Sistema de monitorizacion, control de un inversor fotovoltaico Fronius con raspberry pi y analizador de red.

# Introducción

La mayoría de modelos de inversores fotovoltaicos de conexión a red disponen de algún mecanismo para la recolección de datos con propósitos principalmente de monitorización remota, transmisión de alertas y, en algunos casos, ajuste de parámetros de funcionamiento del equipo. Desafortunadamente no existe un mecanismo estándar entre los diferentes fabricantes de inversores y cada uno ellos a puesto a punto interfaces y protocolos propietarios para su gama de productos.

Así, por ejemplo, los inversores Fronius disponen de un interfaz serie RS 422 (RS232 en los sistemas más antiguos) que permite la recolección de los parámetros de funcionamiento actuales o históricos de un conjunto de inversores. Además en algunos modelos equipados con versiones de firmware más moderno, es posible limitar la potencia máxima del inversor.

El propósito de este trabajo es desarrollar y poner a punto un software monitorización y control con los siguientes objetivos:

1) Emplear un computador de placa reducida (en inglés: Single Board Computer o SBC) con el mínimo consumo y de bajo coste. Se ha elegido emplear una Rapsberry Pi modelo B

2) Incorporar en el ordenador de placa reducida el proceso software de monitorización que recoja información del consumo de la instalación interior

que esté permanentemente recogiendo datos del inversor con propósito de visualización y análisis de la potencia y energía entregada

3) Incorporar un segundo proceso que recoja los datos del inversor con propósito de visualización y análisis de la potencia y energía generada.

4) Oportunamente realizar una limitación de la potencia entregada por el inversor de forma que se pueda controlar la energía exportada a la red en función de diversos factores.

5) Poder realizar una señalización o modulación de la energía a ser consumida por determinados receptores no prioritarios en función del la generación, los consumos prioritarios y ocasionalmente de la predicción de energía generada.

6) Obtener información de los contadores en el punto frontera con la red de distribución eléctrica

7) Obtener en avance los parámetros que permitan la facturación de la energía consumida y la facturación o compensación de la energía exportada.

8) Configurar este proyecto como de fuentes abiertas y de bajo presupuesto para animar la instalación de sistemas de generación fotovoltaica por pequeños consumidores.

# Hoja de ruta del proyecto

La idea es obtener un programa que permita ir añadiendo funcionalidades de monitorización, alerta y control de forma evolutiva. Se propone un sistema formado por varios procesos Linux cada uno de ellos dedicado a una tarea específica sobre cada uno de los elementos del posibles elementos del sistema: inversores de diferentes marcas (protocolos), analizadores, contadores, sensores, relés, dimmers, etc.

Un posible esquema final del proyecto se muestra a continuación





# Fase 1: Monitorización de los datos de consumo de los receptores en la red interior

Para la monitorización de los parametros de red se emplea el analizador del fabricantre Orbis modelo ANRET -M22-BUS.

Para probar el dispositivo se emplea el programa modpoll que simula un dispositivo master de la red RS485. Este programa no está disponible para raspberrypi por lo que la prueba se debe hacer en un linux en hardware x86-64

El comando a emplear es:

root@jubuntu32rasp:/home/juan# ./modpoll -b 9600 -p none -m rtu -a 1 -r 1 -c 10 -f -t 3:float /dev/ttyUSB0

Obteniéndose (sin conexión de la medida de intensidad en el analizador de red):

[1]: 230.747147

[3]: 0.000000

[5]: 0.000000

[7]: 0.000000

[9]: 0.000000

[11]: 1.000000

[13]: 0.000000

[15]: 50.000000

[17]: 0.000000

[19]: 0.000000

El esquema lógico de la monitorización se expone en el siguiente diagrama:

## 

## Proceso daemon consumond

El programa consumond (**consu**mo **mon**itor **d**aemon) utiliza el protocolo MODBUS-RTU a través del interfaz RS485 para comunicar con el analizador de red.

El formato de una trama MODBUS RTU tiene el siguiente formato general:

|  |  |  |
| --- | --- | --- |
| Modbus RTU frame format | | |
| Name | Length (bits) | Function |
| Start | 28 | At least 3 1⁄2 character times of silence (mark condition) |
| Address | 8 | Station address |
| Function | 8 | Indicates the function code; e.g., read coils/inputs |
| Data | n × 8 | Data + length will be filled depending on the message type |
| CRC | 16 bits | Checksum |
| End | 28 | At least 3 1⁄2 character times of silence between frames |

El orden de byte es "Big-Endian" (primer byte contiene el MSB - most significant Byte ).

Los códigos de función de MODBUS son los siguientes:

|  |  |
| --- | --- |
| Function name | Function code |
| Read Input Registers | 4 |
| Read Holding Registers | 3 |
| Read Discrete Inputs | 2 |
| Read Coils | 1 |
| Write Multiple Coils | 15 |

El formato de una trama de petición MODBUS RTU desde el dispositivo master a un "slave" tiene el siguiente formato general:

|  |  |  |
| --- | --- | --- |
| Name | Length (bytes) | Function |
| Address | 1 | Station address |
| Function | 1 | Indicates the function code; e.g., read coils/inputs |
| Initial register/coil | 2 | Registro o coil inicial |
| Register/coil count | 2 | Número de registros o coils solicitados |
| CRC | 16 bits | Checksum |

El formato típico[[1]](#footnote-1) de una trama de respuesta MODBUS RTU desde el dispositivo "slave" tiene el siguiente formato general:

|  |  |  |
| --- | --- | --- |
| Name | Length (bytes) | Function |
| Address | 1 | Station address |
| Function | 1 | Indicates the function code; e.g., read coils/inputs |
| Byte count | 1 | Numero de bytes (N) de la respuesta que siguen y sin incluir los de CRC |
| Data | N | Datos devueltos |
| CRC | 16 bits | Checksum |
|  |  |  |

Para obtener regularmente los 10 parámetros eléctricos (20 registros de 2 bytes) que recoge el analizador Orbis ANRET M22 BUS, se envía periódicamente al interfaz RS485 el siguiente comando MODBUS con destino al analizador de red que en este caso está configurado con la dirección 0x01. (Ver anexo con el mapa de memoria del analizador de red).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x01 | 0x04 | 0x00 | 0x00 | 0x00 | 0x14 | 0xF0 | 0x05 |
| Dirección del dispositivo destino | Función MODBUS | Low | high | low | High | low | High |
| Registro Inicial | | Número de words | | CRC[[2]](#footnote-2) | |

Con este comando se obtienen los 10 parámetros eléctricos (20 registros de 2 bytes )que recoge el analizador Orbis ANRET M22 BUS.

Para la puesta a cero del contador parcial de energía, se envía el siguiente comando MODBUS:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x01 | 0x0F | 0x00 | 0x00 | 0x00 | 0x01 | 0x01 | 0xCB | 0x6F |
| Dirección del dispositivo destino | Función MODBUS | Low | high | low | High | Valor binario:0000 0001 | low | High |
| Coil Inicial | | Número de coils | | CRC | |

La respuesta de este comando es diferente a la de la función 0x04:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x01 | 0x0F | 0x00 | 0x00 | 0x00 | 0x01 | 0xLL | 0xHH |
| Dirección del dispositivo destino | Función MODBUS | Low | high | low | High | low | High |
| Coil Inicial | | Número de coils | | CRC | |

## Control del puerto serie

Para el control del puerto serie se puede leer el contenido del siguiente enlace:

<http://en.wikibooks.org/wiki/Serial_Programming>

<http://en.wikibooks.org/wiki/Serial_Programming/termios>

El puerto inicialmente se abre con el flag O\_NOCTTY para evitar que el proceso lo emplee como "control terminal".

fd = open (portname1, O\_RDWR | O\_NOCTTY);

La configuración del puerto serie es realizada en una función configura\_puerto\_serie(int fd) empleando el interfaz "termios".

Conceptualmente lo más importante es configurar el puerto serie como "raw" para evitar todo filtro y proceso orientado a la gestión de terminales TTY.

La función de configuración de termios es tcsetattr syscall que tiene muchos parámetros de configuración pero que en nuestro caso se emplearan únicamente unas pocos.

void configura\_puerto\_serie(int fd){

struct termios tp;

tcgetattr(fd, &tp); /\* se obtiene la actual configurtación del puerto \*/

cfmakeraw(&tp); /\* modo raw \*/

cfsetspeed(&tp, B9600); /\* velocidad del puerto serie \*/

tcsetattr(fd, TCSANOW, &tp); /\* ¡hagase! \*/

tcflush(fd, TCIOFLUSH);

}

## Memoria compartida

El proceso crea un segmento de memoria compartida con clave 0x00001234 y con permisos de escritura y lectura para cualquier usuario.

/\*

\* crea area de memoria compartida

\*/

int shmid; // identificador de memoria compartida

shmid = shmget(0x00001234, sizeof (struct datos\_publicados), IPC\_CREAT | 0666);

datos\_publicados = shmat(shmid, NULL, 0);

......

De esta manera los diferentes procesos pueden intercambiar información.

Hay que tener en cuenta que este segmento sigue definido aunque se destruya el proceso y hay que liberar este segmento expresamente. Para esto se debe emplear el comando Linux:

#ipcrm -M 0x00001234

Para que el sistema muestre los segmentos de memoria compartidos se puede emplear el comando:

#ipcs

# Fase 2: Monitorización de los datos del inversor

La primera fase del proyecto es un proceso de monitorización de los datos facilitados por un único inversor de la marca Fronius.

El esquema lógico de la monitorización se expone en el siguiente diagrama:



Comentarios para el diseño:

El inversor será un equipo Fronius que emplea el protocolo “Fronius Interface” a través de una red de inversores conectados en semi-anillo mediante conexiones RS-422 entre nodos. El ordenador de tarjeta reducida empleará un puerto serie (típicamente accesible a través del dispositivo /dev/ttyUSB1) para realizar la conexión al primer ordenador de la red de inversores.

En una primera versión del proceso de monitorización, este únicamente recogerá datos de un único inversor conectado a la red. Sin embargo, el programa se diseñará para permitir una fácil evolución del programa para poder recoger datos de varios (no muchos) inversores conectador a la misma red.

En el caso de tener que monitorizar inversores conectados a distintas redes, esto es, a través de diferentes puertos serie (ttyUSB1, ttyUSB2, etc), se arrancará un proceso monitor para cada red. De esta manera un inversor de marca diferente a Fronius y con diferente protocolo de monitorización, será accedido a través de un segundo puerto (p.e. ttyUSB2) por un programa de monitorización diferente al empleado para el primer inversor Fronius. De esta manera se puede hacer crecer un sistema fotovoltaico incorporando inversores nuevos que no tengan que ser de la misma marca y emplear el mismo protocolo de monitorización. El diseño del programa permitirá la coexistencia de varios procesos similares pero sobre interfaces serie (o de otro tipo de comunicación como ethernet) diferentes.

El programa arrancará automáticamente como un “daemon” o servicio del sistema operativo Linux.

Cada programa de monitorización específico de cada puerto/protocolo/marca‑modelo tendrá un nombre diferente que permita identificarlos fácilmente tanto a nivel de fichero binario como de proceso en memoria. Cada programa específico empleará un fichero de configuración, y posiblemente otros diferentes para el almacenamiento de estado de cada inversor, el log de funcionamiento y el de errores.

Además, como función fundamental, cada proceso de monitorización registrará en ficheros separados la información correspondiente a cada inversor. Es decir si un mismo proceso monitoriza tres inversores en la misma red y del mismo fabricante, este proceso generará tres ficheros de datos, uno por cada uno de los inversores que atiende.

Los posibles datos a registrar son:

**Marca de tiempo** dd/mm/aaaa hh:mm:ss

Intervalo temporal (ms)

Energía generada (mW·s)

Potencia media

Desviación estándar potencia

Potencia máxima (W)

Potencia mínima

Limite medio de potencia (%)

Límite máximo de potencia (%)

Límite mínimo de potencia (%)

Desviación estándar del límite de potencia

Nº veces que se ha puesto un límite diferente a la potencia de inversor

El intervalo de registro podrá ser variable. Inicialmente se empleará el valor de 1 minuto. Por tanto el registro tendrá 24\*60= 1440 entradas diarias a unos 200bytes por entrada suponen 400KB/dia (3MB/semana, 12MB/mes, 48MB/cuatrimestre, 200MB/año).

Se empleará un registro circular que mantenga el tamaño del fichero de datos en torno a 50MB. Un futuro proceso de consolidación generará registros con valores de generación, consumo e importación y exportación con frecuencias mucho menores (horario, diario, semanal, mensual, anual, total).

### Detección de interfaces serie

Para confirmar el reconocimiento de los interfaces serie a través del USB, se puede dar el comando siguiente:

$ lsusb

Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

USB-RS422

Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.

USB-RS485

Bus 001 Device 004: ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port

Bus 001 Device 005: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter

### Apertura de dispositivo de puerto serie

En Linux los interfaces serie a traves de USB aparecen en el directorio /dev con los nombres ttyUSB0, ttyUSB1, etc.

$ ls -l /dev/ttyUSB\*

crw-rw---T 1 root dialout 188, 0 Jan 1 1970 /dev/ttyUSB0

crw-rw---T 1 root dialout 188, 1 Jan 1 1970 /dev/ttyUSB1

La apertura del dispositivo se debe realizar con la función open(), en modo read/write e indicando

fd = open (portname1, O\_RDWR | O\_NOCTTY | O\_SYNC);

O\_NOCTTY: If pathname refers to a terminal device—see [tty(4)](http://man7.org/linux/man-pages/man4/tty.4.html)—it will not become the process's controlling terminal even if the process does not have one.

El usuario bajo el que corre el proceso debe estar incluido en el grupo de usuarios "dialout".

# usermod -a -G dialout juan

(Requiere el reinicio de la sesión para que aplique)

En el caso de conectar más de un adaptador USB-serial es preciso confirmar la correspondencia entre cada adaptador USB-Serial y el nombre de dispositivo en el directorio /dev:

$ dmesg |grep usb\*

3.314258] usb 1-1.2: new full-speed USB device number 4 using dwc\_otg

[ 3.440369] usb 1-1.2: New USB device found, idVendor=067b, idProduct=2303

[ 3.449098] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=0

[ 3.458372] usb 1-1.2: Product: USB-Serial Controller

[ 3.465026] usb 1-1.2: Manufacturer: Prolific Technology Inc.

[ 3.568290] usb 1-1.3: new full-speed USB device number 5 using dwc\_otg

[ 3.691526] usb 1-1.3: New USB device found, idVendor=1a86, idProduct=7523

[ 3.700826] usb 1-1.3: New USB device strings: Mfr=0, Product=2, SerialNumber=0

[ 3.709940] usb 1-1.3: Product: USB2.0-Serial

[ 6.264458] usbcore: registered new interface driver usbserial

[ 6.378245] usbcore: registered new interface driver usbserial\_generic

[ 6.386589] usbserial: USB Serial support registered for generic

[ 6.611443] usbcore: registered new interface driver ch341

[ 6.639686] usbcore: registered new interface driver pl2303

[ 6.647023] usbserial: USB Serial support registered for pl2303

RS422 en ttyUSB1

[ 7.006701] usbserial: USB Serial support registered for ch341-uart

[ 7.354497] usb 1-1.2: pl2303 converter now attached to ttyUSB1

RS485 en ttyUSB0

[ 7.468343] usb 1-1.3: ch341-uart converter now attached to ttyUSB0

En el caso mostrado se observa que el interfaz RS422 (PL2303) del inversor corresponde al dispositivo ttyUSB1 y el interfaz RS485 (ch341-uart) del analizador de red al ttyUSB0.

El programa de monitorización del inversor da por supuesto que para acceder al inversor se debe emplear el dispositivo /dev/ttyUSB1. En el caso de no ser así se puede intentar cambiar los puertos USB a los que se conectan los adaptadores para lograr la asignación deseada. Sin embargo, al menos en el caso concreto que nos ocupa, se ha comprobado que este cambio no funciona y lo normal (pero no siempre, especialmente cuando se hace un reboot sin desconexión de potencia) es que el convertidor ch341-uart (RS 485) se asigne a /dev/ttyUSB0 y el convertidoPL2330 (RS422) quede asignado a /dev/ttyUSB1 con independencia de los puertos USB a los que se conecten. Esto debe estar determinado por los tiempos que tarden es ser descubiertos los dispositivos en el bus, lo cual no es un procedimiento fiable. Se ha observado que la conexión en caliente del convertidor ch341-uart (RS 485) provoca el rearranque de la raspberrypi. Existe una técnica para asegurar la denominación del nodo de dispositivo para un dispositivo USB usando la reglas del programa udev. <http://askubuntu.com/questions/49910/how-to-distinguish-between-identical-usb-to-serial-adapters>

[http://www.linux-usb.org/FAQ.html#i6](http://www.linux-usb.org/FAQ.html%23i6)

<http://reactivated.net/writing_udev_rules.html>

Para que la lectura se realice correctamente se debe configurar por API (o por línea de comando) el puerto de comunicaciones serie en modo raw (modo no canonico) y si procesamiento de salida (-opost[[3]](#footnote-3)):

$ sudo stty -F /dev/ttyUSB0 -icanon -echo -echoe -isig -opost

$ sudo stty -F /dev/ttyUSB0 raw

$ sudo stty -F /dev/ttyUSB0

speed 9600 baud; line = 0;

min = 1; time = 0;

-brkint -imaxbel

-opost

-isig -icanon -echo –echoe

**IMPORTANTE:** Fuentes para control de la asignación ttyUSBx independientemente de la numeración de dispositivos en el bus USB:

<http://noctis.de/ramblings/linux/49-howto-fixed-name-for-a-udev-device.html>

<http://stackoverflow.com/questions/7986034/linux-how-to-force-a-usb-device-to-use-the-same-ttyusb-number>

<http://www.redhat.com/magazine/002dec04/features/udev/>

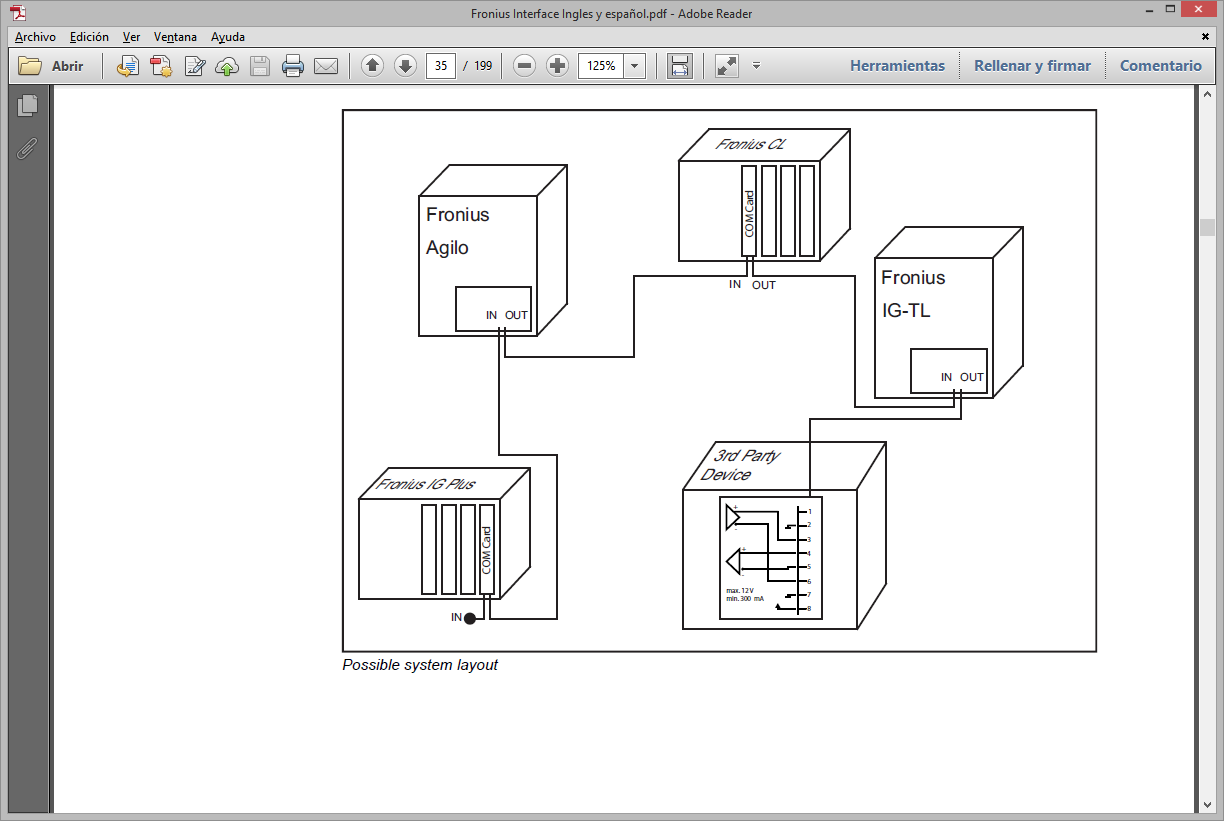
## Breve descripción del protocolo Fronius

A continuación se muestra el formato de las trama básica del protocolo de Fronius.

## 

La trama del protocolo empieza con tres bytes de valor 0x80 (I) seguido de un byte que permite conocer la longitud de la trama (L) ya que indica el número de bytes de datos (V). justo después del campo longitud viene tres bytes con los valores correspondientes al origen o destino de la trama: tipo de dispositivo (D), numero de dispositivo (N) y comando (C). Al final de la trama se ubica un checksum que sirve para comprobar la integridad de la trama que incluye todos los campos menos la cabecera y el propio checksum.

El protocolo es del tipo maestro/esclavo donde el maestro es el dispositivo que interroga a un numero de inversor y el inversor identificado por ese número responde. La trama de interrogación va pasando por los todos los inversores y resto de dispositivos de la red semianillo RS422. Cuando un inversor lee una trama dirigido a él, responde en función del comando solicitado.



Entre los posible valores para el tipo de dispositivo (u opción) el del tipo inversor es el más empleado, pero existen otros posibles dispositivos:

|  |  |
| --- | --- |
| 0x01 | Inverter |
| 0x00 | General data query or query to the interface card (the "Number" byte is ignored) |
| 0x02 | Sensor card |

En el caso de una red RS422 de inversores la tramas con los comandos son enviados de una inversor al siguiente. El inversor afectado es el que emite una trama de repuesta.

Si un inversor recibe una trama de respuesta con el mismo número de inversor que el posee, este inversor sobre escribirá la trama de respuesta con un mensaje de error. Obviamente en la red no pueden existir dos inversores con el mismo número de dispositivo.

Si un comando es enviado a un dispositivo que no existe en la red, el dispositivo maestro recibirá la trama sin ser respondida. Es decir la trama pasa por todos los inversores y ninguno contesta.

Si un comando es enviado a un dispositivo que no soporta el comando solicitado, una trama de error será emitida en respuesta colocando el valor 0x0E (error) en el campo donde estaba el valor del comando en la trama de solicitud.

Los mensajes de error que pueden ser emitidos son:

|  |  |  |
| --- | --- | --- |
| 0x07 | | Set error sending |
| 0x0D | Set error forwarding | |
| **0x0E** | **IFC protocol error** | |
| **0x0F** | **Status Messages** | |

Los errores de protocolo aparecen cuando:

- se dirige una consulta al inversor y el inversor no es capaz de procesar la consulta

- o cuando se detecta un error en la estructura de datos de la consulta

En los errores de protocolo el valor del byte de comando es siempre 0x0E.

El comando que ha provocado el error se muestra como primer byte en el campo de datos.

Un segundo byte informa del error ocurrido, según la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Valor | Explicación | Comentario |
| 0x01 | Comando desconocido |  |
| 0x02 | Timeout (Tiempo límite) | En la red no se ejecuta un comando o una consulta de valor de medición dentro de un determinado tiempo |
| 0x03 | Estructura de datos incorrecta |  |
| 0x04 | La cola de espera de los comandos a ejecutar está llena | Esperar hasta que se haya ejecutado el último comando |
| 0x05 | El equipo o la opción no está disponible | El equipo o la opción a la que se ha dirigido el comando no están disponibles en la red |
| 0x06 | Sin reacción del equipo o de la opción | El equipo o la opción a la que se ha dirigido el comando no responden |
| 0x07 | Error de sensor | El equipo o la opción a la que se ha dirigido el comando emiten un error de sensor |
| 0x08 | Sensor no activo | Se emite cuando el canal seleccionado no se encuentra activo |
| 0x09 | Comando incorrecto para el equipo o la opción | No se puede ejecutar el comando en combinación con el equipo seleccionado o la opción seleccionada |
| 0x0A | Advierte de que hay dos equipos en la red con el mismo número de red. | El equipo que detecta el error sobre escribe el mensaje actual con una trama de error de este mensaje de error. |

Si se produce un error en un inversor, el inversor solo envía el error (el mensaje de estado) una vez. Los mensajes de estado se emiten para todos los tipos de inversores y proporcionan información sobre un error de sistema de un inversor.

Estructura:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inicio | Longitud | Equipo/ opción | Número | Comando | Código de estado | | Extra | Suma de chequeo |
| 3B | 1B | 1B | 1B | 1B | 1B | 1B | 1B | 1B |

Una explicación de los código de estado figura en el manual de instrucciones del correspondiente inversor. ¡OBSERVACIÓN! Si el número de módulo describe un número de 1-15, debe restarse el número 1 del valor indicado. El número calculado corresponde el número de módulo de la dirección de bus "HID" de una etapa de potencia. Ejemplo código de estado 301:

|  |  |
| --- | --- |
| Código de estado | |
| 0x01 | 0x2D |

Información en el byte extra:

|  |  |  |
| --- | --- | --- |
| Bit | Valor | Explicación |
| 7 | 0 | El número de módulo describe el módulo (1-15 = etapa de potencia, 0 = otro módulo, por ejemplo: IG.Brain, etc.) |
| 7 | 1 | El número de módulo debe interpretarse como ID de ventilador (por ejemplo, como en caso del IG 500) |
| 0 - 3 | 0 - 15 | El número de módulo describe el módulo (1-15 = etapa de potencia, 0 = otro módulo, por ejemplo: IG.Brain, etc.) |

¡OBSERVACIÓN! Si el número de módulo describe un número de 1-15, debe restarse el número 1 del valor indicado. El número calculado corresponde el número de módulo de la dirección de bus "HID" de una etapa de potencia.

Se transmite la trama con la última tasa de baudios determinada o ajustada. Si aún no se ha ajustado ninguna tasa de baudios, se transmite la trama con el ajuste de la tasa de baudios "Default”.

## Detección y aislamiento de tramas de protocolo Fronius Interface

Los bytes recibidos en el puerto serie son leídas por una thread *recibida\_trama()* que ejecuta un bucle permanente. Esta thread está bloqueada en una función *read()* de lectura del puerto serie correspondiente al interfaz RS 422. Tan pronto como llegue una ráfaga de bytes, la función *read()* retorna besbloqueando la ejecución de la thread e informando de los bytes efectivamente leídos en el buffer.

El final de la ráfaga queda determinado por el parámetro de configuración del puerto serie *tp.c\_cc[VTIME] = 1* y que expresa el tiempo transcurrido entre caracteres, expresado en décimas de segundo( en este caso 0,1 segundo), a partir del cual se considera como final de ráfaga y que, por tanto acaba, por devolver resultad la función *read().*

Complementariamente se fija el parámetro *tp.c\_cc[VMIN] = 255* como el valor, de 0 a 255, de caracteres recibidos superado el cual se desbloquea y retorna la función *read()*. Para una explicación detallada de los parámetros VMIN y VTIME, consultar <http://www.unixwiz.net/techtips/termios-vmin-vtime.html>

Las lectura se hace sobre un buffer de dimensión doble de la trama más larga del protocolo (135Bytes). El puerto serie se configura con VMIN=255Bytes pero la lectura de la función read() se limita a 135Bytes. De esta forma con mucha probabilidad la función read() entregará un número de tramas enteras y no será preciso realizar operaciones de movimiento de bytes restantes para concatenar con nuevas lecturas.







# Fase 3: Limitación de la potencia de inversor para limitar la exportación de energía a la red

En el caso de que autoconsumo instantáneo, sin retribución ni compensación de la energía exportada, existen varias razones para reducir la potencia entregada de un inversor en aquellos momentos en los que el consumo en las cargas de la red interior sea inferior a la potencia generada y se tenga que exportar la energía a la red de distribución.

Alargar la vida útil del equipo

Cumplimiento de la normativa de la empresa de distribución eléctrica

Cumplimiento de reglamentos de la administración pública competente, especialmente en un futuro, que prevengan la sobrecarga en la red.

En España, es el Reglamento Electrotécnico de Baja Tensión, concretamente en su Instrucción Técnica Complementaria nº 40, la que especifica las condiciones y requisitos técnicos de los generadores de baja tensión. Ni en esta instrucción técnica, ni en el RD de 1699/2011 que regula la conexión de generadores de pequeña potencia para productores, ni el RD 1110/2007 que contiene el Reglamento de Puntos de Medida, ni en RD 1955/2000 que regula los sectores de distribución eléctrica, ni en la Ley 24/2013 del Sector Eléctrico.

Sin embargo, alguna importante empresa de distribución eléctrica con sede en Bilbao, en el caso de que la energía exportada no esté retribuida, obliga a instalar un dispositivo que evite la exportación de energía. Por el contrario, se tienen noticias de que otra importante empresa de distribución eléctrica con sede en Barcelona, no impone ningún requisito adicional a los recogidos en la ITC-BT-40 del Reglamento Electrotécnico de Baja Tensión facilitando y abaratando notablemente la instalación de autoconsumo.

Con independencia de las razones que se tengan y atendiendo al hecho de que es posible que se tenga que limitar la potencia de generación, se plantea el objetivo de monitorizar el equilibrio de potencias con el fin inicial de evitar la exportación de potencia con un tiempo de respuesta inferior a 2s.

A este fin se arrancará un proceso de medición de la potencia consumida ….

# Fase 4: Señalización o modulación de potencia para realizar consumos en receptores no prioritarios

# Fase 5: Cálculo de los costes de energía

En la siguiente dirección se pueden descargar un fichero en XML con datos para el cálculo de los costes de la energía consumida (ejemplo para el dia 7/01/2015):

<http://www.esios.ree.es/Solicitar?fileName=pvpcdesglosehorario_20150106&fileType=xml&idioma=es>

Este el fichero que permite obtener el precio por hora (FEU, PMH, etc).



Fichero Excel con los datos de Detalle para el cálculo del término energía PVPC (ejemplo para el día 7/01/2015):

<http://www.esios.ree.es/Solicitar?fileName=PVPC_DETALLE_DD_20150107&fileType=xls&idioma=es>

Los datos (FEU y TCU) PVPC para un día concreto (sin desglose horario) y para las tres tarifas y sus correspondientes periodos se pueden obtener descargando el fichero XML:

<http://www.esios.ree.es/Solicitar?fileName=preciovoluntariopconsumidor_20150107&fileType=xml&idioma=es>

Los datos con el perfil de consumo de una semana (ejemplo del 3 al 9 de enero de 2015) se pueden obtener descargando el fichero XML:

<http://www.esios.ree.es/Solicitar?fileName=perfilconsumo_20150107&fileType=xml&idioma=es>

# Anexo I. Mapa de memoria analizador de red Orbis ANRET M-22-BUS

**Read holding register:**

Función para la lectura del registro utilizado para memorizar los parámetros programables del instrumento. Los registros se programan con la función “Preset multiple register”.

Listado de “holding register”:

40001: primario del trasformador amperimétrico (en Amperios)

40002: Retroiluminación (1 activa, 0 desactivada)

40003: Parámetros de comunicación serie con la estructura:

A7 A6 A5 A4 A3 A2 A1 A0 --- --- P1 P0 B3 B2 B1 B0

B3 B2 B1 B0 Baud rate

0 0 0 0 1200 bps

0 0 0 1 2400 bps

0 0 1 0 4800 bps

0 0 1 1 9600 bps

P1 P0 parity

0 0 none

0 1 Odd

1 0 Even

**Read input register:**

Función para la lectura de los registros de medida. El ANRET M-22 BUS permite la lectura de 12 registros por solicitud:

Dirección Nº words Parámetro Unidad

30001 2 V-Tensión [V]

30003 2 I - Corriente [A]

30005 2 P – Potencia activa [W]

30007 2 A – Potencia Aparente [VA]

30009 2 Q – Potencia reactiva [var]

30011 2 PF – Factor de potencia ---

30013 2 φ - Ángulo de desfase ---

30015 2 f – Frecuencia [Hz]

30017 2 Ena – Energía activa absoluta [kWh]

30019 2 Ea – Energía activa parcial (reseteable). [kWh\*100]

Todas las medidas que figuran en los registros de entrada (a excepción de los contadores de energía) se expresan en el formato estándar numérico de punto flotante IEEE-754.

**Force multiple coil:**

Función para ejecutar comandos en el instrumento. Los controles son vistos como “Coils”.

0001 reset del contador de energía active parcial reseteable.

**Preset multiple register:**

Función para programar los registros del tipo “Holding”.

Listado de “holding register”: ver listado en apartado Read holding register.

# Anexo II. Ejemplo de curvas de consumo y generación real con limitación de la energía exportada a la red



# Anexo III. Arranque automático de los procesos de monitorización

Idealmente los programas de monitorización del inversor (fronius\_mon) y del analizador de la red interior (anret\_mon) deberían ser configurados para funcionar como daemons. Hasta que se estabilice el funcionamiento los programa de monitorización serán programas normales, con salidas por el stdout y stderr, y ejecutandose desde la shell del sistema.

## Ejecución desde la shell

Para la ejecución desde la shell en el backgroung y evitando la salida por stdout y stderr se deben dar las ordenes de ejecución:

#nohup ./anret\_mon 1> /dev/null 2 > /dev/null &

#nohup ./fronius\_mon 1> /dev/null 2 > /dev/null &

Para comprobar que los procesos se estan ejecutando:

ps aux | grep '[a]nret\_mon'

ps aux | grep '[f]ronius\_mon'

## Empleo de un script para su inclusión en /etc/init.d/

Tomado de:

<http://stackoverflow.com/questions/7221757/run-automatically-program-on-startup-under-linux-ubuntu>

<https://github.com/fhd/init-script-template>

<http://www.linux.com/learn/tutorials/442412-managing-linux-daemons-with-init-scripts>

Se parte del patron de script que se muestra a continuación:

#!/bin/sh

### BEGIN INIT INFO

# Provides: *scriptname*

# Required-Start: $remote\_fs $syslog

# Required-Stop: $remote\_fs $syslog

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: Start daemon at boot time

# Description: Enable service provided by daemon.

### END INIT INFO

dir="/home/pi/"

cmd="/home/pi/anret\_mon"

user=""

name=`basename $0`

pid\_file="/var/run/$name.pid"

#stdout\_log="/var/log/$name.log"

stdout\_log="/dev/null"

stderr\_log="/var/log/$name.err"

get\_pid() {

cat "$pid\_file"

}

is\_running() {

[ -f "$pid\_file" ] && ps `get\_pid` > /dev/null 2>&1

}

case "$1" in

start)

if is\_running; then

echo "Already started"

else

echo "Starting $name"

cd "$dir"

$cmd >> "$stdout\_log" 2>> "$stderr\_log" &

echo $! > "$pid\_file"

if ! is\_running; then

echo "Unable to start, see $stdout\_log and $stderr\_log"

exit 1

fi

fi

;;

stop)

if is\_running; then

echo -n "Stopping $name.."

kill `get\_pid`

for i in {1..10}

do

if ! is\_running; then

break

fi

echo -n "."

sleep 1

done

echo

if is\_running; then

echo "Not stopped; may still be shutting down or shutdown may have failed"

exit 1

else

echo "Stopped"

if [ -f "$pid\_file" ]; then

rm "$pid\_file"

fi

fi

else

echo "Not running"

fi

;;

restart)

$0 stop

if is\_running; then

echo "Unable to stop, will not attempt to start"

exit 1

fi

$0 start

;;

status)

if is\_running; then

echo "Running"

else

echo "Stopped"

exit 1

fi

;;

\*)

echo "Usage: $0 {start|stop|restart|status}"

exit 1

;;

esac

exit 0

Copy template to /etc/init.d and rename it to something meaningful. Then edit the script and enter that name after Provides: (between ### BEGIN INIT INFO and ### END INIT INFO).

Now set the following three variables in the script:

dir : The working directory of your process.

user: The user that should execute the command.

cmd: The command line to start the process.

Here's an example for an app called algorithms:

dir="/var/apps/algorithms"

user="node"

cmd="node server.js"

Script usage

Start: Starts the app.

/etc/init.d/algorithms start

Stop: Stops the app.

/etc/init.d/algorithms stop

Restart: Restarts the app.

/etc/init.d/algorithms restart

Status: Tells you whether the app is running. Exits with 0 if it is and 1 otherwise.

/etc/init.d/algorithms status

Logging

By default, standard output goes to /var/log/scriptname.log and error output to /var/log/scriptname.err. If you're not happy with that, change the variables stdout\_log and stderr\_log.

<https://wiki.debian.org/LSBInitScripts>

A partir de él se crean en el subdirectorio */etc/init.d* dos ficheros de script : *anret\_mon\_s.sh* y *fronius\_mon\_s.sh*

Originalmente este script utiliza sudo para ejecutar el programa/daemon bajo un usuario dado. El programa/daemon se debe especificar en la variable cmd al inicio del programa. El usuario bajo el que se ejecuta se especifica en la variable user al inicio del programa. **Sin embargo en nuestro caso se ha retirado la ejecución bajo sudo , ya que bajo init al iniciar el sistema, lo hace con permisos de root.**

La salida stdout se redirige al fichero especificado en la variable stdout\_log. En nuestro caso se a redirigido a "/dev/null".

Se fuerza a que el propietario y grupo de los ficheros de script sean *root:root* y con permisos "*rwxr-xr-x*", igual que el resto de ficheros del subdirectrorio /etc/init.d.

# chown root:root /etc/init.d/fronius\_mon\_s.sh

# chmod 755 /etc/init.d/fronius\_mon\_s.sh

# chown root:root /etc/init.d/anret\_mon\_s.sh

# chmod 755 /etc/init.d/anret\_mon\_s.sh

Para indicar la ejecución automática en el arranque y cierre en la parada en el sistema, se ejecutan las órdenes:

# update-rc.d anret\_mon\_s.sh defaults

update-rc.d: using dependency based boot sequencing

insserv: warning: # update-rc.d fronius\_mon\_s.sh defaults

# update-rc.d: using dependency based boot sequencing

insserv: warning: script 'mathkernel' missing LSB tags and overrides script 'mathkernel' missing LSB tags and overrides

NOTAS:

Funciona en el orden deseado porque "anret" está antes que "fronius" (alfabeticamente???) --> hay que forzar orden de ejecución deseado con independencia del alfabético???.

~~Hay que ejecutar NTPDATE antes del inicio de los demonios.~~ Provisionalmente se ha incluido en el fichero /etc/rc.local la orden de ejecucion. Este fichero se ejecuta al final de cada runlevel.

# Anexo V. Instalación sistema operativo en Raspberry

1) Se descarga la imagen del sistema operativo Raspbian y se extrae del fichero ZIP.

2) Se copia la imagen a la tarjeta sd

En linux:

Find the name of the device of the plugged in SD-card. For instance I typed ls -la /dev/sd\* before and after plugging in the sd-card. I saw the date of /dev/sdc change and thus decided that that was the one Alternatively it is possible recognize the drive name with the help of sudo fdisk -l.

Find the place of the unzipped image, which was /home/username/Downloads/2012-10-28-wheezy-raspbian.img for me.

Type the following):

sudo dd if=/home/username/Downloads/2012-10-28-wheezy-raspbian.img of=/dev/sdc

and wait for the command to return. It may be quite a while, especially over usb2. Adding bs=4M to the dd can significantly improve performance

Resizing can be done from the Pi itself in the raspi-config program that starts automatically the first time you boot.

En Windows:

Se puede usar el programa ***Win32DiskImager*** con privilegios de administrador.

3) Se introduce la sd card en la RPI , se conecta a la red ethernet donde exista un servidor DHCP y se alimenta para que arranque. Pasados unos segundos la RPI tendrá asignada una dirección de IP. Para descubrir esta dirección será preciso interrogar al servidor de DHCP a través del mecanismo que él provea[[4]](#footnote-4). La RPI se identifica en la red con el nombre DNS "raspberrypi".

4) Utilizando el terminal Putty se accede a la RPI. Se puede emplear la IP asignada o, en el caso de que no exista una con el mismo nombre en la red, el nombre DNS "raspberrypi". Se debe utilizar el usuario "pi" y la contraseña "raspberry".

5) Se recomienda cambiar la comfiguración de la red de la RPI para asignarle un nuevo nombre y dirección IP fija. Se edita el fichero /etc/network/interfaces con lo siguiente:

iface eth0 inet static

address 192.168.1.8

netmask 255.255.255.0

gateway 192.168.1.1

6) Hasta que no se complete la configuración inicial de la RPI, esta mostrará el mensaje: "the software on this Raspberry Pi has not been fully configured. Please run 'sudo raspi-config' ". Se ejecuta el comando indicado.

Se expande el Filesystem. El sistema informará:

Root partition has been resized.

The filesystem will be enlarged upon the next reboot

Opciones de internacionalización:

Se configura el área geográfica: Europa -> Madrid

Opciones avanzadas:

Se cambia el nombre de la RPI en la red.

Se ajusta el tamaño de memoria para la GPU 64 -->16MB

7) Instalación del programa ntpdate

Para permitir la sincronización horaria con servidores de tiempo en Internet se instala el cliente del protocolo NTP (Network Time Protocol).

# apt-get install ntpdate

Se instalarán programas y librerías que ocuparan unos 410KB.

Al ejecutar el programa se obtiene el el siguiente error.

# ntpdate hora.roa.es

10 Sep 15:08:45 ntpdate[2860]: the NTP socket is in use, exiting

La razón es que está ejecutándose el servicio NTP (programa ntpd). Se puede parar el servicio con:

# service ntp stop

[ ok ] Stopping NTP server: ntpd.

Al reintentar se consigue la sincronización:

# ntpdate hora.roa.es

16 Oct 22:09:02 ntpdate[2895]: step time server 150.214.94.5 offset 3135343.405050 sec

El problema es que cuando reinicia RPI volverá a iniciarse el servicio NTP y la sincronización fallará otra vez. **Es necesario desactivar el arranque automático del servicio NTP**:

# update-rc.d ntp disable

update-rc.d: using dependency based boot sequencing

insserv: warning: current start runlevel(s) (empty) of script `ntp' overrides LSB defaults (2 3 4 5).

insserv: warning: current stop runlevel(s) (2 3 4 5) of script `ntp' overrides LSB defaults (empty).

Alternativamente se puede indicar al cliente ntpdate para que emplee un socket alternativo cuando el servicio NTP está arrancado:

# ntpdate -u hora.roa.es

16 Oct 22:15:07 ntpdate[2158]: step time server 150.214.94.5 offset 17.628070 sec

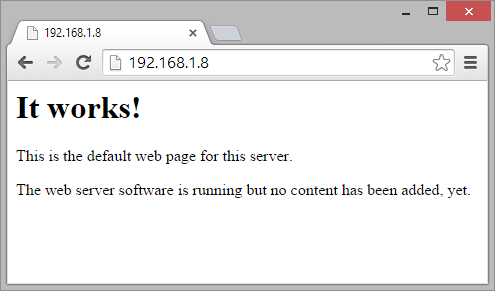
8) Instalación de Apache

Para instalar el servidor apache:

# apt-get install apache2 -y

Su instalación ocupara unos 1,4MB de espacio en el disco de la RPI

Una vez instalado se puede comprobar su funcionamiento desde un navegador en otra máquina de la red.



Más información en:

<http://www.raspberrypi.org/documentation/remote-access/web-server/apache.md>

9) Nombrado de dispositivos USB-Serial

A continuación se muestra la numeración de los puertos USB de la RPI modelos B y B+.

usb 1-1.**2**

usb 1-1.**3**

eth0 conectado a usb 1-1.**3**





usb 1-1.**3**

usb 1-1.**4**

usb 1-1.**2**

eth0 conectado a usb 1-1.1

usb 1-1.**5**

La nomenclatura 1-1.N significa: controlador usb 1- hub usb 1. número de puerto N del hub.

Por defecto los dispositivos serie se ver en el sistema de ficheros como /dev/USBtty*n.* Desgraciadamente un fichero /dev/USBtty*n* no siempre se corresponde con el dispositivo conectado a un puerto USB determinado, lo que hace que los procesos de monitorización puedan fallar al enviar comandos a dispositivos que están en otra linea-bus-red serie.

Para asegurar la constancia entre el nombre del dispositivo serie y el puerto USB se utiliza reglas de UDEV.

Se edita un fichero de reglas de UDEV /etc/udev/rules.d/10-local.rules con el siguiente contenido:

Para raspberry pi b (2 puertos USB):

KERNELS=="1-1.2", SUBSYSTEMS=="usb", SYMLINK+="rs485-modbus"

KERNELS=="1-1.3", SUBSYSTEMS=="usb", SYMLINK+="rs422-fronius"

De esta manera:

el adaptador usb-serie que se conecte al puerto USB 1-1.2 se verá en el sistema como /dev/rs485-modbus y será el que se conecte al analizador de red (o a la red de dispositivos MODBUS donde esté conectado el analizador )

el adaptador usb-serie que se conecte al puerto USB 1-1.3 se verá en el sistema como /dev/rs422-fronius y será el que se conecte al inversor (o red de inversores)

Por tanto es **absolutamente necesario fijar el puerto USB donde se conecta cada uno de los adaptadores** de acuerdo con lo definido en el fichero de reglas UDEVG ya que si se conectan en otros puertos USB los procesos de monitorización no funcionarán fiablemente.



# Anexo VI. Instalación de módulo RTC y configuración para sincronización horaria automática

Raspberry Pi no incorpora un reloj permanentemente alimentado que mantenga la hora cuando se apaga.

Parta evitar eso se hace uso de un módulo RTC (Real Time Clock) . Se ha empleado el modulo *DS3232 for PI* que incorpora una batería.

A continuación se explican los pasos para activar el módulo RTC Module en la distribución Raspbian.

**Conexión del módulo RTC al conector GPIO**

Se conecta el modulo RTC al conectro GPIO tal y como se muestra en la figura.

|  |  |
| --- | --- |
| http://ecx.images-amazon.com/images/I/61fBSyDoY0L._SL1000_.jpg | http://ecx.images-amazon.com/images/I/61XtSzFMn3L._SL1000_.jpg |

Activación bus i2c

Normalmente será preciso habilitar el protocolo I2C, que por defecto está deshabilitado al estar incluido el correspondiente módulo en una lista negra, concretamente en el fichero */etc/modprobe.d/raspi-blacklist.conf*. Por tanto se debe editar este fichero y comentar la línea que excluye al módulo insertando un carácter # en la primera posición

sudo nano /etc/modprobe.d/raspi-blacklist.conf

quedando el fichero:

blacklist spi-bcm2708

#blacklist i2c-bcm2708

Pruebas de funcionamiento del bus i2c y el módulo RTC

Load the module now

$ sudo modprobe i2c-bcm2708

Notify Linux of the Dallas RTC device

$ echo ds1307 0x68 | sudo tee /sys/class/i2c-adapter/i2c-1/new\_device

ds1307 0x68

Test whether Linux can see our RTC module.

$ sudo hwclock

Fri 09 Jan 2015 23:22:46 CET -0.896106 seconds

En caso de no tener acceso se obtendrá:

hwclock: Cannot access the Hardware Clock via any known method.

hwclock: Use the --debug option to see the details of our search for an access method.

|  |
| --- |
| Usage:  hwclock [function] [option...]  Functions:  -h, --help show this help text and exit  -r, --show read hardware clock and print result  --set set the RTC to the time given with --date  -s, --hctosys set the system time from the hardware clock  -w, --systohc set the hardware clock from the current system time  --systz set the system time based on the current timezone  --adjust adjust the RTC to account for systematic drift since  the clock was last set or adjusted  --getepoch print out the kernel's hardware clock epoch value  --setepoch set the kernel's hardware clock epoch value to the  value given with --epoch  --predict predict RTC reading at time given with --date  -V, --version display version information and exit  Options:  -u, --utc the hardware clock is kept in UTC  --localtime the hardware clock is kept in local time  -f, --rtc <file> special /dev/... file to use instead of default  --directisa access the ISA bus directly instead of /dev/rtc  --badyear ignore RTC's year because the BIOS is broken  --date <time> specifies the time to which to set the hardware clock  --epoch <year> specifies the year which is the beginning of the  hardware clock's epoch value  --noadjfile do not access /etc/adjtime; this requires the use of  either --utc or --localtime  --adjfile <file> specifies the path to the adjust file;  the default is /etc/adjtime  --test do not update anything, just show what would happen  -D, --debug debugging mode |

**Configuración para ajuste de tiempo con el RTC en el proceso de boot**

|  |
| --- |
| Este enlace parece que es el que hace las sugerencias más precisas desaconsejando la configuración en rc.local  <http://www.raspberrypi.org/forums/viewtopic.php?f=66&t=85683> |

Se añade el de reloj a los módulos del Kernel.

Se edita el fichero

# nano /etc/modules

y se añade

rtc-ds1307

Se reconfigura el script hwclock.sh

# nano /etc/init.d/hwclock.sh

Después de "unset TZ" en la parte superior, se añade la función:

init\_rtc\_device()  
{  
  [ -e /dev/rtc0 ] && return 0;  
  
  # load i2c and RTC kernel modules  
  modprobe i2c-dev  
  modprobe rtc-ds1307  
  
  # iterate over every i2c bus as we're supporting Raspberry Pi rev. 1 and 2  
  # (different I2C busses on GPIO header!)  
  for bus in $(ls -d /sys/bus/i2c/devices/i2c-\*);  
  do  
    echo ds1307 0x68 >> $bus/new\_device;  
    if [ -e /dev/rtc0 ];  
    then  
      log\_action\_msg "RTC found on bus `cat $bus/name`";  
      break; # RTC found, bail out of the loop  
    else  
      echo 0x68 >> $bus/delete\_device  
    fi  
  done  
}

Se localiza 'case "$1" in' y se edita lo que está en negrita:

case "$1" in

start)

# If the admin deleted the hwclock config, create a blank

# template with the defaults.

if [ -w /etc ] && [ ! -f /etc/adjtime ] && [ ! -e /etc/adjtime ]; then

printf "0.0 0 0.0\n0\nUTC" > /etc/adjtime

fi

**init\_rtc\_device**

**# Raspberry Pi doesn't have udev detectable RTC**

**#**if [ -d /run/udev ] || [ -d /dev/.udev ]; then

**#**return 0

**#**fi

..........

Se actualiza el real HW Clock y se deshabilita (remove) el "fake".

# update-rc.d hwclock.sh enable

update-rc.d: using dependency based boot sequencing  
# update-rc.d fake-hwclock remove

update-rc.d: using dependency based boot sequencing

Se elimina el paquete fake-hwclock y las configuraciones de cron correspondientes.

# apt-get remove fake-hwclock

# rm /etc/cron.hourly/fake-hwclock

# rm /etc/init.d/fake-hwclock

Finalmente se realizar un reinicio del sistema

# reboot

**Configuración para sincronización horaria automática**

Para mantener la precisión horaria de los registros del sistema de monitorización se automatiza la sincronización diaria del tiempo del reloj de sistema con un fuente de tiempo de de internet e inmediatamente después la sincronización con el reloj hardware proporcionado por el modulo RTC.

Se configura crontab bajo el usuario root.

Haciendo uso del editor crontab:

# crontab -e

Se añaden las líneas (importante poner el path absoluto al comando):

50 23 \* \* \* /usr/sbin/ntpdate hora.roa.es

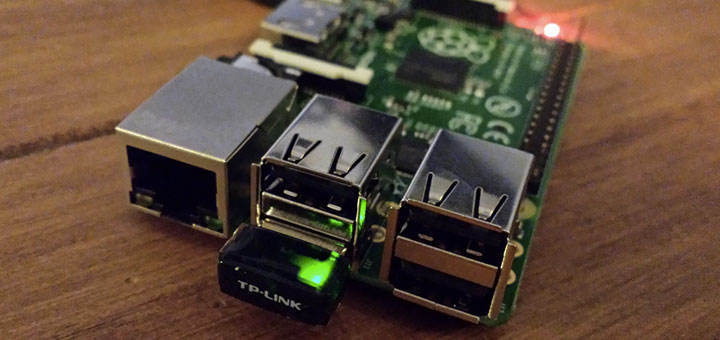
51 23 \* \* \* /sbin/hwclock -w

De esta manera se ejecuta la sincronicación con Internet a las 23:50 y un minuto más tarde se sincroniza el reloj hardware.

# Anexo VI Instalación y configuración de pincho USB Wifi TP-LINK

Opcionalemente se puede realizar una conexión a red a través de un segmento de red wifi.

Se emplea el adaptador TP-Link WN72n v2.1



Para instalar el driver (sobre versión Raspbiam 3.12.28+):

# wget https://dl.dropboxusercontent.com/u/80256631/8188eu-20140908.tar.gz  
# tar -zxvf 8188eu-20140908.tar.gz  
# sudo install -p -m 644 8188eu.ko /lib/modules/$(uname -r)/kernel/drivers/net/wireless  
# sudo insmod /lib/modules/$(uname -r)/kernel/drivers/net/wireless/8188eu.ko  
# sudo depmod -a

Para configurar el interfaz wifi (wlan0) se edita el fichero de configuración de la red /etc/network/interfaces como sigue:

……

allow-hotplug wlan0

iface wlan0 inet static

address 192.168.1.118

netmask 255.255.255.0

gateway 192.168.1.1

wpa-ssid "casitademartolas"

wpa-psk "carmenguilsalvador"

Nota: Para descubrir los ssids detectados por el adaptador se puede emplear el comando:

# iwlist wlan0 scan

# Anexo VII Monitores de red (multifucntion meter) MODBUS alternativos

FORLONG DRS-202M

precio 36€ en aliexpres + 20€ transporte

<http://www.china-meters.com/En/product_show.aspx?id=49&refId=19>



Tiene el protocolo MODBUS publicado en:

<http://www.bg-etech.de/_picts/drs202m/DRS202M__Modbus_register_Map.pdf>

HIKING DDS238-1 ZN

Single Phase, Multifunction Energy Meter, 32Amax. Medida Bireccional opcional.



http://www.chinaenergymeter.com/1-1-5-single-phase-multifunction-energy-meter.html

Venta internet 36€ : http://www.aliexpress.com/item/DDS238-1-single-phase-din-rail-type-energy-meter/671491862.html

HUABANG DDM100SCR



<http://www.china-huabang.com/en/product-show.asp?id=22&tt=101>

25€ + 14€ de envío

EPP KWH1C32

Monófasico no bidireccional. Medida directa 32A max. Multifuncion (analizador red). 76€.



<http://www.direct-electro.es/files/productos/epp/Contador_energia_KWH1C32.pdf>

SCHNEIDER EM3500



<http://download.schneider-electric.com/files?p_File_Id=462992762&p_File_Name=PLSED310035EN_(web).pdf>

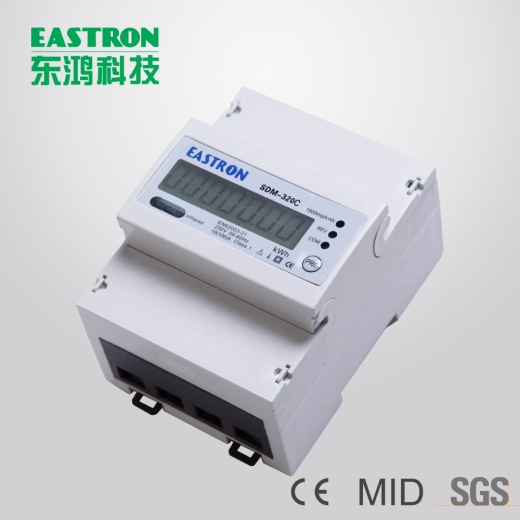
EASTRON SDM320M

Monófasico. Medida directa. Multifuncion (analizador red).



EASTRON SDM320C

Monofasico. Medida directa. Solo medidor energía



EASTRON SDM630

Trifasico. Medida indirecta con CTs.

EASTRON SDM630 DC



Trifásico. Medida directa. Precio 85€

http://www.eastrongroup.com/uploads/Eastron%20SDM630%20DC%20user%20manual%202013.pdf

http://bg-etech.de/os/product\_info.php/cPath/25\_28/products\_id/195

Selec MFM384



Trifásico, Medida indirecta con CTs

http://www.selec.com/Content/DownlodableFiles/Datasheet/Meter/Multifunction%20Meter/MFM384-R-C.pdf

1. No siempre es así, por ejemplo la respuesta de la function 0x15 es diferente [↑](#footnote-ref-1)
2. Para calcular el CRC se puede emplear el siguiente calculador on-line:  
    <http://www.lammertbies.nl/comm/info/crc-calculation.html> Ojo el resultado del CRC hay que ponerlo en "little endian". [↑](#footnote-ref-2)
3. Se observó que si no se incluía -opost, cuando se enviaba un 0x0A (NL) el driver enviaba a la línea dos caracteres (0x0A 0x0A) por lo que se desalineaba la trama dando un error el inversor. Posteriormente se descubrió el empleo de la opción "raw" evitaba que eliminación a su recepción de ciertos caracteres con el 0x11 (17 dec, device control 1) y el 0x13 (19 dec, device control 3). [↑](#footnote-ref-3)
4. En el caso de dnsmask el fichero de concesiones es /var/lib/misc/dnsmasq.leases [↑](#footnote-ref-4)