TABLA DISPOSITIVOS

- Nombre
- Tipo
- Código de casa
- Código de unidad
- Estado actual
- Var (información adicional)

TABLA DISPARADORES

- Nombre
- Tipo
- Disparador
- Acciones

TABLA ACCIONES

- Nombre
- Permiso requerido



- Condiciones
- Acción si cierto
- Acción si falso

TABLA ICONOS

- Ruta

TABLA TIPOS DE ESTADOS

- Tipo
- Posibles estados

TABLA TIPOS DE DISPARADORES

- Tipo
- Formato de disparador

TABLA REG

- Nombre de lo ocurrido
- Fecha

Ejemplo de archivos UCI

Aquí se muestra un ejemplo del aspecto que tiene un archivo de configuración utilizado por FonHome, en este caso se muestra la tabla de dispositivos denominada “fh_devices”:

```
config 'n_devices' 'n_devices'  
  option 'total' '1'  
config 'device' '1'  
  option 'name' 'Flexo'  
  option 'type' 'X10 appliance'  
  option 'house_code' 'A'  
  option 'unit_code' '1'  
  option 'actual_state' '0'  
  option 'var' '5'
```

En este ejemplo, se pueden observar:

1. Hay que diferenciar primero entre “config” y “option”. “config” se refiere a cada una de las entradas que tiene el fichero UCI y “option” son cada uno de los campos de los que está compuesta cada entrada del fichero.
2. La primera entrada del archivo denominada “n_devices” es una entrada de la tabla donde se almacena, en la variable total, el número total de entradas que tiene la tabla.
3. La segunda entrada del archivo, “device” 1, es el dispositivo identificado como 1, siendo su **id** 1.

Aplicación para La Fonera

Características

La aplicación para La Fonera se ha desarrollado teniendo en cuenta sus limitados recursos, sobre todo de espacio de almacenamiento.

Se han seguido todas las pautas dadas por FON para el desarrollo de aplicaciones para La Fonera en la Fonosfera, utilizando HTML junto con Lua para hacer las páginas dinámicas y Javascript para hacerla interactiva. Para lanzar la aplicación se utiliza la herramienta Fonstated (ver <http://trac.fonosfera.org/fon-ng/wiki/fonstated>).

Se ha tratado de reutilizar todas las librerías que utiliza FON en la interfaz, desde las librerías de LuCI hasta la barra de progreso en Javascript que FON utiliza en la página de la instalación de aplicaciones. La hemos utilizado en la página de control de dispositivos para mostrar el nivel de intensidad. Las distintas páginas se ha preferido que fuesen simples y ligeras a completas y pesadas.

El servidor web del que dispone La Fonera, lucittp, está preparado para servir las páginas basándose en scripts Lua. Nos permite ejecutar comandos u otros scripts al cargar una página. Así conseguimos comunicar la aplicación con el proceso que se ejecuta en segundo plano.

La aplicación está traducida completamente a español e inglés y está preparada para ser traducida a cualquier idioma mediante el uso de archivos de idioma para la librería de traducción de LuCI llamada i18n. También se ha tenido en cuenta que hay países en los que la semana empieza en lunes y otros en domingo, por ello se permite elegir uno u otro en la sección “Configuración”.

Como medida de seguridad, para acceder a la gestión de dispositivos y a la creación/restauración de copias de seguridad se pide al usuario que se identifique mediante introducción de la contraseña.

La carga y descarga de copias de seguridad no es sencilla. El servidor lucittp no permite el envío de archivos que no estén en el directorio /www y no está pensado para cargar archivos. El archivo de copia de seguridad se crea en el directorio /tmp y gracias a la librería de LuCI “ltn12” se permite su descarga. Para la carga se ha utilizado la función “setFileHandler” de la librería “http”.

Estructura

A continuación se describe la estructura de directorios y archivos que tiene la aplicación para La Fonera siguiendo las especificaciones de la Fonosfera (<http://trac.fonosfera.org/fon-ng/wiki/package>).

La estructura siguiente contiene los siguientes paquetes para OpenWRT:

- [setserial 2.17-1 mips.ipk](#)
- [heyu 2.3.2-1 mips.ipk](#): depende de setserial.
- [kmod-usb-serial-pl2303 2.6.26.2-fonera2-1 mips.ipk](#): necesario para utilizar el controlador CM11, que utiliza un conversor pl2303 para pasar de RS232 a USB.
- [libstdc++ 4.1.2-13 mips.ipk](#): librerías de C++.

Los siguientes paquetes se eliminaron al dejar de utilizar SQLite3:

- [libncurses 5.6-1 mips.ipk](#)
- [libreadline 5.2-1 mips.ipk](#)
- [sqlite3-cli 3.4.2-1 mips.ipk](#): depende de libncurses y libreadline.
- [libsqlite3 3.4.2-1 mips.ipk](#): librería de sqlite3.

Directorio raíz de la aplicación:

- **upgrade**: información sobre la instalación de la aplicación.
- **files/**: directorio a copiar sobre el directorio raíz de La Fonera al instalar la aplicación.

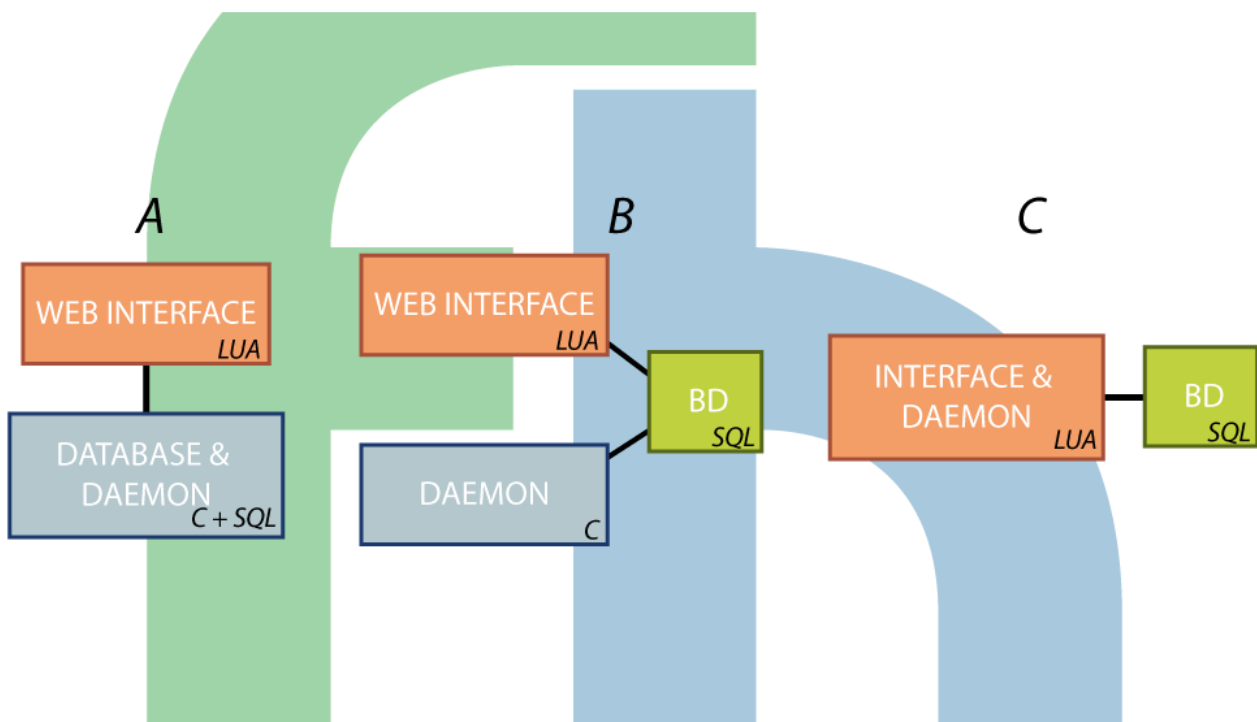


- **etc/**
 - **config/**: ubicación de los archivos de configuración UCI.
 - **fh_actions**: lista de acciones disponibles.
 - **fh_config**: opciones de FonHome.
 - **fh_devices**: lista de dispositivos.
 - **fh_icons**: lista de iconos (no implementado).
 - **fh_reg**: lista del registro de eventos.
 - **fh_triggers**: lista de acciones programadas
 - **fh_types_states**: define los posibles estados que pueden tomar los dispositivos según su tipo.
 - **fh_types_triggers**: define los tipos de programaciones disponibles.
 - **fh_users**: lista de usuarios (no implementado).
 - **pl2303**: archivo de configuración del conversor USB-serie.
 - **fonstated/StartFonhome**: script para arrancar el servicio.
 - **heyu/x10.conf**: archivo de configuración de Heyu.
 - **hotplug.d/ttyusb/pl2303**: script con las acciones a ejecutar al detectar el conversor PL2303.
 - **init.d/setserial**
 - **modules.d/65-usb-serial-pl2303**
- **lib/**
 - **modules/2.6.26.2/pl2303.ko**
 - **libstdc++.so.6(.0.8)**
- **usr/**
 - **bin/**
 - **fonhome**: binario de la aplicación.
 - **heyu**: binario de Heyu.
 - **lib/lua/luci/**
 - **controller/fonhome**
 - **backup.lua**: declaración y funciones Lua de la página de copia de seguridad.
 - **config.lua**: declaración y funciones Lua de la pagina de opciones.
 - **control.lua**: declaración y funciones Lua de la página de control de dispositivos.
 - **devices.lua**: declaración y funciones Lua de la página de dispositivos.
 - **fh_iphone.lua**: declaración y funciones Lua de la página para iPhone.
 - **fh_pl2303.lua**: crea el icono del conversor pl2303 en la página principal de La Fonera.
 - **fonhome.lua**: declaración y funciones Lua de la página principal.
 - **log.lua**: declaración y funciones Lua de la página del registro.
 - **timer.lua**: declaración y funciones Lua de la página del registro.
 - **i18n/**: Archivos de idiomas.
 - **fonhome.en.lua**: inglés.
 - **fonhome.es.lua**: español.

- **model/cbi/fonhome/**
 - **pl2303.lua**
- **view/fonhome/**
 - **backup.htm**: plantilla para la copia de seguridad.
 - **config.htm**: plantilla para las opciones.
 - **control.htm**: plantilla para el control de dispositivos.
 - **devices.htm**: plantilla para la configuración de dispositivos.
 - **fh_iphone.htm**: plantilla para la interfaz para iPhone.
 - **fh_running.htm**: plantilla para ver si se está ejecutando nuestro servicio.
 - **log.htm**: plantilla para el registro.
 - **timer.htm**: plantilla para el programador.
- **sendFH.lua**: librería en Lua para la comunicación con el servicio.
 - **sbin/setserial**
- **www/luci-static/resources/**
 - **fonhome/**
 - **img/**: imágenes de la plantilla para iPhone.
 - **res/**: javascript y estilo de la plantilla para iPhone.
 - **...**: javascript e imágenes de la interfaz web.
 - **icons/fonhome.png**: icono de la aplicación del panel de control de La Fonera.

Comunicación módulo FH – aplicación

Diseño





Una vez hecho el diseño, planteamos distintas formas de implementación e interconexión del sistema al foro de desarrollo. En este momento todavía se pensaba hacer la base de datos en SQL (que posteriormente se hizo en UCI). Se optó por el diseño B para facilitar la portabilidad y la ampliación.

La comunicación entre el interfaz (programado en Lua) y el proceso fonhome (programado en C++) se realiza mediante IPC (*Interprocess Communication*). Se basa en un pipe o tubería para el paso de mensajes.

Características generales

Longitud de los mensajes: 128 bytes máximo

Estructura general: **[PRE][MENSAJE...]**

Donde PRE es un prefijo de 3 chars que indica el tipo de mensaje, y MENSAJE, de hasta 125 bytes, es el mensaje. Así por ejemplo tenemos mensajes de X10 como este: **[X10][off.A01]**, o mensajes para añadir elementos a la BD como este: **[ADD][ev.nombre_evento]**

Mensajes X10

En cuanto a los mensajes de X10, la estructura es la siguiente:

[X10][ope][idn][Dxx][vlr(op)]

Donde ope es la operación, Dxx es el dispositivo sobre el que actuar, idn es el identificador del dispositivo en la base de datos, y vlr, que sólo aparece para dim y bright, es el valor de la operación. Todos son de 3 bytes. Un ejemplo sería el siguiente (sin los espacios): X10 DIM 004 050 001

Los valores de ope son los siguientes:

Encender	onn (tiene que tener 3 bytes)
Apagar	off
Subir	bri
Bajar	dim

Mensajes de configuración

En cuanto a los mensajes de configuración, su estructura es la siguiente:

[CNF][tipo][oper][mensaje]

El tipo puede ser:

Dispositivo	Dvc
Disparador	Trg
Acción	Act
Icono	lcn

Las operaciones pueden ser:

Añadir	Add
Borrar	Del
Modificar	Set

Los mensajes son variables dependiendo del tipo y de la operación:

Dispositivo:

Add	nombre,tipo,house_code,unit_code
Set	número,nombre,tipo,house_cod,unit_cod

Disparador:

Add	nombre,tipo,trigger,acciones
Set	número,nombre,tipo,trigger,acciones

Del	número
Acción:	
Add	nombre,tipo,condicion,accion_true,accion_false
Set	numero,nombre,tipo,condicion,accion_true,accion_false
Del	número

Del	número
Icon:	
Add	Ruta
Set	número,nueva_ruta
Del	Número

Un ejemplo de añadir sería el siguiente:

CNF Dvc Add L234,lamp,A,01

Recordar que cada uno de los campos en el mensaje van separados por comas y estos deben ser tipo "string" a la hora de llamar a la BD.

Un ejemplo de borrar sería el siguiente:

CNF Dvc Del 001

Un ejemplo de modificar sería el siguiente:

CNF Dvc Set 001,L234,lampx10,A,01

DESARROLLO

Planificación

Fecha	Hito
Junio de 2008	<p>Proponemos el proyecto a Juan Carlos. Nos propuso utilizar La Fonera.</p> <p>Empezamos a investigar sobre domótica. Las opciones que se barajaban entonces eran implementar el sistema sobre Android y utilizar Bluetooth. Todavía no sabíamos qué tecnología domótica utilizar.</p> <p>Abrimos el grupo de trabajo en Google: http://groups.google.com/group/si0809</p> <p>Abrimos el repositorio de código en Google: http://code.google.com/p/si0809/</p>
Agosto de 2008	Nos decidimos por X10 sobre otras tecnologías como ZigBee, etc., por ser más barata y más extendida.
Octubre de 2008	Compramos un módulo X10 y un controlador además de una La Fonera+. Juan Carlos nos propone utilizar una FPGA para implementar el controlador. Se descartó Bluetooth por su corto alcance y por ser difícil de integrar en La Fonera.



Hasta febrero de 2009	<p>Se hizo el diseño y se decidió qué íbamos a usar para programar.</p> <p>Recibimos 3 La Fonera 2.0 por cortesía de FON.</p> <p>Se compraron controladores y módulos X10 y se descartó utilizar la FPGA.</p> <p>Se hizo un script en CGI y se colgó un video en YouTube.</p> <p>Se abrió la página web y se publicaron posts en los blogs de FON.</p> <p>Se investigó cómo hacer la aplicación para La Fonera.</p>
Hasta mayo de 2009	<p>FON quitó CGI, y se tuvo que cambiar el diseño.</p> <p>Se hizo una implementación de la base de datos en SQLite.</p> <p>Después de 8 cambios de firmware, nos quedamos con la versión Platero (v2.2.4.0).</p> <p>Cambio de SQLite a UCI (con Platero)</p> <p>Implementación, prototipos, etc.</p>
Junio de 2009	<p>Pruebas y más pruebas.</p> <p>Segundo post de FON.</p> <p>Memoria y presentación.</p>

Casos de uso – Trabajo futuro

Esta es una colección de ideas que se tuvieron al principio del desarrollo, mientras se hacía el diseño. Con el grado de implementación conseguido no se pueden realizar todas ellas, pero se tuvieron siempre en mente para que el proyecto se pudiera ampliar y convertir estos casos de uso en realidad.

Videoportero en el móvil

Aprovechando las funcionalidades que ofrece La Fonera, y con una simple webcam, se podría implementar un sistema de videoportero, con la comodidad de tenerlo disponible en el teléfono móvil o en el PC a través de la interfaz web. Las ventajas van desde abrir la puerta desde el sofá, identificando a la persona que ha llamado a través de la cámara, a abrir la puerta remotamente, perfecto si esperamos el envío del supermercado (y nos fiamos mucho del repartidor), o si algún familiar (o uno mismo) se ha dejado las llaves en casa.

Control por email

Otro ejemplo de domótica aplicada es el siguiente: si enviamos un email con el asunto “Activar” a una dirección de correo determinada, se encienden las luces de la casa, el equipo de música, y se abren las persianas, simulando la presencia de personas.

Control desde otras fuentes

Aprovechando la conexión a Internet de La Fonera, sería fácil implementar sistemas externos de control. Por ejemplo, se podría utilizar un calendario, como por ejemplo los que ofrece Google de forma gratuita, para programar distintos eventos en la casa. También podrían utilizarse feeds RSS de páginas de pronóstico meteorológico para ajustar la climatización de la casa con antelación suficiente, o subir los toldos en caso de haber viento (añadiendo la instalación de un anemómetro en el sistema).

Gestión inteligente de temperatura

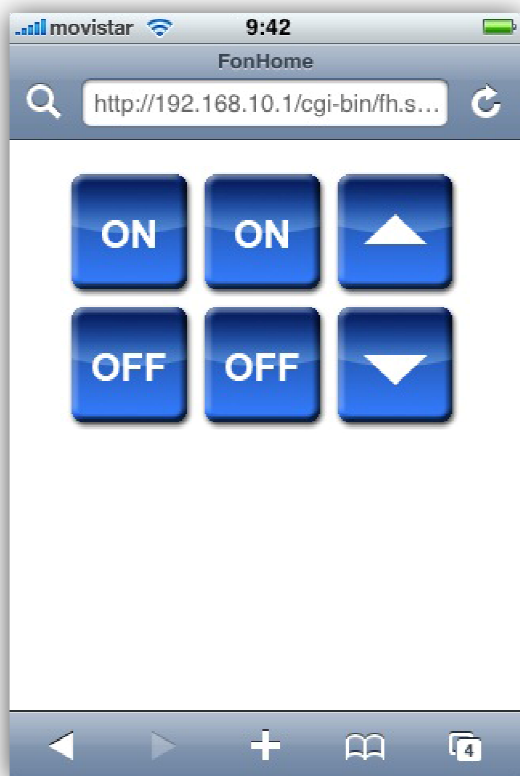
El sistema podría ofrecer al usuario un gráfico de la temperatura en la casa y en el exterior, y además podría preguntarle si está cómodo o no, ajustando la temperatura automáticamente, sin que el usuario tenga que saber qué temperatura es la más apropiada en cada momento. Con la inteligencia artificial apropiada, el sistema podría aprender las costumbres del usuario. Además, podría comprobar qué otros elementos de la vivienda afectan a la temperatura (toldos, persianas) e incluirlos en el control de temperatura.

Gestión inteligente de iluminación

Con la estructura apropiada, podría existir una instrucción del tipo "encender luces de camino al dormitorio principal", lo que encendería sucesivamente las luces del pasillo, la escalera, el distribuidor del piso de arriba, y del dormitorio (e incluso las iría apagando, también sucesivamente, pasado un intervalo de tiempo). Estos patrones podrían ser aprendidos por el sistema, aplicando un sistema de inteligencia artificial apropiado.

EJEMPLOS DE EJECUCIÓN

En este apartado se pueden ver algunos ejemplos de ejecución. Para ver en detalle el funcionamiento de la versión final, consulte el apartado *Manual* en la página 46.



La foto de la izquierda corresponde a la primera prueba de interfaz que se hizo, y la que apareció en el primer video que se publicó sobre el proyecto.

Simplemente era una colección de botones que controlaban un dispositivo, usando un script CGI (Common Gateway Interface) que se ejecutaba sobre La Fonera. Este script recibía las instrucciones desde la URL.

Dado que FON retiró el intérprete de CGI que existía cuando hicimos este interfaz, se abandonó el prototipo.

Modos de ejecución para depuración

Existen 5 modos de ejecución:

- Modo depuración, la orden para ejecutar en este modo es `"fonhome --debug"`
- Modo depuración base de datos, la orden para ejecutar en este modo es `"fohome --debug_bd"`
- Modo sin monitor, la orden para ejecutar en este modo es `"fohome --nomonitor"`
- Modo ruta de Heyu, la orden para ejecutar en este modo es `"fohome --heyupath ruta_Heyu"`
- Modo por defecto, la orden para ejecutar en este modo es `"fonhome"`

Modo depuración

Con este modo de ejecución se puede ver a través del shell todo lo que se está ejecutando en fonhome.

```

root@Fonera:~# fonhome --debug

FonHome ver. 0.01

- Debug mode activated.
Loading...

DB: Getting actions
DB: Getting devices
DB: Getting triggers
Starting Heyu relay. Please wait...
Ready! Press ENTER to exit

1. IPC says: X10off001
2. Executing off C01
3. HEYU sent: off C1
4. IPC says: X10onn001
5. Executing on C01
6. HEYU sent: on C1
7. "party mood on" triggered with "Party Lights on"!
   Executing action "Lampara cama on" from trigger "party mood on"
8. No condition on action Lampara cama on
9. HEYU sent: on A2

```

Modo depuración de base de datos

Con este modo se puede ver todo lo que ocurre cuando se producen cambios en la BD.

```

root@Fonera:~# fonhome --debug_bd

FonHome ver. 0.01

- Debug BD activated.
Loading...

DB: Getting actions
DB: Getting devices
DB: Getting triggers
GETTING NUM_ACTIONS_TOTAL
GETTING NUM_DEVICES_TOTAL
GETTING NUM_TRIGGERS_TOTAL
GETTING NUM_REG_TOTAL
Starting Heyu relay. Please wait...
Analizando mensaje...
ADDING TRIGGER IN THE BD..
TRIGGER ADDED
ADDING CHANGES IN THE REG
CHANGES ADDED IN THE REG
ADDING TRIGGER IN THE LIST OBJECT
TRIGGER ADDED IN THE LIST OBJECT
//////////LISTA DE TRIGGERS//////////

Objeto trigger 1
Name: party on
Type: X10
trigger: 1
Actions: 5
//////////
//////////FIN LISTA TRIGGERS//////////

```



Modo sin monitor

Este modo deshabilita la monitorización. El sistema funciona igual, pero no recibe entradas o cambios de estado del sistema X10.

```
root@Fonera:~# fonhome --nomonitor
FonHome ver. 0.01
- Monitor is disabled.
Loading...
Ready! Press ENTER to exit
```

Modo ruta de Heyu

Con este modo se puede cambiar el path de Heyu para fonhome.

```
root@Fonera:~# fonhome --heyupath /usr/bin/heyu
FonHome ver. 0.01
- Heyu path is now /usr/bin/heyu.
Loading...
Starting Heyu relay. Please wait...
```

Modo por defecto

```
root@Fonera:~# fonhome
FonHome ver. 0.01
Loading...
Starting Heyu relay. Please wait...
```

Hay que aclarar que el modo de ejecución por defecto es el modo que se ha utilizado para incluir la aplicación FonHome como servicio en “La Fonera”.

PROMOCIÓN DEL PROYECTO

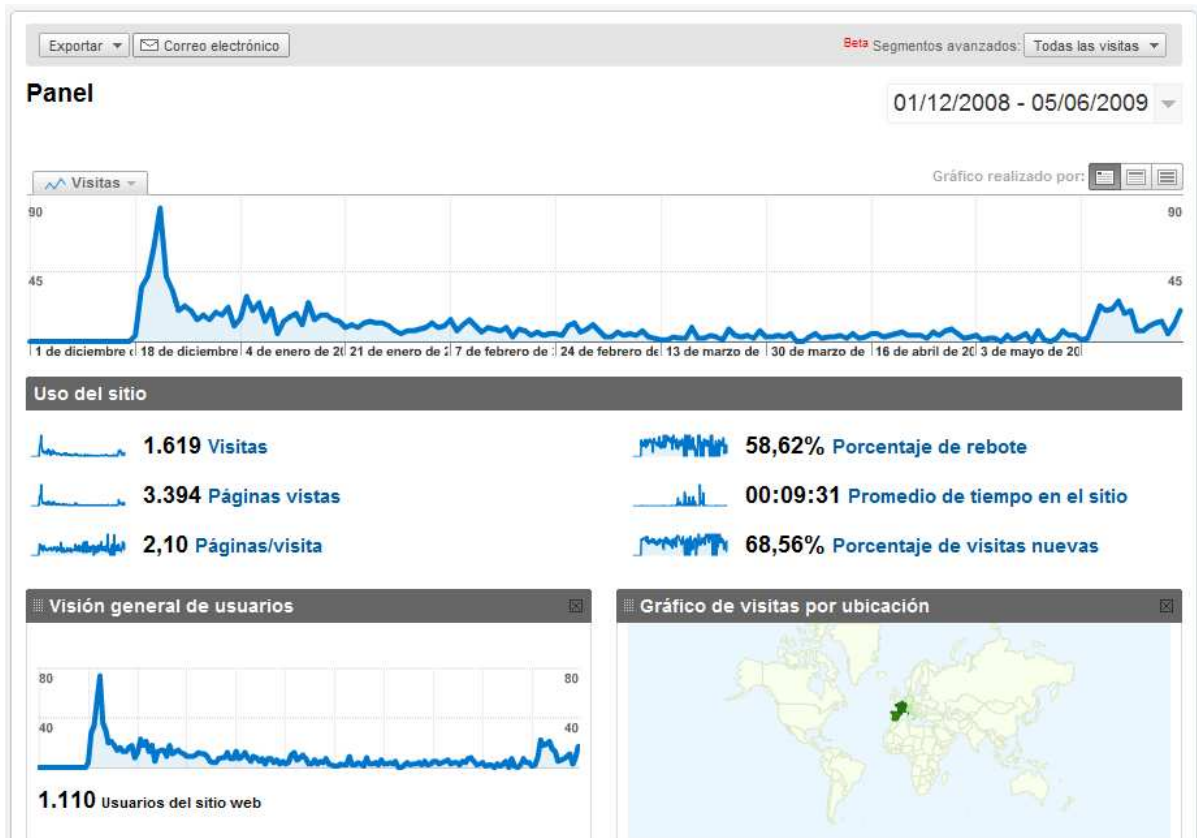


Como ya se ha comentado, desde el principio se intentó hacer el proyecto atractivo para la comunidad FON.

En diciembre de 2008 se registró el dominio www.fonhome.com, y se abrió una página web informativa.

Esta página, disponible completamente tanto en inglés como en español, diseñada con Joomla, mantiene informados a los usuarios de FonHome sobre las novedades que se van produciendo.

A fecha de la edición de esta memoria se han recibido 1619 visitas en esta página, con 3394 páginas vistas.



Estadística de visitas de www.fonhome.com



En esta página, y en los blogs de FON, se publicaron diversos artículos sobre nuestro proyecto, el más importante de todos el que incluyó el primer video que se grabó sobre FonHome. En el video se puede ver una demostración de FonHome con un interfaz provisional que se hizo en CGI (que después se desechó por cambios en el firmware de La Fonera):

FonHome first test

there are no computers in the system.

1:08 / 1:48

AVISO Este vídeo contiene una pista de audio que no ha sido autorizada por WMG. El audio se ha desactivado. Más información sobre copyright

★★★★★ 5 puntuaciones 5122 reproducciones

Favoritos Compartir Listas de reproducción Marcar

Enviar video Tuenti Facebook más

FonHomeProject
17 de diciembre de 2008
(más información)

This is the first test of the FonHome project. For more information, please visit us at www.fonhome.com "Fon" and "La Fonera" are registered trademarks of Fon Wireless, Ltd.

URL
<http://www.youtube.com/watch?v=7FpLUgxexhw>

Insertar
<object width="425" height="344"><param name="n

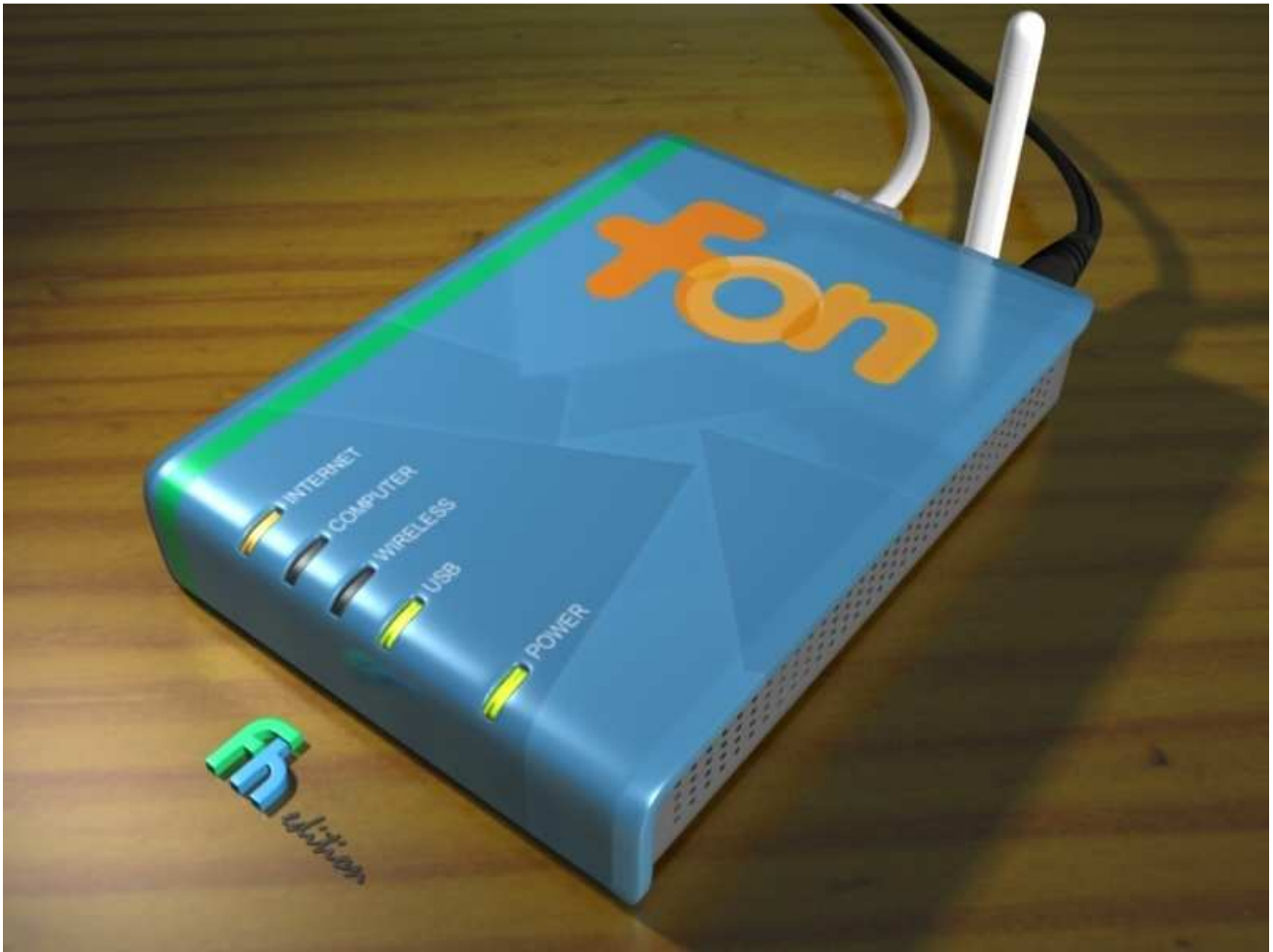
► Más de: FonHomeProject

▼ Vídeos relacionados

- Domótica con la fonera 4598 reproducciones sistemasorp 2:43
- Hacking Is FON! 1611 reproducciones neivallon 6:15
- LaComunatv - Adelanto villancico 384 reproducciones lacomunatv 0:15
- LCD on Fonera by liberailvoip.it 7444 reproducciones

Este video ha recibido a fecha de la edición de esta memoria 5265 reproducciones.

Recientemente se propuso a FON hacer un concurso de diseño de Foneras, y se envió una posible propuesta de FonHome, La Fonera FonHome edition:





CONCLUSIONES

Hemos realizado un proyecto basado en el diseño e implementación de una aplicación capaz de gestionar dispositivos domóticos. Este proyecto nos ha parecido muy interesante por diferentes razones:

- Pueden facilitar la vida de la personas.
- Es una tecnología poco extendida en España.
- Este tipo de proyectos, situados en el marco del software libre, permiten que la domótica pueda llegar a muchos hogares debido a su menor coste.

Aunque parezca obvio, nos hemos dado cuenta de que no es muy recomendable realizar un proyecto sobre hardware o software en tempranas fases de desarrollo.

Una aplicación de esta índole “nunca se acaba”, siempre será posible mejorar la aplicación, ya sea por la utilización de nuevas tecnologías domóticas como por la utilización de distintas herramientas de programación. Bajo esta perspectiva, esperamos haber conseguido una aplicación altamente modular, de forma que el proyecto quede abierto a posibles ampliaciones de una manera sencilla.

Como conclusión final, podemos decir que estamos muy contentos de haber tenido la oportunidad de investigar en un proyecto como este y esperamos que, gracias a aplicaciones como FonHome, cada día haya más hogares que puedan disfrutar de los beneficios que la domótica puede aportar.

INSTALACIÓN

Instrucciones para instalar la aplicación

Prototipo CGI

Es necesario disponer de la versión de firmware para desarrolladores para poder conectarse por SSH a La Fonera.

Instalar paquetes necesarios

Para instalar los paquetes necesarios, hace falta ejecutar los siguientes comandos:

```
opkg install http://downloads.openwrt.org/snapshots/atheros/packages/setserial_2.17-1_mips.ipk
opkg install http://downloads.openwrt.org/snapshots/atheros/packages/heyu_2.3.2-1_mips.ipk
opkg install http://ipkg.nslu2-linux.org/feeds/openwrt/fon-ng/mips/kmod-usb-serial-pl2303_2.6.26.2-fonera2-1_mips.ipk
```

Crear archivo de configuración de Heyu

Crear en alguna ubicación el archivo `x10.conf`, por ejemplo, en la carpeta `/fh`. Este archivo indica dónde está el dispositivo del controlador. Para crearlo ejecutar el siguiente comando:

```
echo "TTY /dev/ttyUSB0" > /fh/x10.conf
```

Copiar el script CGI del interfaz

Dentro de este fichero comprimido hay unos archivos de imagen, que deben copiarse en la carpeta `/www/luci-static/resources/icons`, y un archivo `.sh` que debe copiarse en la carpeta `/www/cgi-bin/` con permisos de ejecución.

```
chmod +x /www/cgi-bin/fh.sh
```

Acceder al interfaz

Desde un navegador web con acceso a La Fonera, acceder a la dirección de la puerta enlace, `http://192.168.x.1/cgi-bin/fh.sh` (por defecto `x=10`).

Habilitar acceso al interfaz desde internet

Para acceder al interfaz desde la WAN de La Fonera, primero hay que abrir el acceso con el siguiente comando:

```
iptables -A input_daemon -p tcp --dport 80 -j zone_wan_ACCEPT
```

Para bloquear el acceso, el comando es:

```
iptables -A input_daemon -p tcp --dport 80 -j zone_wan_REJECT
```

Este comando se anula al reiniciar La Fonera. Para hacer el cambio permanente, ver las instrucciones aquí: http://wiki.fon.com/wiki/F2_MgtConsole_tweaks#allow_access_from_WAN_to_LAN.2FMyplace.

Post sobre el prototipo con CGI en los blogs de FON

Blog de la Fonosfera: <http://blog.fonosfera.org/la-fonera-20-a-home-automation-device/>

Blog de FON en español: <http://blog.fon.com/es/archivo/fonera-20/fonhome-domatica-con-la-fonera-20.html>



Aplicación para La Fonera

Para instalar la aplicación en La Fonera se deben cumplir los siguientes requisitos:

- Disponer de una versión de firmware igual o superior a la 2.2.4.0 (Platero), no necesariamente versión para desarrolladores.
- Un hub USB 2.0.
- Un controlador CM11.

La instalación es muy sencilla. Basta con seguir estos tres sencillos pasos:

1. En el panel de control de La Fonera pinchar en Aplicaciones
2. En el apartado "Fichero de aplicación FON" hacer click sobre el botón Examinar y seleccionar el archivo ".tar.gz" de la aplicación.
3. Después aparecerá algo parecido a la siguiente imagen:

Aplicación instalada correctamente.

Aplicaciones
Aquí puedes instalar o eliminar aplicaciones

Espacio restante en el dispositivo

Tipo	Libre	Total	Usado
Interno	122.34 KB	1.19 MB	89%

Plugins instalados

Nombre	Versión	Tamaño	Proveedor
FonHome	0.1	1.07 MB	FonHome Team

Se habrán añadido los iconos FonHome y PL2303 al panel de control.

Post sobre la aplicación para La Fonera en los blogs de FON

Blog de la Fonosfera: <http://blog.fonosfera.org/an-update-on-the-fonhome-project/>

MANUAL

Índice

1. [Introducción](#)
2. [Panel de control](#)
3. [FonHome](#)
4. [Control de dispositivos](#)
5. [Gestionar dispositivos](#)
6. [Programar acciones](#)
7. [Registro de eventos](#)
8. [Copia de seguridad](#)
9. [Configuración](#)
10. [FonHome Móvil](#)

Introducción

A continuación se realizará un recorrido por los distintos menús de la aplicación y se explicarán todas las posibilidades que ofrece.

Panel de control

Para utilizar la aplicación lo primero que hay que hacer es conectarse a La Fonera y acceder al panel de control. Debería ver algo muy parecido a lo siguiente (versión de firmware Platero 2.2.4.0):



Una vez hecho esto, lo siguiente que debemos hacer es pulsar sobre el icono FonHome para poder acceder a la aplicación.

FonHome

Éste es el menú de FonHome:



Como se puede observar las posibilidades que se ofrecen son las siguientes:

1. Control de dispositivos
2. Gestionar dispositivos
3. Programar acciones
4. Registro de eventos
5. Copia de seguridad
6. Configuración
7. FonHome :: Interfaz para móvil

De ahora en adelante definiremos estos puntos uno a uno con el máximo detalle posible.

Control de dispositivos

En este apartado se puede probar el funcionamiento de sus dispositivos registrados en el sistema y ver su estado actual.

Nombre	Tipo	C. Casa	C. Dispo.	Estado	Intensidad	Acciones
Pasillo	X10 lamp	A	1	ON		Enviar
Cafetera	X10 appliance	B	1	OFF	<input type="text"/>	Enviar
Riego	X10 appliance	C	1	ON	<input type="text"/>	Enviar
Presencia jardín	X10 m/l sensor	C	2	OFF	<input type="text"/>	Enviar
Luz jardín	X10 m/l sensor	C	3	OFF	<input type="text"/>	Enviar

Existen 3 tipos de dispositivos: "X10 appliance", "X10 lamp", "X10 m/l sensor". Dependiendo del tipo de dispositivo que tenga se podrán realizar unas acciones u otras. A continuación explicamos las posibilidades de cada tipo de dispositivo:

- X10 appliance: módulo X10 para control de aparatos.
 - Con este tipo de dispositivos sólo se permiten las acciones de apagado o encendido del dispositivo conectado, para ello basta con que se pulse sobre el botón situado en la columna de ESTADO (en la fila correspondiente al dispositivo en cuestión) y su estado cambiará de OFF a ON y viceversa.
- X10 lamp: módulo X10 para control de lámparas.
 - Éste tipo de dispositivo tiene varias posibilidades:
 - ON/OFF: Con este tipo de dispositivos se permiten las acciones de apagado o encendido de una lámpara, basta con que pulsemos sobre el botón situado en la columna Estado (en la fila correspondiente al dispositivo en cuestión) y el estado del dispositivo cambiará de ON a OFF y viceversa.
 - Subir o bajar intensidad de lámpara: Con este tipo de dispositivo se puede subir o bajar el nivel de intensidad de una lámpara pulsando sobre los triángulos situados en la columna Intensidad. La barra de intensidad sube o baja en función del botón pulsado. Una vez alcanzada la intensidad deseada hay que pulsar en el botón enviar para que el sistema registre y ejecute la acción.

Recordatorio, no utilizar nunca lámparas de bajo consumo con dispositivo X10 lamp.

- X10 m/l sensor: sensor X10 de movimiento y amanecer/atardecer.
 - Con este tipo de dispositivos se permite simular la recepción de señales de detección de movimiento de un sensor X10, para ello basta con que se pulse sobre el botón situado en la

columna de ESTADO (en la fila correspondiente al dispositivo en cuestión) y su estado cambiará de OFF a ON y viceversa.

Gestionar dispositivos

En este apartado se podrán gestionar todos los dispositivos registrados en el sistema.



Panel de control >> FonHome >> Gestionar dispositivos Logout

Gestionar dispositivos

Aquí puedes gestionar tus dispositivos.

Dispositivos

Nombre	Tipo	C. Casa	C. Dispo.	Acciones
Pasillo	X10 lamp	A	1	 
Cafetera	X10 appliance	B	1	 
<input type="text" value="Riego"/>	<input type="text" value="X10 appliance"/>	<input type="text" value="C"/>	<input type="text" value="1"/>	<input type="button" value="Modificar"/>
Presencia jardín	X10 m/l sensor	C	2	 
Luz jardín	X10 m/l sensor	C	3	 
<input type="text" value="Luz perimetral"/>	<input type="text" value="X10 appliance"/>	<input type="text" value="C"/>	<input type="text" value="4"/>	<input type="button" value="Añadir"/>

Posibles opciones:

- Añadir dispositivo.
- Modificar dispositivo.
- Borrar dispositivo.

AÑADIR DISPOSITIVO

Los pasos a realizar para añadir un dispositivo son los siguientes:

1. En la fila vacía del final de la tabla introducimos nombre, tipo, código de casa y código de dispositivo.
2. Pulsar sobre el botón Añadir.

Una vez introducidos los datos del dispositivo, se añadirán automáticamente dos acciones asociadas a éste, serán:

- Nombre_dispositivo on
- Nombre_dispositivo off

Estas acciones se podrán usar para la programación de acciones en el apartado "Programar acciones".

MODIFICAR DISPOSITIVO

Los pasos a realizar para modificar un dispositivo son los siguientes:

1. Pulsar sobre el icono con forma de lápiz situado en la fila correspondiente al dispositivo que queremos modificar.
2. Modificar los campos que nos interesen.
3. Pulsar botón Modificar.

Al igual que cuando añadimos una acción, si cambiamos el nombre del dispositivo, también cambiarán los nombres de las acciones asociadas a ese dispositivo. Los nuevos nombres de las acciones serán:

- Nuevo_nombre_dispositivo on
- Nuevo_nombre_dispositivo off

BORRAR DISPOSITIVO

Los pasos a realizar para borrar un dispositivo son los siguientes:

1. Pulsar sobre el botón con forma de aspa del dispositivo que quiera eliminar.
2. Confirmar que realmente se quiere borrar el dispositivo.

Programar acciones

En este apartado se podrán programar acciones sobre los dispositivos registrados en el sistema.

Posibles opciones:

- Programar Acción TIME: unos días de la semana a una hora determinada.
- Programar Acción DATE: en una fecha y hora determinada.
- Programar Acción X10: al recibir una señal de un sensor o ejecutar un comando X10.

Cuando se borre un dispositivo también se borrarán las acciones asociadas a ese dispositivo y además todos los disparadores que contuvieran algunas de esas acciones.

PROGRAMAR ACCIÓN TIME



LA FONERA 2.0

Panel de control >> FonHome >> Programar acciones Logout

Programar acciones
Aquí puedes programar acciones.

Acciones programadas

Nombre	Tipo	Disparador	Acción	Eliminar
Riego inicio	TIME	L_X_V_D - 22:00	Riego on	✗
Riego fin	TIME	L_X_V_D - 23:00	Riego off	✗
Movimiento en jardín	X10	Presencia jardín on	Luz perimetral on	✗
Café inicio	TIME	L M X J V _ _ - 06:50	Cafetera on	✗

Formulario de programación:

Café fin | TIME | L M X J V S D | @ 7 : 05 | Añadir

- Pasillo on
- Pasillo off
- Cafetera on
- Cafetera off
- Riego on
- Riego off
- Presencia jardín on
- Presencia jardín off
- Luz jardín on
- Luz jardín off
- Luz perimetral on
- Luz perimetral off

Los pasos a realizar para programar una acción TIME son los siguientes:

1. En la fila vacía del final de la tabla introducimos el nombre.
2. Seleccionamos el tipo TIME.
3. Marcamos los días que queremos que se realice la acción.
4. Seleccionamos la hora.
5. Seleccionamos una acción.
6. Por último pulsamos sobre el botón Añadir.



Los pasos a realizar para programar una acción DATE son los siguientes:

1. En la fila vacía del final de la tabla introducimos el nombre.
2. Seleccionamos el tipo DATE.
3. Seleccionamos la fecha en el calendario mostrado al pulsar sobre cualquiera de los campos de la fecha.
4. Seleccionamos la hora.
5. Seleccionamos la acción a realizar
6. Por último pulsamos sobre el botón Añadir.

PROGRAMAR ACCIÓN X10



The screenshot shows the 'LA FONERA 2.0' interface. At the top, there is a navigation bar with 'Panel de control >> FonHome >> Programar acciones' and a 'Logout' link. Below this is the 'Programar acciones' section, which includes a table of programmed actions and a form to add a new one.

Programar acciones
Aquí puedes programar acciones.

Acciones programadas

Nombre	Tipo	Disparador	Acción	Eliminar
Riego inicio	TIME	L_X_V_D - 22:00	Riego on	X
Riego fin	TIME	L_X_V_D - 23:00	Riego off	X
Movimiento en jardín	X10	Movimiento jardín on	Luz perimetral on	X
Café inicio	TIME	L M X J V __ - 06:50	Cafetera on	X
Café fin	TIME	L M X J V __ - 07:05	Cafetera off	X
Simular presencia 1	DATE	09/06/2009-14:15	Pasillo on	X
Simular presencia 2	DATE	09/06/2009-14:20	Pasillo off	X

Formulario de adición:

Los pasos a realizar para programar una acción X10 son los siguientes:

1. En la fila vacía del final de la tabla introducimos el nombre.
2. Seleccionamos el tipo X10.
3. Seleccionamos un disparador.
4. Seleccionamos una acción.
5. Por último pulsamos sobre el botón Añadir.

Registro de eventos

En este apartado se ve el registro de lo que va ocurriendo en el sistema.

Panel de control >> FonHome >> Registro de eventos Logout

Registro de eventos

Aquí puede ver cuándo han ocurrido eventos.

Registro de eventos

Instante	Evento
09/06/2009-13:55	Acción Añadida: Luz jardín off
09/06/2009-13:55	Acción Añadida: Luz jardín on
09/06/2009-13:55	Dispositivo añadido: Luz jardín
09/06/2009-13:50	Acción Añadida: Movimiento jardín off
09/06/2009-13:50	Acción Añadida: Movimiento jardín on
09/06/2009-13:50	Dispositivo añadido: Movimiento jardín
09/06/2009-13:48	Acción modificada: Cafetera off
09/06/2009-13:48	Acción modificada: Cafetera on
09/06/2009-13:48	Dispositivo modificado: Cafetera
09/06/2009-13:48	Acción Añadida: Cafetera off
09/06/2009-13:48	Acción Añadida: Cafetera on
09/06/2009-13:48	Dispositivo añadido: Cafetera
09/06/2009-13:47	Acción Añadida: Dormitorio off
09/06/2009-13:47	Acción Añadida: Dormitorio on
09/06/2009-13:47	Dispositivo añadido: Dormitorio
09/06/2009-13:47	Acción Añadida: Cocina off
09/06/2009-13:47	Acción Añadida: Cocina on
09/06/2009-13:47	Dispositivo añadido: Cocina
09/06/2009-13:46	Acción Añadida: Salón off
09/06/2009-13:46	Acción Añadida: Salón on

Páginas: [1](#) [2](#)

Si el número de entradas del registro es grande, se crean páginas dinámicamente para que la visualización sea más cómoda. Se muestran primero las entradas más recientes. No es necesario refrescar la página para que muestre las entradas más recientes, lo hace automáticamente cada 30 segundos.

Si pulsa sobre el botón “Borrar el registro”, el registro de eventos se vaciará.

Copia de seguridad

En este apartado se puede hacer una copia de seguridad de los datos de la aplicación o restaurar una existente.



Posibles opciones:

- Crear copia de seguridad.
- Restaurar copia de seguridad.

CREAR COPIA DE SEGURIDAD

Esta opción permite hacer una copia de seguridad de los datos de la aplicación. Lo único que hay que hacer es pulsar sobre el botón Descargar y guardar el fichero “fh_backup.tgz”. Se puede renombrar para distinguirlo de otras copias de seguridad.

RESTAURAR COPIA DE SEGURIDAD

Esta opción permite restaurar una copia de seguridad hecha anteriormente.

Para realizar la restauración basta con pulsar sobre Examinar, buscar la copia de seguridad a restaurar y pulsar el botón Cargar.

Configuración

En este apartado puede configurar las opciones de FonHome.



FONOSFERA
beta

LA FONERA 2.0

Panel de control Estado de la Fonera

Panel de control >> FonHome >> Configuración Logout

Configuración

Aquí puedes configurar FonHome.

- Zona horaria:

Fecha actual: 23/06/2009

Hora actual: 21:37

Zona horaria:

Info: http://en.wikipedia.org/wiki/List_of_time_zones

- Primer día de la semana:

La Fonera viene configurada de fábrica a zona horaria UTC. Al no disponer de pila para mantener la hora al perder la alimentación, La Fonera actualiza la hora a través de internet en cada inicio.

Para que las horas que maneja FonHome sean correctas es necesario configurar la zona horaria a la región en la que se vaya a utilizar.

Según el país, el primer día de la semana es el lunes o el domingo. Se da la opción de elegir uno u otro.

Para que los cambios tengan efecto hay que hacer click sobre el botón Guardar.



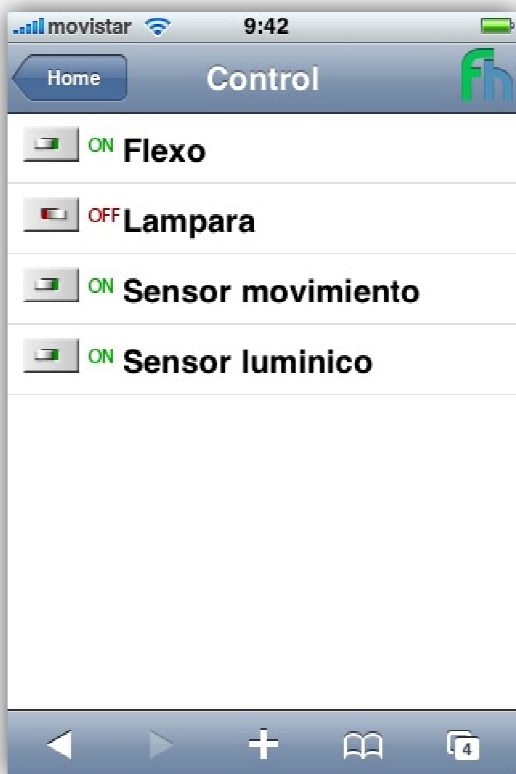
FonHome móvil



A través de este apartado se puede acceder a la aplicación con un dispositivo móvil.

Posibles opciones:

- Control de dispositivos
- Ver registro



CONTROL DE DISPOSITIVOS

Esta opción permite saber cuál es el estado actual de nuestros dispositivos y modificar su estado. Para ello basta con seleccionar la opción del menú principal "Control" y después pulsar sobre aquellos dispositivos que queramos cambiar de estado.



REGISTRO

En este apartado se puede ver el registro de acciones del sistema domótico.

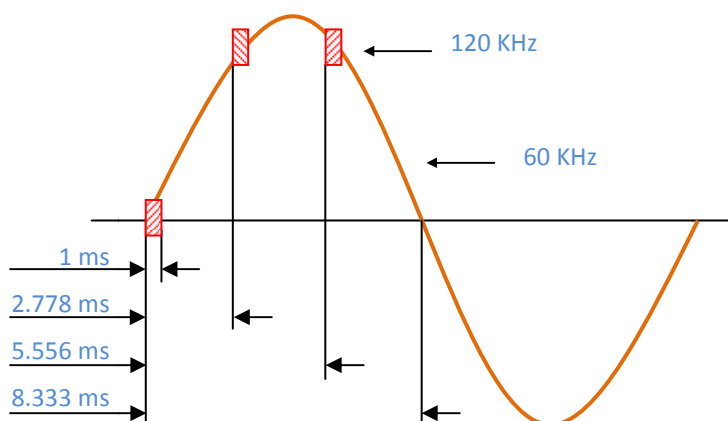


ANEXOS

Anexo 1. Protocolo X10

X10 es un protocolo de comunicaciones para el control remoto de dispositivos eléctricos. Utiliza la línea eléctrica (220V o 110V) para transmitir señales de control entre equipos de automatización del hogar en formato digital.

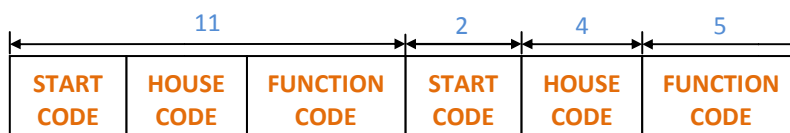
Un 1 binario se representa por un pulso de 120 KHz. durante 1 milisegundo, en el punto cero, y el 0 binario se representa por la ausencia de ese pulso de 120 KHz. El pulso de 1 milisegundo se transmite tres veces



para que coincida con el paso por el cero en las tres fases para un sistema trifásico. La Figura muestra la relación entre estos pulsos y el punto cero de la corriente alterna.

La transmisión completa de un código X-10 necesita once ciclos de corriente. Los dos primeros ciclos representan el Código de Inicio. Los cuatro siguientes ciclos representan el Código de Casa (letras A-P), los siguientes cinco representan o bien el Código Numérico (1-16) o bien el Código de Función (Encender, Apagar, Aumento de Intensidad, etc.). Este bloque completo (Código de Inicio, Código de Casa y Código de Función o Numérico) se transmite siempre dos veces, separando cada 2 códigos por tres ciclos de la corriente, excepto para funciones de regulación de intensidad, que se transmiten de forma continua (por lo menos dos veces) sin separación entre códigos.

Ciclos red eléctrica



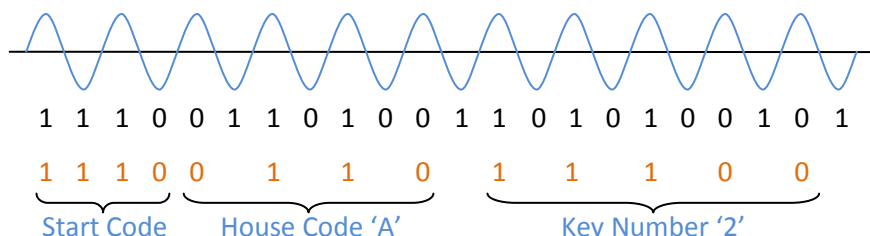
Código transmitido al pulsar un botón de número



Código transmitido al pulsar un botón de función

Las funciones de regulación de intensidad son excepciones a esta regla, y se transmiten de forma continua (al menos dos veces) sin separación entre códigos.

Dentro de cada bloque de códigos, cada cuatro o cinco bits de código deben ser transmitidos en modo normal y complementario en medios ciclos alternados de corriente. Por ejemplo, si un pulso de 1 milisegundo se transmite en medio ciclo (1 binario), entonces no se transmitirá nada en la siguiente mitad del ciclo (0 binario).



El protocolo X10 consta de bits de direcciones y de órdenes. Por ejemplo, puedes decir “lámpara #3”, “¡enciéndete!” y el sistema procederá a ejecutar dicho mandato, también puedes direccionar varias unidades antes de dar la orden: “lámpara #3, lámpara #12”, “¡encendeos!”. Existen múltiples instrucciones utilizadas por el protocolo entre las cuales destacan: “ON”, “OFF”, “ALL LIGHTS ON”, “ALL LIGHTS OFF”, “DIM” y “BRIGHT”.

Los dispositivos están generalmente enchufados a módulos X10 (receptores). X10 distingue entre *módulos de lámparas* y *módulos de dispositivos*. Los módulos de dispositivos proporcionan energía a los dispositivos eléctricos y aceptan órdenes X-10. Los módulos de dispositivos son capaces de gestionar cargas grandes (ej. máquinas de café, calentadores, motores, etc.), simplemente encendiéndolos y apagándolos.

Si desea controlar luces vía mandatos X-10, debería conectar la luz en un módulo de luz en la red y a continuación asignarle una dirección (A1, por ejemplo). Así, cuando envíe la orden «A1 ON» a través de los cables de la red eléctrica, la luz se debería encender. Cabe destacar que los módulos de lámparas no pueden soportar grandes cargas y sólo pueden controlar cargas resistivas, es decir, no sirven para bombillas de bajo consumo.

Formato de instrucciones X10 y sus tipos:

	CÓDIGOS DE CASA					CÓDIGOS DE UNIDAD (NÚMERO)				
	H1	H2	H4	H8		D1	D2	D4	D8	D16
A	0	1	1	0	1	0	1	1	0	0
B	1	1	1	0	2	1	1	1	0	0
C	0	0	1	0	3	0	1	1	0	0
D	1	0	1	0	4	1	0	1	0	0
E	0	0	0	1	5	0	0	0	1	0
F	1	0	0	1	6	1	0	0	1	0
G	0	1	0	1	7	0	1	0	1	0
H	1	1	0	1	8	1	1	0	1	0
I	0	1	1	1	9	0	1	1	1	0



J	1	1	1	1	10	1	1	1	1	0
K	0	0	1	1	11	0	0	1	1	0
L	1	0	1	1	12	1	0	1	1	0
M	0	0	0	0	13	0	0	0	0	0
N	1	0	0	0	14	1	0	0	0	0
O	0	1	0	0	15	0	1	0	0	0
P	1	1	0	0	16	1	1	0	0	0

Funciones: Apagar todas las unidades	0	0	0	0	1
Encender todas las luces	0	0	0	1	1
Encender	0	0	1	0	1
Apagar	0	0	1	1	1
Atenuar intensidad	0	1	0	0	1
Aumentar intensidad	0	1	0	1	1
Códigos de función que se añaden para controladores OEM (Original Equipment Manufacturer)					
Apagar todas las luces	0	1	1	0	1
Código extendido	0	1	1	1	1
Petición de saludo (1)	1	0	0	0	1
Aceptación de saludo	1	0	0	1	1
Atenuación preestablecida (2)	1	0	1	X	1
Datos extendidos (analógicos) (3)	1	1	0	0	1
Estado = on	1	1	0	1	1
Estado = off	1	1	1	0	1
Petición de estado	1	1	1	1	1

- (1)** La petición de saludo se transmite para comprobar si existen otros transmisores X-10 dentro del rango de escucha. Esto permite al OEM asignar un código de casa diferente si se recibe un mensaje de aceptación de saludo.
- (2)** En una instrucción de atenuación preestablecida, el bit D8 representa el bit más significativo del nivel; H1, H2, H4 y H8 representan los bits menos significativos.
- (3)** El código de datos extendidos se sigue de bytes que pueden representar información analógica (después de una conversión A/D). No debe existir separación entre los bytes de datos, ni entre el código de datos extendidos y de datos reales. El primer byte se puede utilizar para indicar cuántos bytes de información le seguirán. Si se dejan separaciones entre los bytes de información, estos códigos podrán ser recibidos por módulos X-10 causando un funcionamiento erróneo.

Anexo 2. Manipulando La Fonera

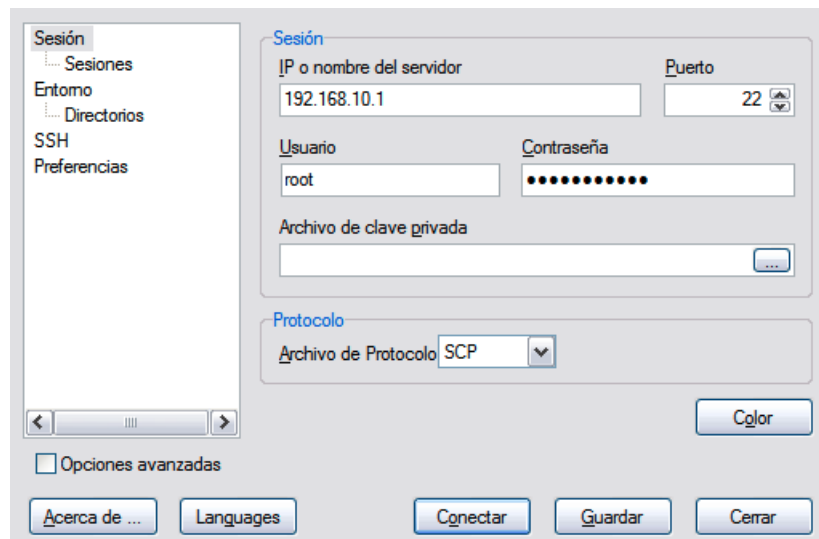
Cómo cargar/descargar archivos a/desde La Fonera

Para hacer esto desde Windows vamos a utilizar el programa WinSCP (<http://winscp.net/>).

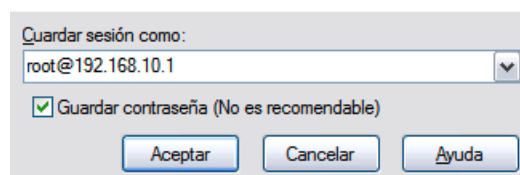
WinSCP es una aplicación de Software Libre. Es un cliente SFTP gráfico para Windows que emplea SSH. Su función principal es facilitar la transferencia segura de archivos entre dos sistemas informáticos, el local y otro remoto que ofrezca servicios SSH. Es especialmente útil para ahorrar tiempo durante el diseño de la interfaz para La Fonera, ya que permite mantener sincronizado un directorio local y uno remoto.

Pasos para conectar WinSCP con La Fonera:

1. Conectar con La Fonera mediante cable Ethernet o mediante WiFi (red "MyPlace" por defecto).
2. Configurar la conexión de WinSCP de la siguiente forma:



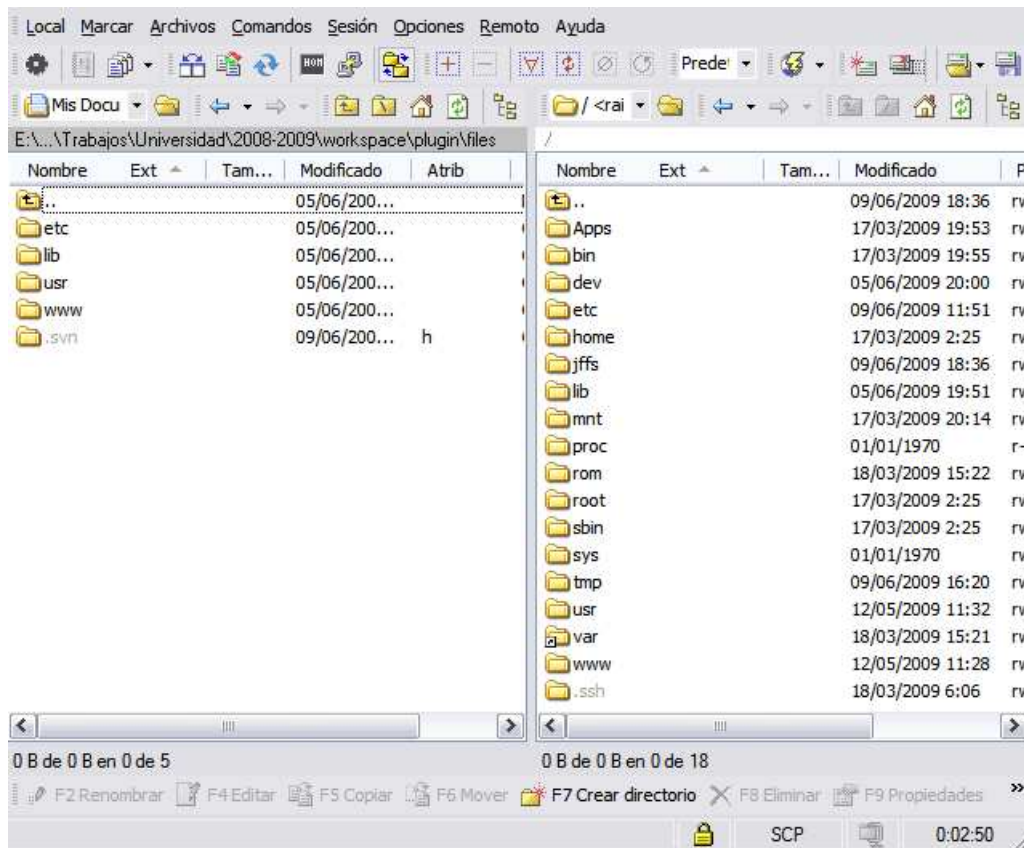
- Donde:
 - IP o nombre del servidor: dirección IP de La Fonera.
 - Puerto: puerto SSH.
 - Usuario: "root".
 - Contraseña: contraseña de La Fonera, por defecto "admin".
 - Protocolo: SCP
- Pulsar el botón "Guardar", asignarle un nombre cualquiera a la sesión y marcar la opción "Guardar contraseña" si no se desea introducir la contraseña cada vez que queramos conectarnos.





- Pulsar el botón "Aceptar" y luego "Conectar".



- La primera vez que nos conectemos nos preguntará si queremos conectarnos con un hardware que no conoce. Le diremos que sí. Dará un error al conectarse por no tener La Fonera el comando "groups". Pulsamos "Aceptar" en la ventana y aparecerá una ventana como la siguiente:



- Ya tendremos a la vista los archivos de La Fonera.

Dos opciones muy recomendables al trabajar con WinSCP son utilizar "Navegación sincronizada"  y, como ya se ha comentado, "Mantener actualizado el directorio remoto" .

Para hacerlo desde Linux podemos utilizar WinSCP sobre WINE (<http://www.winehq.org/>), un gestor de archivos como Konqueror sobre KDE (<http://www.konqueror.org/>) o por consola con los siguientes comandos:

- Descargar: `scp {usuario}@{ip}:/ruta/del/archivo/ {archivo}`
 - Ejemplo: Descargar el archivo "/tmp/wget.out" al directorio actual.
 - `scp root@192.168.10.1:/tmp/wget.out .`
- Cargar: `scp {archivo} {usuario}@{ip}:/ruta/del/archivo/`
- Para copiar carpetas completas, utilizar el parámetro "-r".
 - Ejemplo: Cargar la carpeta "./Carpeta" al directorio "/tmp" de La Fonera.
 - `scp -r Carpeta root@192.168.10.1:/tmp`

Cómo compilar para La Fonera

Compilar código en C

Pasos necesarios:

1. Descargar el SDK de OpenWRT para Atheros (procesador de La Fonera, arquitectura MIPS):
 - Versión para Linux de 32 bits: <http://downloads.openwrt.org/kamikaze/7.09/atheros-2.6/OpenWrt-SDK-atheros-2.6-for-Linux-i686.tar.bz2>
 - Versión para Linux de 64 bits: http://downloads.openwrt.org/kamikaze/7.09/atheros-2.6/OpenWrt-SDK-atheros-2.6-for-Linux-x86_64.tar.bz2
2. Descomprimir el archivo descargado.
3. Exportar el PATH:
 - `#PATH=$PATH:/dondesea/OpenWrt-SDK-atheros-2.6-for-Linux-i686/staging_dir_mips/bin`
 - `#export PATH`
 - Para comprobar que se ha hecho bien, al ejecutar `"echo $PATH"` debería mostrar dicha ruta entre otras.
4. Compilar el programa en C
 - `$ mips-linux-gcc ORIGEN.c -o DESTINO`
5. Copiar el ejecutable a La Fonera y darle permiso de ejecución. Hay muchas formas de hacerlo. Una de ellas puede ser copiarlo desde un servidor http:
 - Colocar el ejecutable compilado en la carpeta "www" del servidor http .
 - Copiar a La Fonera desde una sesión ssh:
 - `root@Fonera:~$ wget http://SERVIDOR/DESTINO`
 - `root@Fonera:~$ chmod +x DESTINO`
6. Ejecutar el archivo
 - `root@Fonera:~# ./DESTINO`

Compilar código en C++

Para poder ejecutar binarios de C++ hace falta instalar el paquete "libstdc++" en La Fonera:

```
root@Fonera:~# opkg install libstdc++_4.1.2-13_mips.ipk
```

Los pasos para compilar código en C++ son los mismos que para C, salvo el paso 4, que pasa a ser:

4. Compilar el programa en C++
 - `$ mips-linux-g++ ORIGEN.cpp -o DESTINO`

Cómo compilar la librería SQLite3 en C++ para La Fonera

Para generar archivos libsqlite3.so de librería dinámica hay que seguir los siguientes pasos:

1. Compilar librería (fuente de sqlite3: <http://sqlite.org/sqlite-amalgamation-3.6.11.zip>):
 - `$ mips-linux-gcc -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"sqlite3.d" -MT"sqlite3.d" -o "sqlite3.o" "sqlite3.c"`
2. Crear el archivo .so:
 - `$ mips-linux-g++ -shared -o"libsqlite3.so" ./sqlite3.o`

Una vez generado el archivo libsqlite3.so, para poder compilar código que enlaza a esta librería hay que copiar los siguientes archivos:



- El archivo de cabeceras "sqlite3.h" al directorio "OpenWrt-SDK-atheros-2.6-for-Linux-i686/staging-dir-mips/include/".
- El archivo generado "libssqlite3.so" al directorio "OpenWrt-SDK-atheros-2.6-for-Linux-i686/staging-dir-mips/usr/lib/".

Para compilar código en C++ para La Fonera que enlace a la librería de SQLite3 hay que ejecutar el siguiente comando:

- `$ mips-linux-g++ ORIGEN.cpp -L/dondesea/OpenWrt-SDK-atheros-2.6-for-Linux-i686/staging_dir_mips/usr/lib/ -o DESTINO -lsqlite3 -ldl -lpthread`

Anexo 3. Enlaces

La Fonera

- FON: <http://www.fon.com>
- Fonosfera: <http://trac.fonosfera.org/fon-ng>
- FON Boards: <http://boards.fon.com/>
- OpenWRT: <http://openwrt.org/>
- ATHEROS: <http://www.atheros.com/>
- Arquitectura MIPS: [http://es.wikipedia.org/wiki/Arquitectura MIPS](http://es.wikipedia.org/wiki/Arquitectura_MIPS)
- Repositorios con paquetes para La Fonera:
 - <http://ipkg.nslu2-linux.org/feeds/openwrt/fon-ng/>
 - <http://www.fonera.be/rep/FON2202/>
 - <http://members.home.nl/hans.zandbelt/openwrt/7.09/mips/packages/>
 - <http://downloads.openwrt.org/kamikaze/7.09/packages/mips/>

Lua

- Lua: <http://www.lua.org/>
- Manual de referencia: <http://www.lua.org/manual/5.1/es/>
- "Programming in Lua", primera edición: <http://www.lua.org/pil/>
- LuaForge: <http://luaforge.net/>

LuCI

- LuCI project: <http://luci.freifunk-halle.net/News>
- Documentación: <http://luci.freifunk-halle.net/Documentation>
- Aplicaciones desarrolladas con LuCI: <http://svn.luci.subsignal.org/luci/trunk/applications/>

Domótica

- Protocolo X10: <http://es.wikipedia.org/wiki/X10>
- Manuales X10: http://www.x10.com/support/support_manuals.htm
- Introducción a X10: <http://www.superinventos.com/Queesx10.htm>
- Domótica Viva: <http://www.domoticaviva.com/X-10/X-10.htm>
- Marmitek: <http://www.marmitek.com/>

- Heyu: <http://heyu.tanj.com/>
- Tiendas :
 - SuperInventos: <http://www.superinventos.com/>
 - DomoDesk: <http://www.domodesk.com/>
 - Domolandia: <http://www.domolandia.com/>
 - Seguridad 100: <http://www.seguridad100.com/>
 - Albedo: <http://www.albedo.biz/>
 - TodoElectrónica: <http://todoelectronica.com/>
 - CasaActiva: <http://www.casaactiva.com/>
 - DomóticaViva: <http://www.domoticaviva.com/>

Conversor USB-serie

- Linux Device Drivers, 3er edition: <http://lwn.net/Kernel/LDD3/>
- Major and minor numbers: <http://www.linux-tutorial.info/modules.php?name=MContent&pageid=94>
- Post en el foro de OpenWRT: <http://forum.openwrt.org/viewtopic.php?id=4745>
- Usbserial.h: <http://www.gelato.unsw.edu.au/lxr/source/drivers/usb/serial/usb-serial.h>

Interfaz web

- Calendario en Javascript: <http://www.mattkruse.com/javascript/calendarpopup/index.html>
- Barra de progreso en Javascript: <http://www.bram.us/2007/12/14/my-javascript-non-ajax-progress-percentage-bar-jsprogressbarhandler-02/>
- W3Schools: <http://www.w3schools.com/>
- Carga dinámica de HTML: <http://www.cryer.co.uk/resources/javascript/script4.htm>
- Universal iPhone UI Kit: <http://www.minid.net/iphone/>

Compilación cruzada

- Post en el blog "Kiríadas": <http://kiriadas.blogspot.com/2008/09/compilacin-cruzada-openwrt.html>
- Post en el blog "life ideas": <http://manoftoday.wordpress.com/2007/10/11/writing-and-compiling-a-simple-program-for-openwrt/>
- OpenWRT Buildroot: <http://downloads.openwrt.org/docs/buildroot-documentation.html>



AGRADECIMIENTOS

A nuestro tutor Juan Carlos Fabero Jiménez por haber aceptado nuestra propuesta y habernos ayudado en el desarrollo del proyecto.

A Inouk Bourgon por su ayuda, por el material cedido al proyecto y sobre todo por sus ánimos.

A Charles Sullivan, creador de Heyu, por su ayuda y dedicada atención.

Al chat de desarrollo de FON por habernos ayudado con los problemas que hemos tenido en el desarrollo de la aplicación.

A nuestros amigos y familiares, sobre todo a aquéllos que han mostrado un interés especial en el proyecto, y que nos han animado a seguir adelante.



AUTORIZACIÓN A LA UCM

Por la presente se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto esta memoria, como el código, la documentación y el prototipo desarrollado.

Firma de los autores:

Miguel Ángel Díaz Calvo

Daniel Dionne González

David Martín Moreno



PALABRAS CLAVE

Listado de palabras clave utilizadas en FonHome:

- Sistema domótico
- FON
- La Fonera
- X10
- OpenWRT
- Programar eventos y acciones
- FonHome

