

E.T.S. de Ingeniería Industrial, Informática y de
Telecomunicación

Monitorización del consumo eléctrico de un hogar: procesado de datos mediante Raspberry Pi



Grado en Ingeniería Eléctrica y Electrónica

Trabajo Fin de Grado

Aritz Legarrea Oyarzun

Javier Marcos Álvarez

Pamplona, 30 de junio del 2015

RESUMEN

Hoy en día, el ahorro y la eficiencia energética son tendencias que cada vez están cobrando más importancia. Debido a esto, el uso de contadores inteligentes en hogares se está extendiendo cada vez más. Un contador inteligente permite al usuario conocer en cualquier instante el consumo de electricidad del hogar mediante la representación de los datos medidos en diferentes gráficas. Esto permite al consumidor incrementar notoriamente la eficiencia energética mediante pequeños cambios de hábitos.

Se ha realizado un proyecto en el que se ha diseñado e implementado un contador inteligente. El proyecto se divide en el sensado de la corriente, procesado de datos medidos y visualización de los datos procesados en página web. Este trabajo se encarga de la parte de procesado de datos, el que se realiza mediante el uso de una Raspberry Pi. Dentro del procesado de datos pueden diferenciarse tres funciones generales:

En primer lugar, se realiza la lectura de corriente instantánea proporcionada por el sensado de la corriente. En segundo lugar, cada 10 minutos, se realiza un promedio de la corriente eficaz que ha habido durante esos 10 últimos minutos. Por último, los datos de valor eficaz de la corriente calculados son subidos a una nube de datos.

LISTA DE PALABRAS CLAVE

Contador inteligente

Raspberry Pi

Convertidor analógico-digital

Google-Drive

Corriente

ABSTRACTS

Nowadays, energy saving and efficiency are tendencies that are gaining more importance. Therefore, the use of smart meters in households is progressively spreading. A smart meter allows the user to know the electricity consumption of the household at any time by graphic representations of the data. In that way, consumers are able to increase energy efficiency through slight change of habits.

A project has been developed in which a smart meter has been designed and installed. The project has been divided in the current sensing, measured data processing and display of the processed data on a web page. This work is in charge of the data processing that it is made using a Raspberry Pi. There are three main functions in the data processing:

First, the reading of the instant current, provided by the current sensing, is done. Then, every 10 minutes, an average of the RMS current consumed in those 10 minutes is calculated carried out. Finally, the calculated RMS values of the current are uploaded to the cloud.

KEYWORDS

Smart meters

Raspberry Pi

Analog to digital converter

Google-Drive

Current

LABURPENA

Gaur egun, eraginkortasun energetikoa eta energia aurrezteko gero eta garrantzi handiagoa hartzen ari diren joerak dira. Hori dela eta, etxebizitzetan kontagailu berriak erabiltzeko joera gero eta gehiago zabaltzen ari da. Kontagailu berri hauek, etxebizitzan kontsumitzen ari den elektrizitatea une oro ezagutzeko aukera ematen diote erabiltzaileari, horretarako, grafiko ezberdinen bitartez neurtutako datuen irudikapena egiten dute. Honek, kontsumitzaileari, ohitura aldatzea txiki-ekin, eraginkortasun energetikoa nabarmenki areagotzeko aukera ematen diote.

Proiektuan kontagailu adimendu baten diseinua eta hornikuntza garatu dira. Proiektua hurrengo ataletan zatitzen da: korrontearen neurketa, datu neurtuen prozesamendua eta prozesatutako datuen irudikatzea web orri batean. Lan honi dagokion atala “datuen prozesamendua” da, Raspberry Pi baten erabileraren bitartez egiten dena. Datu prozesamenduaren barruan hiru funtzio orokor bereiz daitezke:

Lehenik eta behin korrontearen neurketak eskainitako bat-bateko korrontearen irakurketa egiten da. Bestalde, 10 minuturo, azken 10 minutu horietan egon den korrante eraginkorraren batzbestekoa egiten da. Azkenik, korrontearen balio eraginkorra duten datuen kalkuluak datu hodei batera igotzen dira.

GAKO-HITZEN ZERRENDA

Kontagailu adimendua

Raspberry Pi

Bihurtzaile analogiko-digitala

Google-Drive

Korrontea

ÍNDICE

1. INTRODUCCIÓN	1
1.1. OBJETIVO DEL PROYECTO.....	1
2. ESTADO ACTUAL DE LA CUESTIÓN	1
3. REPARTO DEL TRABAJO	3
4. DECISIONES GENERALES.....	3
5. RASPBERRY PI	4
5.1. INTRODUCCIÓN	4
5.2. PRIMER ENCENDIDO Y PRIMEROS PASOS EN LA RASPBERRY PI	5
5.3. COMANDOS BÁSICOS PARA EL USO DE LA RASPBERRY	6
5.4. CONEXIÓN A INTERNET DE LA RASPBERRY PI	7
5.5. TERMINALES GPIO	7
6. MEDIDA DE LA CORRIENTE.....	10
6.1. CIRCUITO DE SENSADO	10
6.2. CONVERTIDOR ANALÓGICO-DIGITAL	11
6.3. CONEXIÓN DEL CONVERTIDOR ANALÓGICO-DIGITAL A LA RASPBERRY PI.....	13
7. PROGRAMACIÓN DE LA RASPBERRY PI	14
7.1. CARGA DE LAS LIBRERÍAS NECESARIAS	16
7.2. OBTENCIÓN DE CREDENCIALES.....	17
7.3. DEFINICIÓN DE LAS FUNCIONES Y ASIGNACIÓN DE VALORES INICIALES A VARIABLES.....	18
7.3.1. Lectura del dato proporcionado por el convertidor analógico digital.....	18
7.3.2. Obtención del valor de tensión que proporciona la son.....	18
7.3.3. Medida de la corriente instantánea.....	19
7.3.4. Cálculo del cuadrado de la corriente instantánea.....	19
7.3.5. Cálculo del valor eficaz de la corriente.....	19
7.3.6. Paso de la hora del día a segundos.....	20
7.4. BUCLE INFINITO	22
8. ENSAYO FINAL	24
9. LÍNEA TEMPORAL	26
10. PRESUPUESTO	27
11. PROBLEMAS ENCONTRADOS	27



12. COMPARATIVA DE RASPBERRY PI CON ARDUINO	28
13. CONCLUSIONES Y LÍNEAS FUTURAS	29
13.1. CONCLUSIONES	29
13.2. LÍNEAS FUTURAS	30
14. REFERENCIAS	31
ANEXOS	32
ANEXO I	32
ANEXO II	34
ANEXO III	34

ÍNDICE DE FIGURAS Y TABLAS

FIGURAS

Figura 1. Tipo de contador del mercado	2
Figura 2. Tipo de contador del mercado	2
Figura 3. Tipo de contador del mercado	2
Figura 4: Raspberry Pi 2 Model B	5
Figura 5. Terminales GPIO	7
Figura 6. Configuración de los GPIO	9
Figura 7. Circuito de sensado	10
Figura 8. Pines convertidor analógico digital	12
Figura 9. Conexión convertidor analógico-digital pines GPIO	13
Figura 10. Circuito de acondicionamiento de señal completo	14
Figura 11. Diagrama de flujos	15
Figura 12. Corriente medida por el sensor	18
Figura 13. Onda de tensión generada por el sensor.	19
Figura 14. Onda de tensión muestreada	20
Figura 15. onda de corriente muestreada	20
Figura 16. Cuadrados de corriente	21
Figura 17. Producto final	24
Figura 18. Representación de los datos del ensayo	25

TABLA

Tabla 1	27
Tabla 2. Presupuesto total	27
Tabla 3. Comparativa	29

1. INTRODUCCIÓN

El Real Decreto 110/2007 obliga a las distribuidoras a sustituir los contadores actuales por equipos inteligentes en el caso de los usuarios con una potencia igual, o menor, a 15 kW. El 31 de diciembre de 2018 todos los clientes domésticos deberán tener un contador inteligente integrado en el sistema de telegestión de su distribuidora. Un medidor inteligente o contador inteligente es un tipo de medidor o contador eléctrico avanzado que calcula el consumo de energía eléctrica de una forma más detallada que los contadores convencionales. Ya que estos nuevos contadores permiten al usuario conocer los datos sobre su consumo a través de gráficas donde se representan los datos medidos por éste.

Por otro lado, Internet se ha convertido en una herramienta muy importante en la mayoría de los ámbitos de trabajo. Estos aparatos también ofrecen la posibilidad de comunicar esta información a través de alguna red a un centro de control de la compañía de servicio local, la cual puede utilizar los datos a efectos de facturación o seguimiento.

1.1. OBJETIVO DEL PROYECTO

El objetivo de este trabajo es diseñar e implementar un contador inteligente que realice continuamente la medición del consumo de energía de una instalación. El contador que se va a realizar estará orientado hacia la investigación y, principalmente, hacia el uso doméstico. Ya que este contador facilitará a los usuarios que lo instalen en sus viviendas datos sobre el consumo eléctrico como el consumo de potencia instantáneo, el consumo eléctrico del último día, el consumo mensual, anual... y de esta forma, el usuario podrá aumentar la eficiencia energética y disminuir el importe de su factura.

Se intentará fabricar un contador inteligente de bajo coste, el cual sea de fácil instalación, para que su uso no quede limitado únicamente a personas que tengan conocimientos sobre electricidad. Además de esto, los datos obtenidos por el contador inteligente serán enviados a una página web a la que el usuario podrá acceder cuando y donde quiera, pudiendo conocer el consumo de su hogar en todo momento.

2. ESTADO ACTUAL DE LA CUESTIÓN

El ahorro energético es un concepto que cada vez está cobrando más importancia hoy en día. Debido a esto, diversas empresas han desarrollado contadores inteligentes con un fin similar al de este trabajo, es decir, que proporcione información sobre el consumo eléctrico doméstico. De hecho, en el mercado ya se dispone de diferentes contadores inteligentes. Lo que diferencia unos contadores de otros, principalmente, es la forma que tienen éstos de mostrar los datos. Según esto, se diferencian tres tipos de contadores:

1. Visualización de datos en pantalla LCD: este tipo de contador es el más simple del mercado, ya que solamente muestra los datos del consumo en una pantalla LCD que va incorporada al contador. También suelen contener diferentes alarmas de aviso. Es una alternativa interesante para personas que no tienen conocimientos tecnológicos o cuyo interés sea general y no de datos específicos.



Figura 1. Tipo de contador del mercado

2. Visualización de datos en pantalla LCD y exportación de datos a un ordenador: este otro modelo de contador es igual que el anterior. La única diferencia entre ambos es que este modelo lleva incorporado un puerto USB, que da la opción de exportar los datos medidos por el contador a un ordenador, en donde se podrán visualizar gráficas sobre los datos de consumo.



Figura 2. Tipo de contador del mercado

3. Visualización de datos vía página web: este tipo de contador es el más avanzado. Ofrece gran variedad de datos, de los que se puede disponer en cualquier momento, ya que éstos son mostrados en una página web.



Figura 3. Tipo de contador del mercado

El contador que se va a desarrollar en este trabajo se puede decir que está dentro del tercer tipo de contadores inteligentes explicados, ya que los datos obtenidos se mostrarán en una página web al igual que en dicho tipo.

3. REPARTO DEL TRABAJO

Para proporcionarnos la información deseada sobre el consumo eléctrico, las operaciones que realiza el aparato puede clasificarse o resumirse en tres funciones generales. A partir de estas tres funciones generales, el trabajo se ha dividido en tres partes, asignando una función de éstas a diferentes personas:

1. Medida de la corriente que se está consumiendo en la vivienda: esta medida de la corriente se realizará mediante un circuito de sensado. Dicho circuito ha sido llevado a cabo por mi compañero Joseba Revuelta Irisarri.
2. Lectura de las medidas obtenidas por el circuito de sensado y posterior envío del valor del promedio de la corriente eficaz de las medidas de corriente obtenidas a un sistema de almacenamiento de datos en la red (nube de datos). Esta parte se realizará de dos maneras diferentes: en una de ellas, de la que se ha encargado mi compañero Víctor Erice Carbonero, se utilizará un arduino para llevar a cabo la función descrita. Mientras que en la otra, la que se explica en este trabajo, se utilizará una Raspberry Pi para realizarla.
3. Creación de una página web en la que se muestren los datos del consumo eléctrico de la vivienda. Los datos sobre el consumo se obtienen de la nube de datos. Primero, mediante *query language* reference se crean cinco tablas diferentes sobre datos subidos a la nube. Después, a partir de las cinco tablas creadas y mediante el uso de R (utilizando la plataforma shiny) se crea la página web. Esta parte del trabajo ha sido desarrollada por mi compañera Nahia Barriola Hernandorena. La dirección url de la página web para visualizar los datos es <https://consumoenergia.shinyapps.io/definitivo>. [1]

4. DECISIONES GENERALES

Antes de comenzar a trabajar cada uno en la parte asignada del trabajo, se han tomado una serie de decisiones acerca del contador que se va a diseñar. El diseño e implementación se realizarán en base a estas decisiones, que son las siguientes:

- El circuito de sensado únicamente medirá la corriente que se está consumiendo. Se aparta la idea de medir la tensión, ya que, aunque proporcionaría la capacidad para medir el factor de potencia y por lo tanto se podrían ofrecer mayor variedad de datos y de mayor precisión, la instalación del contador se complicaría, cuando una de las características que se pretende que tenga este contador es que sea de fácil instalación. Mientras que para realizar la medida de la corriente no hace falta abrir el circuito para colocar el sensor, sino que basta con colocar una pinza sobre el cable.
- Dado que la onda de corriente de la red tiene una frecuencia de 50 Hz, su periodo es de 20ms. Por lo tanto, tanto la Raspberry Pi como el arduino tomarán datos de la corriente instantánea cada 1 ms, obteniendo 20 muestras por cada onda. Por lo que la onda quedará suficientemente definida.
- El contador inteligente a desarrollar irá colocado en el cuadro eléctrico de la vivienda, ya que el sensor de corriente deberá ir colocado en el cable que entra al cuadro eléctrico. Dado que la distancia entre el cuadro eléctrico y el lugar donde se coloca el router suele ser bastante grande, se ha decidido que la conexión a internet sea realizada por WiFi. De esta manera, se evita el inconveniente de tener que llevar un cable Ethernet desde el router hasta el contador. En el caso del Arduino resulta muy cara la utilización de WiFi, con lo que se ha

decidido usar dos Arduinos que se comunican por radiofrecuencia. El primer Arduino va colocado en el cuadro eléctrico y manda al segundo Arduino los datos de corriente por radiofrecuencia. El segundo Arduino, sube los datos de corriente a internet.

- Aunque se realiza la medición constante de la corriente consumida en la instalación, se ha decidido subir valores eficaces de la corriente medida cada diez minutos. Se cree que con esta frecuencia de subida de datos se obtiene un buen compromiso entre la no excesiva subida de datos a la nube y la inexistencia de pérdidas de información acerca del consumo.
- Se han valorado dos tipos de almacenamiento diferentes, hoja cálculo o base de datos. Al final se ha elegido utilizar la hoja de cálculo de Google-Drive por su simplicidad y por el conocimiento que se posee sobre la misma.
- A partir de la corriente medida por el sensor, se mostrarán los siguientes datos acerca del consumo: potencia instantánea, consumo de energía diario, consumo mensual de energía, consumo anual de energía, datos significativos (día del año de mayor consumo, mes del año de menor consumo...), factura eléctrica y consejos que pueden aplicar los usuarios para disminuir el consumo eléctrico.

5. RASPBERRY PI

5.1. INTRODUCCIÓN

Raspberry Pi es una placa computadora (SBC) de bajo costo desarrollada en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

En realidad, se trata de una diminuta placa base de 85 x 54 milímetros (del tamaño aproximado de una tarjeta de crédito) en el que se aloja un chip Broadcom BCM2835 con procesador ARM hasta a 1 GHz de velocidad (modo Turbo haciendo overclock), GPU VideoCore IV y 512 Mbytes de memoria RAM (Las primeras placas contaban con sólo 256MB de RAM).

Para que funcione, necesita de un medio de almacenamiento (Raspberry Pi utiliza tarjetas de memoria SD o microSD), conectarlo a la corriente utilizando cualquier cargador microUSB de al menos 1000mah para las placas antiguas y de al menos 2000mah para las modernas (sobre todo si va a utilizarse un adaptador inalámbrico WIFI), y si se desea, guardarlo todo utilizando una carcasa para que todo quede a buen recaudo y su apariencia sea más estética.

En función del modelo que se escoge, se dispone de más o menos opciones de conexión, aunque siempre se dispone de al menos un puerto de salida de video HDMI y otro de tipo RCA, minijack de audio y un puerto USB 2.0 (modelos A y A+, B dispone de dos USB y B+ y Raspberry Pi 2 disponen de 4 USB) al que conectar un teclado y ratón. [2]

En cuanto a la conexión de red, se dispone de un puerto Ethernet (los modelos A y A+ no disponen de puerto Ethernet) para enchufar un cable RJ-45 directamente al router o se puede recurrir a utilizar cualquier adaptador inalámbrico WiFi compatible.

Para este proyecto se va a utilizar el modelo Raspberry pi 2, ya que de esta forma, al disponer de cuatro puertos USB, se puede utilizar teclado, ratón y adaptador inalámbrico simultáneamente, evitando la necesidad de cable para conectarse a internet. Además de contar con 1GB de memoria RAM, en lugar de los 512MB de memoria de los últimos modelos anteriores a la Raspberry pi 2.

En cuanto al software, existe una gran variedad de sistemas operativos (NOOBS, BerryBoot, Raspbian, OpenELEC, Windows 10 Pi 2...). Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux. Para este trabajo se ha elegido la el sistema operativo Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi.



Figura 4: Raspberry Pi 2 Model B

5.2. PRIMER ENCENDIDO Y PRIMEROS PASOS EN LA RASPBERRY PI

Una vez que se ha obtenido todo el hardware necesario para poner en funcionamiento la Raspberry Pi (ratón, teclado, adaptador WIFI, cargador micro-USB, tarjeta micro-SD y pantalla), lo primero que se debe hacer es instalar el sistema operativo en la tarjeta micro-SD. El sistema operativo elegido para este proyecto es Raspbian, una modificación de Debian, el cual se puede descargar de la página raspberrypi.org, en la sección downloads. En esta página, además de Raspbian se pueden encontrar una gran variedad de sistemas operativos que se pueden utilizar en la Raspberry pi. Una vez en esa página, se debe descargar el documento zip y descomprimir la imagen. Por otro lado, es necesario otro software llamado Win32 disk imager, que habrá que descargar e instalar para poder grabar el sistema operativo en la tarjeta micro-SD.

Una vez grabado el sistema operativo en la tarjeta micro-SD, se debe conectar todo el hardware. Una vez que se conecta la Raspberry Pi a la corriente, ésta se enciende. La primera vez que se enciende la Raspberry Pi, se muestra un menú de configuración, al cual se puede acceder en cuando se desee mediante el comando `raspi-config`. En este menú de configuración se pueden realizar cambios como el nombre de usuario, contraseña...

Tras terminar la configuración, se reinicia la Raspberry Pi. Al iniciar la Raspberry Pi, se pide un nombre de usuario y contraseña, que por defecto son las siguientes:

- Nombre de usuario: pi
- Contraseña: raspberry

Aunque, como se ha comentado antes, se pueden cambiar en el menú de configuración, al que se accede ejecutando el comando `raspi-config` en la consola.

Por último, se deben ejecutar dos comandos para actualizar la Raspberry Pi. Estos dos comandos a ejecutar son los siguientes:

1. `sudo apt-get update`: descarga y actualiza las definiciones de paquetes de `apt-get`.

Cuando termina de actualizar, se ejecuta el segundo comando:

2. `sudo apt-get upgrade`: actualiza los últimos paquetes de la Raspberry Pi.

Con estos dos comandos, se asegura que se tiene la última versión.

Una vez realizado esto, la Raspberry Pi está lista para utilizarse. Para más información sobre la primera configuración de la Raspberry pi, se puede consultar el video tutorial “*Iniciando tu Raspberry Pi Type B+ por primera vez*” de la página web youtube [3], donde se explican más detalladamente los pasos anteriormente citados. Aunque el video está orientado para una Raspberry Pi B+, los pasos a seguir en la Raspberry Pi 2 son los mismos, por lo que es totalmente válido para los dos modelos.

5.3. COMANDOS BÁSICOS PARA EL USO DE LA RASPBERRY

Al iniciar la Raspberry Pi, y tras introducir el nombre de usuario y contraseña, el usuario se encuentra en la consola, donde se ejecutan los comandos para realizar diferentes acciones. Los comandos básicos para comenzar a usar una Raspberry Pi son los siguientes:

- `sudo`: ejecución de comandos como superusuario.
- `apt-get update`: actualiza la lista de paquetes de repositorios.
- `apt-get upgrade`: instala las actualizaciones de los paquetes de la lista anterior.
- `raspi-config`: para ir al menú de configuración.
- `startx`: acceso a la interfaz gráfica.
- `Mkdir nombre de directorio`: crear directorio con el nombre *nombre de directorio*
- `cd: nombre de directorio`: acceder dentro del directorio *nombre de directorio*.
- `ls`: muestra la lista de archivos y carpetas del directorio en el que te encuentras.
- `cd..`: volver al directorio anterior.
- `nano/ruta/nombre de archivo`: editor de textos para crear o modificar archivos.
- `apt-get install nombreprograma`: instalar el programa *nombreprograma* desde la terminal.
- `reboot`: reiniciar el sistema.
- `poweroff`: apagar el sistema.

5.4. CONEXIÓN A INTERNET DE LA RASPBERRY PI

Existen tres formas de conectar la Raspberry Pi a internet: realizar la conexión con un cable Ethernet, utilizar un adaptador WiFi o conocer la dirección IP de la Raspberry Pi. Como ya se ha comentado antes, para este trabajo, la mejor opción es utilizar un adaptador WiFi. En este apartado se explicará cómo conectar la Raspberry Pi a internet por WiFi mediante la configuración de WiFi por medio de la interfaz gráfica.

Para conectarnos debemos acceder a menú, preferencias, WiFi Config. Al entrar en WiFi Config. aparecerá una ventana sobre el estado de la conexión. En la primera conexión que se realice habrá que pulsar sobre Scan, lo que abrirá una segunda ventana donde aparecerá una lista de redes WiFi. A continuación, se debe pinchar sobre la red a la que se quiere conectar. Tras introducir la contraseña de la red y darle a Add (agregar), en la primera ventana se podrá apreciar que la Raspberry Pi está conectada a la red (la conexión está establecida cuando la dirección IP aparece en la parte inferior de la ventana). En la primera ventana también se da la opción de conectar y desconectarse de internet. Por último, comentar que el proceso descrito únicamente habrá que realizarlo la primera vez. De ahí en adelante, la Raspberry Pi realizará el proceso automáticamente. [4]

5.5. TERMINALES GPIO

Lo que distingue a la Raspberry Pi de otros Ordenadores de Placa reducida (es decir, un ordenador entero encima de una placa) es la posibilidad de utilizar los pines GPIO para controlar sensores (IR, de video, la temperatura, la orientación de 3 ejes, aceleración...), servomotores, motores de corriente continua (mediante PWM) y dispositivos externos.

Los GPIO (*General Purpose Input/Output*, Entrada/Salida de Propósito General) son pines que actúan como una interfaz física entre la Raspberry Pi y el mundo exterior. Estos pines pueden configurarse de dos maneras, como salidas o como entradas.



Figura 5. Terminales GPIO

En la configuración de los pines GPIO como salidas, los pines realizan la función de un interruptor con una batería conectada en serie, pudiendo estar en nivel alto (interruptor cerrado), proporcionando una tensión de 3.3V, o en nivel bajo (interruptor abierto), proporcionando 0V de tensión.

En la configuración de los pines GPIO como entradas, los terminales pueden ser alimentados por con tensiones de 0 a 3.3V. La configuración como entrada de los GPIO se utiliza en funciones como lectura de sensores etc.

Entre los terminales GPIO encontramos también otros dos tipos de pines. Estos dos tipos de pines son, por un lado, los SPI (del inglés *Serial Peripheral Interface*), y por otro lado, los I2C.

El Bus SPI es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación síncrona).

Incluye una línea de reloj, dato entrante, dato saliente y un pin de *chip select*, que conecta o desconecta la operación del dispositivo con el que uno desea comunicarse. De esta forma, este estándar permite multiplexar las líneas de reloj.

La ventaja de un bus serie es que minimiza el número de conductores, pines y el tamaño del circuito integrado. Esto reduce el coste de fabricar, montar y probar la electrónica. El hardware consiste en señales de reloj, data in, data out y chip select para cada circuito integrado que tiene que ser controlado. Casi cualquier dispositivo digital puede ser controlado con esta combinación de señales.

El SPI es un protocolo síncrono. La sincronización y la transmisión de datos se realizan por medio de 4 señales:

- SCLK (*Clock*): es el pulso que marca la sincronización. Con cada pulso de este reloj, se lee o se envía un bit.
- MOSI (*Master Output Slave Input*): salida de datos del Master y entrada de datos al Slave. También llamada SIMO.
- MISO (*Master Input Slave Output*): Salida de datos del Slave y entrada al Master. También conocida por SOMI.
- SS/Select: para seleccionar un Slave, o para que el Master le diga al Slave que se active. También llamada SSTE.

La cadena de bits es enviada de manera síncrona con los pulsos del reloj, es decir, con cada pulso, el Master envía un bit. Para que empiece la transmisión, el Master baja la señal SSTE ó SS/Select a cero, con esto el Slave se activa y empieza la transmisión, con un pulso de reloj al mismo tiempo que el primer bit es leído.

En cuanto a los I²C, decir que son buses de comunicaciones en serie. Su nombre viene de *Inter-Integrated Circuit* (Inter-Circuitos Integrados). La velocidad de éstos es de 100kbit/s en el modo estándar, aunque también permite velocidades de 3.4 Mbits/s. Es muy usado para comunicar circuitos integrados entre sí que normalmente residen en un mismo circuito impreso.

La principal característica de I²C es que utiliza dos líneas para transmitir la información, una para los datos y otra para la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como suelen comunicarse circuitos en una misma placa que comparten una misma masa, esta tercera línea no suele ser necesaria.

Las líneas se llaman:

- SDA: datos
- SCL: reloj
- GND: tierra

La Raspberry Pi 2 posee 40 terminales en total, siendo el pin número 1 el más cercano a la tarjeta SD. A continuación se muestra la configuración de los pines de la Raspberry Pi utilizada en este proyecto (Raspberry Pi 2 Model B):

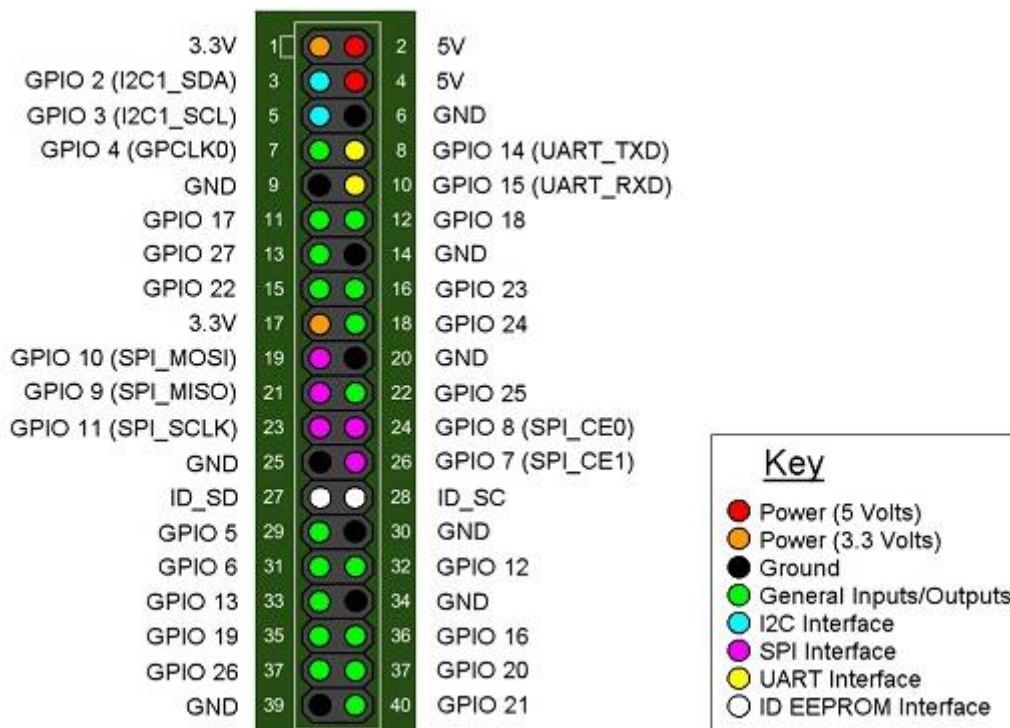


Figura 6. Configuración de los GPIO

El problema de los terminales GPIO es que son entradas y salidas digitales, mientras que las medidas proporcionadas por los sensores que miden las variables de interés del mundo real (temperatura, humedad...) son analógicas. Por lo tanto, se requerirá la utilización de un convertidor analógico-digital para pasar la señal analógica medida por los sensores a digital. De este modo, obtendremos una señal que podrá ser procesada por la Raspberry Pi.

6. MEDIDA DE LA CORRIENTE

Para que la Raspberry Pi pueda obtener el valor de la corriente, ésta deberá ser medida y acondicionada para transmitírselo a la Raspberry Pi a través de sus terminales GPIO.

6.1. CIRCUITO DE SENSADO

El circuito de sensado de la corriente está compuesto por un sensor de corriente y un pequeño circuito de acondicionamiento de la señal.

El sensor es una pinza que rodea el cable por el cual pasa la corriente que se quiere medir. Este sensor, al fin y al cabo, es un transformador que genera una corriente en el devanado de salida debido al paso de la corriente por el cable que rodea. La corriente que se genera pasa por una pequeña resistencia, con lo que a la salida del sensor se obtiene una tensión proporcional a la corriente que se quiere medir.

El circuito de acondicionamiento se basa en un divisor de tensión para la obtención de 1.65V. Esto se realiza para fijar los 1.65 V como los 0 A de la onda de corriente que se mide. Ya que las medidas que se dan a la Raspberry Pi deben ser en continua, y ésta puede ser alimentada con un máximo de 3.3V.

Visto esto, a la salida del circuito de sensado de corriente se obtiene una onda senoidal de tensión proporcional a la de la corriente medida y centrada en 1.65 V. Esta onda de tensión se llevará a uno de los canales de entrada del convertidor analógico-digital mediante la conexión de la salida del sensor con dicha entrada del convertidor analógico-digital. En la figura 7 se muestra el dibujo del circuito de sensado. [5]

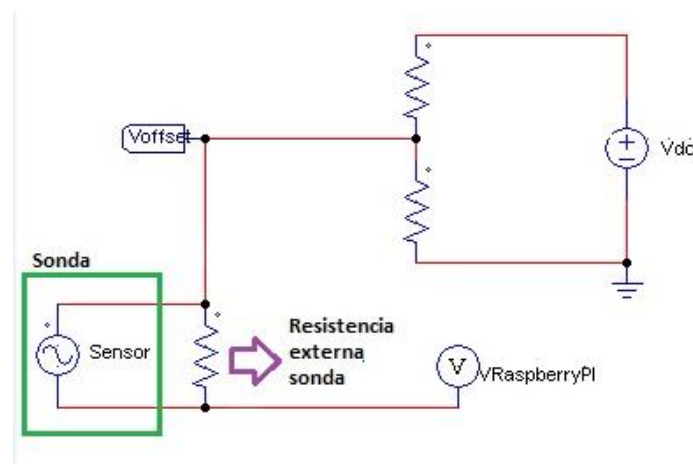


Figura 7. Circuito de sensado

6.2. CONVERTIDOR ANALÓGICO-DIGITAL

La medida de la corriente proporcionada por el sensor, como ya se ha comentado antes, es analógica. Por lo tanto, no podemos conectar directamente el cable de salida del sensor a los terminales GPIO de la Raspberry Pi. Para conectar el sensor de corriente con la Raspberry Pi, se utilizará un convertidor analógico-digital como medio de enlace de ambas.

Para seleccionar el convertidor que se vaya a utilizar, se deberán tener en cuenta dos características principalmente. Estas dos características son el tiempo de muestreo y la resolución del convertidor. Hay que asegurarse de que estas dos características del convertidor elegido cumplen con las especificaciones fijadas.

Los datos de corriente deberán subirse a la nube en valores de corriente eficaz promediados cada 10 minutos. El promedio de la corriente cada diez minutos se realiza con los datos instantáneos de corriente proporcionados por el sensor. Se ha decidido tomar valores de corriente instantáneos cada 1 ms, para que la onda de la corriente quede bien definida. Dado que la corriente de la red es de 50Hz, la onda tiene un periodo de $1/50=20$ ms, y dado que se cogerán datos cada milisegundo, se tomarán 20 muestras por cada onda de corriente. Por lo tanto, el periodo de muestreo del convertidor elegido deberá ser menor de 1 ms.

Por otro lado, se ha decidido que se quiere una precisión de corriente de 0.1 A. Sabiendo que la Raspberry Pi solo puede ser alimentada con una tensión de 3.3V, y siendo la relación de transformación entre la tensión y la corriente $1000/34$, se necesita una resolución mínima de 10 bits en el convertidor. Esta resolución necesaria queda probada con los siguientes cálculos:

$$2^{10 \text{ bits}} = 1024 \text{ valores diferentes}$$

$$\text{Resolución} = \frac{3.3V}{1024} = 3.22 \times 10^{-3} \frac{V}{\text{valor digital}}$$

Si se aplica la relación de transformación para obtener el incremento del valor de la corriente por cada valor digital que se aumenta:

$$\text{Resolución} = 3.22 \times 10^{-3} \frac{V}{\text{valor digital}} \times \frac{1000}{34} = 0.09471 \frac{A}{\text{valor digital}}$$

Una vez definidas las características que debe tener el convertidor (resolución mínima de 10 bits y un periodo de muestreo menor de 1 ms), el convertidor elegido ha sido el MCP 3008.

El convertidor analógico-digital MCP3008 tiene una resolución de 10 bits. Puede ser alimentado a una tensión continua que oscila entre 2.7V y 5 V. Según a la tensión que se alimente, el número de muestras que proporcione por segundo será diferente, dando 200000 muestras por

segundo cuando está alimentado a 5.5V, y 75000 muestras por segundo cuando se alimenta a 2.7 V. El tiempo de muestreo del convertidor cuando está alimentado a 2.7 V es el siguiente:

$$Tiempo\ de\ muestreo = \frac{1}{75000\ muestras/s} = 1.33 \times 10^{-5}s$$

Incluso el mayor tiempo de muestreo del convertidor ($1.33 \times 10^{-5}s$) es menor que el tiempo de muestreo que se va usar en el programa para leer los datos de corriente. Así pues, se puede alimentar el convertidor a la tensión que se desee, dentro del rango de 2.7-5.5V. En este caso se ha elegido una tensión de 3.3V, ya que la Raspberry Pi posee en uno de sus terminales GPIO una fuente de tensión de este valor, por lo que resulta lo más cómodo alimentarlo a esta tensión.

Otra característica importante que se debe comentar es que el convertidor analógico-digital transmite los datos a la Raspberry Pi utilizando la interfaz de datos serie SPI de la misma. Posee 8 entradas analógicas y una única salida digital, por la cual salen los datos en serie. El convertidor envía por este canal de salida digital 3 bytes en serie, los que contienen la medida de la corriente en binario. La salida del circuito de sensado, es decir, el terminal que nos da el valor de la corriente medida y acondicionada para el convertidor, se conecta a una de las 8 entradas analógicas que posee el convertidor. En este proyecto se ha conectado al canal analógico número 0 (CH0).

Aparte de las características más importantes, ya comentadas, se detallan también todo el resto de características del convertidor analógico-digital en la hoja de características de éste que se adjunta en los anexos.

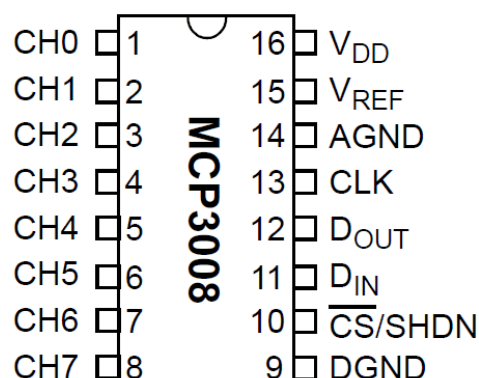


Figura 8. Pines convertidor analógico digital

6.3. CONEXIÓN DEL CONVERTIDOR ANALÓGICO-DIGITAL A LA RASPBERRY PI

A continuación, se muestra el conexionado que se debe realizar de los terminales del convertidor con los de la Raspberry Pi. El terminal de la Raspberry Pi el cual se deberá conectar el terminal del convertidor es el que está en la columna de su derecha para cada caso.

Terminales del convertidor analógico-digital	Terminales GPIO de la Raspberry Pi
V _{DD} (terminal 16)	3.3V (terminal 1)
V _{REF} (terminal 15)	3.3V (terminal 1)
AGND (terminal 14)	GROUND (terminal 9)
CLK (terminal 13)	SPIO_SCLK (terminal 23)
DOUT (terminal 12)	SPIO_MISO (terminal 21)
DIN (terminal 11)	SPIO_MOSI (terminal 19)
CS (terminal 10)	SPIO_CEO_N (terminal 24)
DGND (terminal 9)	GROUND (terminal 39)

Seguidamente, se muestra un esquema de las conexiones que deben realizarse entre la Raspberry Pi y el convertidor analógico-digital que ya se han comentado.

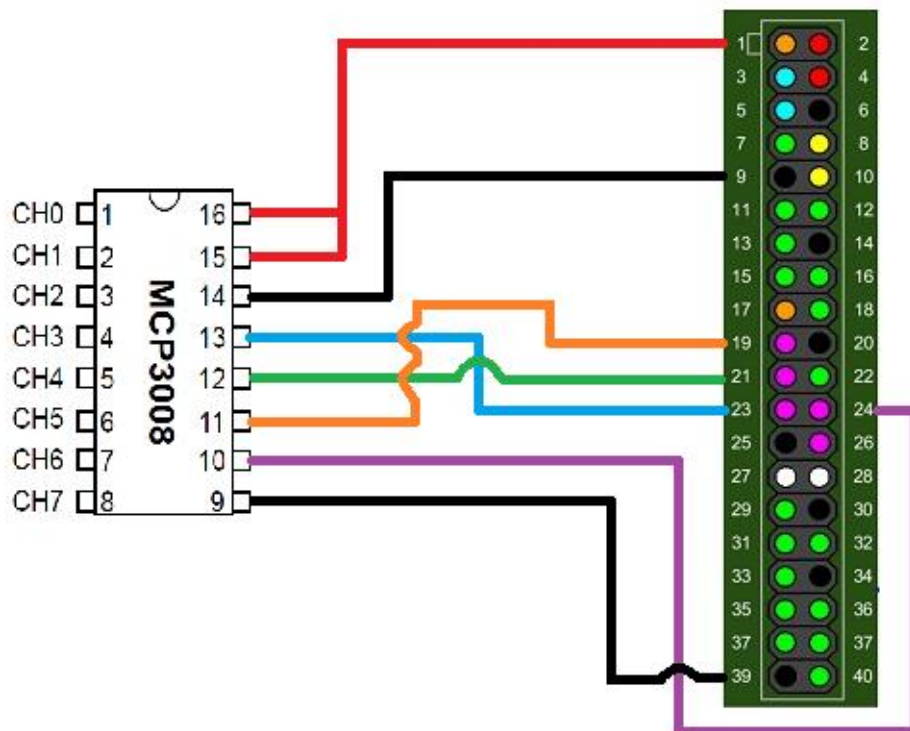


Figura 9. Conexión convertidor analógico-digital pines GPIO

7. PROGRAMACIÓN DE LA RASPBERRY PI

Una vez realizado todo el montaje, quedando un circuito total compuesto por la conexión del circuito de sensado con el convertidor analógico-digital y la conexión del convertidor analógico-digital con la Raspberry Pi, se procederá a escribir el programa.

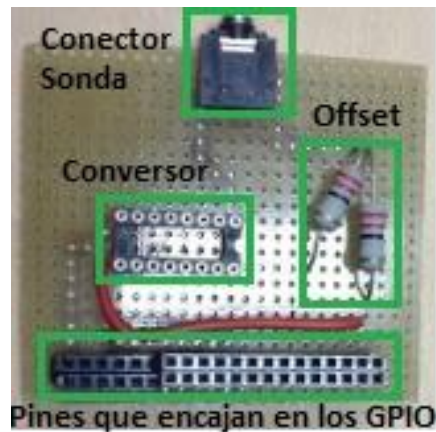


Figura 10. Circuito de acondicionamiento de señal completo

El programa deberá leer los datos de corriente instantánea que llegan a los terminales GPIO de la Raspberry Pi, realizar el cálculo del promedio del valor eficaz de la corriente en intervalos de 10 minutos, y por último deberá subir el valor de corriente eficaz calculado y promediado durante este tiempo a la hoja de cálculo creada en Google-Drive. En la figura 11 se muestra mediante un diagrama de flujos el funcionamiento del programa.

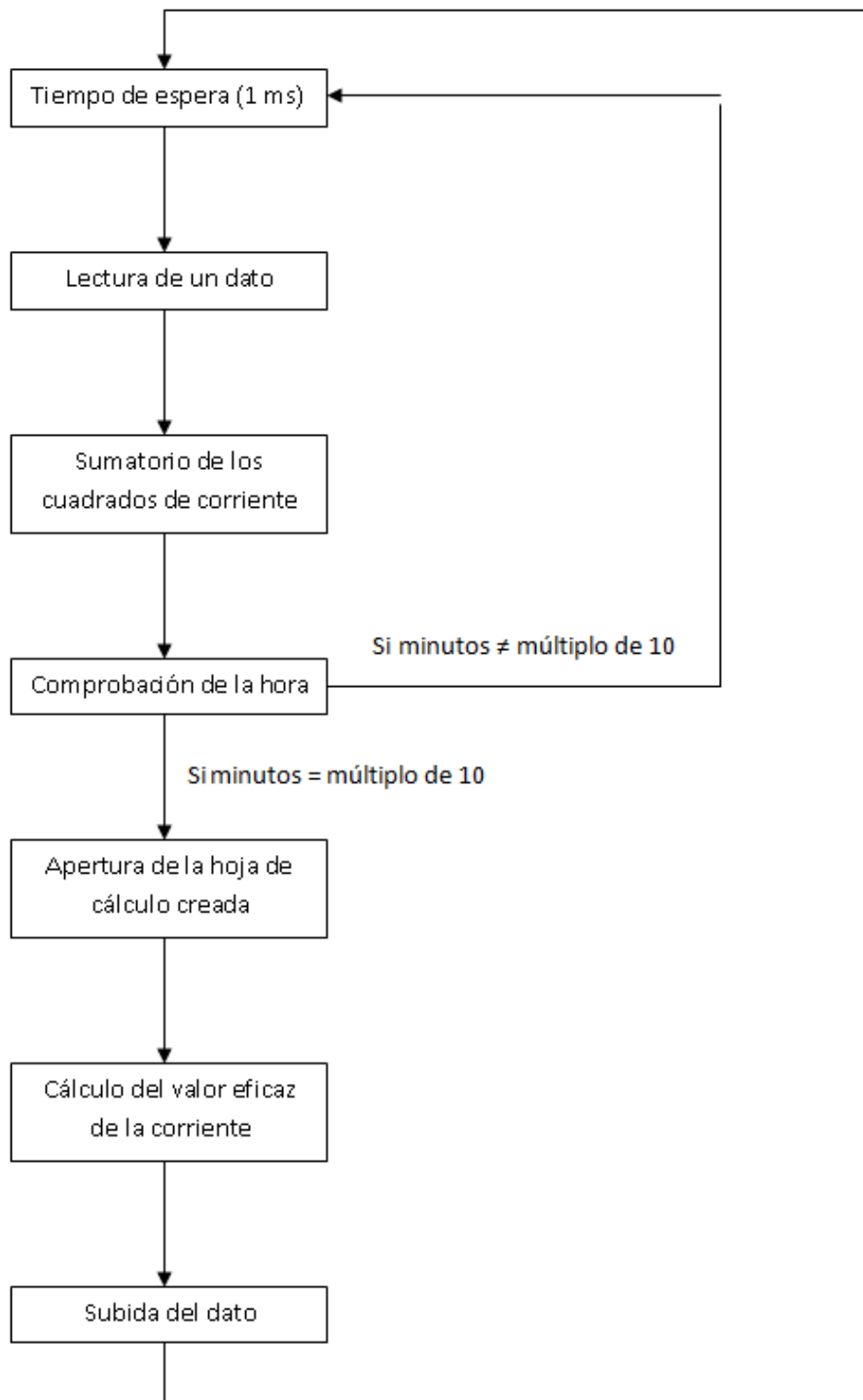


Figura 11. Diagrama de flujos

Este ciclo deberá realizarse continuamente. A continuación, se explicará el programa realizado para que el contador inteligente realice la función asignada a este trabajo.

7.1. CARGA DE LAS LIBRERÍAS NECESARIAS

El primer paso a realizar en el programa es realizar la carga de todas las librerías necesarias que se vayan a utilizar en el programa. Esto se realiza mediante el comando import, y queda de la siguiente manera:

```
import spidev
import time
import os
import math
import sys
import datetime
import json
import gspread
from oauth2client.client import SignedJwtAssertionCredentials
```

Una parte de las librerías que se deben cargar ya están instaladas en la Raspberry Pi como os (provee una amplia variedad de funciones para interactuar con el sistema), math (para realizar operaciones matemáticas), sys (provee acceso a funciones y objetos mantenidos por el intérprete), time y datetime (para tener noción del tiempo). Otras en cambio no lo están, por lo que previamente deben ser instaladas en la Raspberry Pi. Estas librerías son spidev (para poder usar los puertos SPI), gspread (para subir los datos a la hoja de cálculo de Google-Drive), y oauth2 (necesario para subir datos a la hoja de cálculo).

- Instalación de las librerías necesarias para subir datos a la hoja de cálculo de Google-Drive:

1. gspread:

```
sudo apt-get install python-pip
sudo pip install gspread
```

2. oauth2client:

```
sudo pip install --upgrade oauth2client
```

3. PyOpenSSL:

```
sudo pip install PyOpenSSL==0.13
```

Aunque esta última librería no se carga en el programa, es necesario tenerla instalada en la Raspberry Pi para poder subir datos a Google-Drive. El compararlo con 0.13 (==0.13) se realiza para instalar la versión 0.13 de PyOpenSSL, ya que con otras versiones no funciona el programa.

- Instalación de los SPI:

Se deben ejecutar en la consola los siguientes comandos:

```
sudo apt-get update
sudo apt-get upgrade
sudo reboot
```

Tras realizar las actualizaciones pertinentes y reiniciar la Raspberry Pi, instalamos el paquete *Python Dev*:

```
sudo apt-get install python-dev
```

Después de haber instalado el paquete anterior, deben habilitarse los terminales SPI. Para ello, entramos en el menú de configuración (*raspi-config*) con los comandos explicados en el apartado 5.3 de la memoria. Una vez dentro de este menú, se accede a *Advanced options* y se le da a habilitar los SPI (SPI Enable).

También se necesita el módulo *SpiDev* de Python. Para ello, se deben ejecutar los siguientes comandos en la consola:

```
mkdir python-spi
cd python-spi
wget https://raw.githubusercontent.com/doceme/py-spidev/master/setup.py
wget https://raw.githubusercontent.com/doceme/py-spidev/master/spidev_module.c
sudo python setup.py install
```

Una vez realizado esto, los terminales SPI ya están listos para utilizarse en Python. [6]

Tras instalar todas estas librerías, se podrán cargar en el script de Python todas las librerías necesarias para realizar el programa. A continuación, se realizará la apertura del Bus SPI en el programa Python escribiendo las siguientes líneas de código:

```
spi = spidev.SpiDev()
spi.open(0,0)
```

7.2. OBTENCIÓN DE CREDENCIALES

Las credenciales sirven como sistema de seguridad para acceder a la hoja de cálculo de Google-Drive. Gracias a éstas, no es necesario introducir en el programa ni la dirección de correo ni la contraseña del propietario de la hoja de cálculo. Para obtener las credenciales que habrá que introducir en el programa, primero deberán realizarse una serie de pasos en la página web *Google Developers Console*.

Lo primero que debe hacerse en esta página es habilitar la API "Drive API". Después de esto, hay que entrar en *credentials* (credenciales) y crear un nuevo cliente ID (tipo de aplicación: *Service account*). Al realizarlo, se descarga automáticamente un archivo JSON con una serie de datos (las credenciales que se utilizarán en la programación). Para este trabajo el archivo descargado y utilizado para la programación es el siguiente:

```
{
  "private_key_id": "31caf84608220005916d8b0d04e3133ac3f092a8",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvwIxA..... u003d\n-----END PRIVATE KEY-----\n",
  "client_email": "370794536162-jd3i3sjbionjoenkec2q3lqg1g2j6as4@developer.gserviceaccount.com",
  "client_id": "370794536162-jd3i3sjbionjoenkec2q3lqg1g2j6as4.apps.googleusercontent.com",
  "type": "service_account"
}
```

Por último, se debe compartir la hoja de cálculo creada a la que se pretenden subir los datos con la cuenta de correo que ha sido proporcionada en el archivo descargado `json_key['client_email']`.

Todos estos pasos están más detallados en la página Using Oauth2 for Authorization.[7]

Obtenidas las credenciales tras realizar los pasos anteriores, se pueden introducir éstas en el programa que se está realizando en Python.

```
json_key = json.load(open('monitordeconsumo-31caf8460822.json'))
scope = ['https://spreadsheets.google.com/feeds']

credentials = SignedJwtAssertionCredentials(json_key['client_email'],
json_key['private_key'], scope)
```

7.3. DEFINICIÓN DE LAS FUNCIONES Y ASIGNACIÓN DE VALORES INICIALES A VARIABLES

Al comienzo del programa será necesario dar a las variables que se van a utilizar en el mismo ciertos valores iniciales. Algunas de estas variables cambiarán su valor durante la ejecución del programa, mientras que otras se mantendrán constantes. Las variables que se fijan al comienzo del programa son las siguientes:

```
current_channel = 0
delay = 0.001
Sumadecorriente = 0
contador = 0
```

Para que el programa esté midiendo la corriente consumida en la vivienda cada 1 ms, se definen todas las funciones (medida de la corriente, cálculo del valor eficaz...) al principio del programa, y después se ejecutan estas funciones dentro de un bucle infinito. A continuación, se muestran las funciones creadas para este programa, y el resultado que dan sobre una señal tomada como ejemplo. La señal seleccionada es la máxima que puede procesar la Raspberry Pi, con 34.31 A de valor eficaz y 50 Hz.

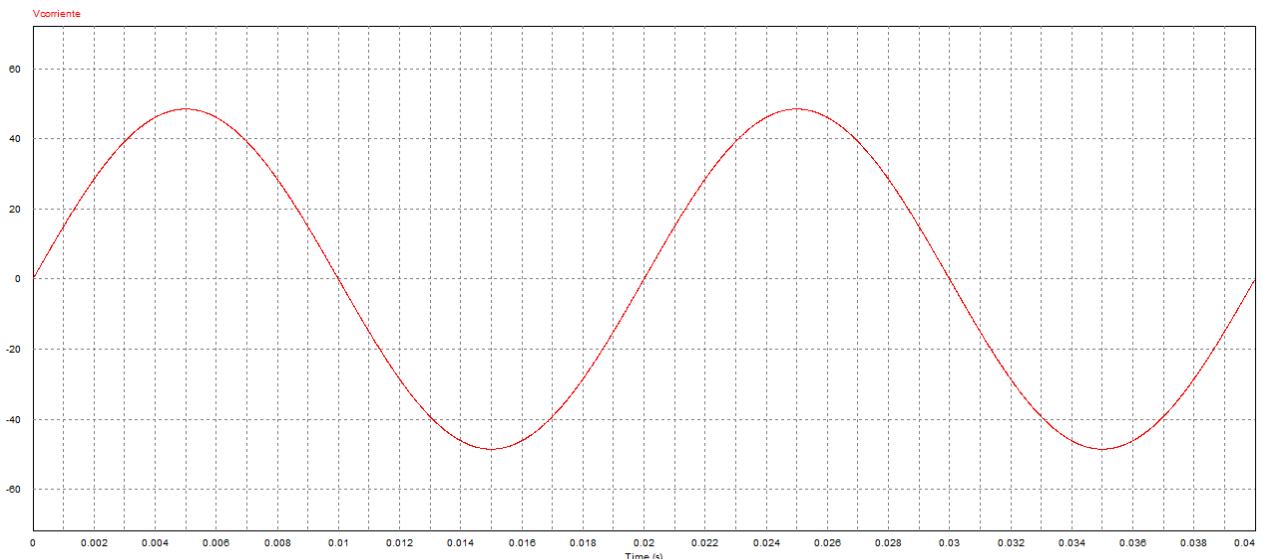


Figura 12. Corriente medida por el sensor

El sensor mide esta señal y a tras acondicionar la señal con el circuito de sensado, a la entrada del convertidor se recibe una señal de 3.3 V de pico y 50 Hz, como se muestra en la figura 13.

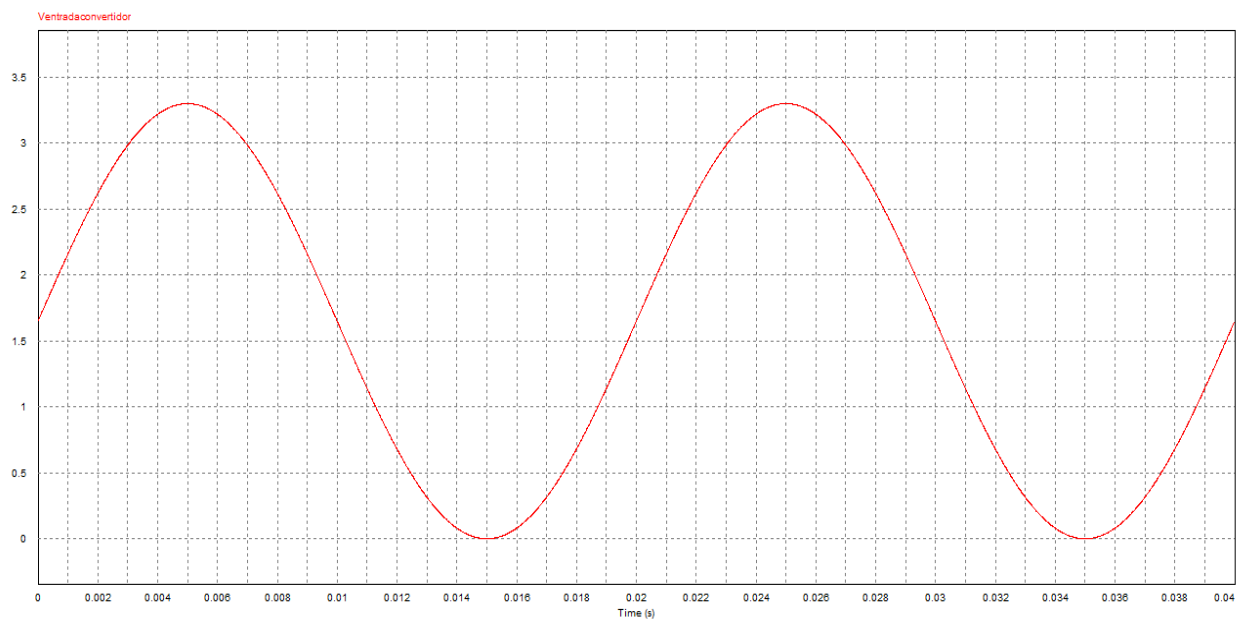


Figura 13. Onda de tensión generada por el sensor.

7.3.1. LECTURA DEL DATO PROPORCIONADO POR EL CONVERTIDOR ANALÓGICO-DIGITAL

Esta función realiza la lectura de los datos de corriente que llegan a los terminales GPIO desde el convertidor analógico-digital. Los datos proporcionados por el convertidor analógico-digital se guardan en una variable data que oscila entre los valores 0-1023. [8]

```
def ReadChannel(channel):
    adc = spi.xfer2([1, (8+channel)<<4, 0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data
```

7.3.2. OBTENCIÓN DEL VALOR DE TENSIÓN QUE PROPORCIONA LA SONDA

Mediante esta función, el programa realiza la conversión del valor guardado en data a un valor de tensión. El valor de tensión que se obtiene es el valor que proporciona el circuito de sensado al convertidor analógico-digital, el cual oscilará entre 0 y 3.3V.

```
def ConvertVolts(data, places):
    volts = (data*3.3)/float(1023)
    volts = round(volts, places)
    return volts
```

En realidad, este valor no es necesario conocerlo para la finalidad del programa, pero ha sido muy valioso conocerlo, para ver si el convertidor analógico-digital funciona como es debido.

En la figura 14 se puede apreciar los valores de tensión que proporciona esta función después de haber leído cada 1 ms los datos del ejemplo que llegan a los terminales GPIO:

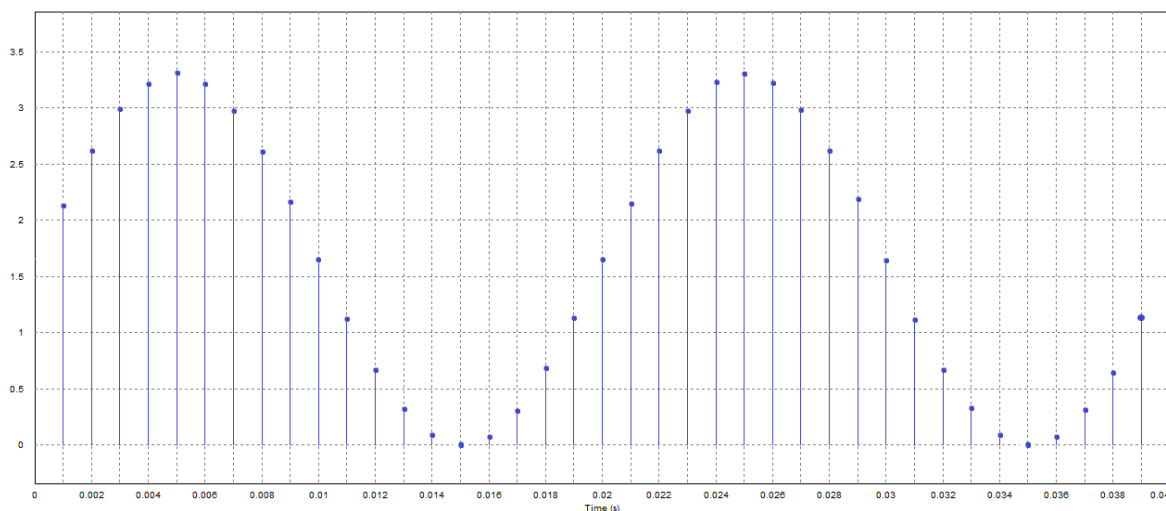


Figura 14. Onda de tensión muestreada

Como se puede apreciar, cada vez que se ejecute esta función, se obtendrá un valor de tensión comprendido entre 0 y 3.3 V.

7.3.3. MEDIDA DE LA CORRIENTE INSTANTÁNEA

Esta función obtiene el valor de corriente instantánea que se está consumiendo a partir de la variable data. Es muy parecida a la función anterior, ya que realiza la conversión del valor de la variable data a tensión, pero al añadirle la relación de corriente/tensión de la sonda, se obtiene la corriente que se está consumiendo.

```
def ConvertCurrent(data,places):
    current = ((data*3.3)/float(1023))-1.65)*(1000/34)
    current = round(current,places)
    return current
```

Seguidamente, se muestra un gráfico en el que se puede apreciar la conversión que realiza esta función, la que cada dato proporcionado por el convertidor lo pasa a corriente. Cuando se ejecuta la función varias veces, se va obteniendo dicho gráfico.

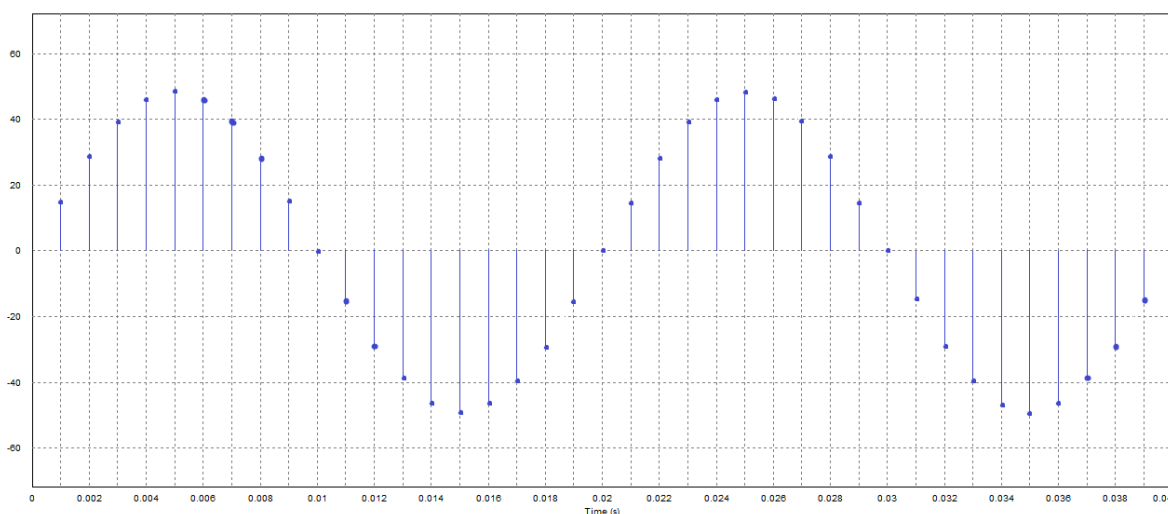


Figura 15. onda de corriente muestreada

7.3.4. CÁLCULO DEL CUADRADO DE LA CORRIENTE INSTANTÁNEA

En esta función, como el propio nombre del apartado indica, se calcula el cuadrado de la corriente instantánea. Este dato es necesario para realizar el cálculo del valor eficaz de la corriente, ya que como se podrá apreciar más adelante, el valor eficaz de la corriente depende del cuadrado de la corriente instantánea.

```
def CorrienteCuadrado(current):  
    corrienteber2 = current**2  
    return corrienteber2
```

El resultado del cálculo del cuadrado de cada dato de corriente que se va logrando cada milisegundo puede apreciarse en la figura 16:

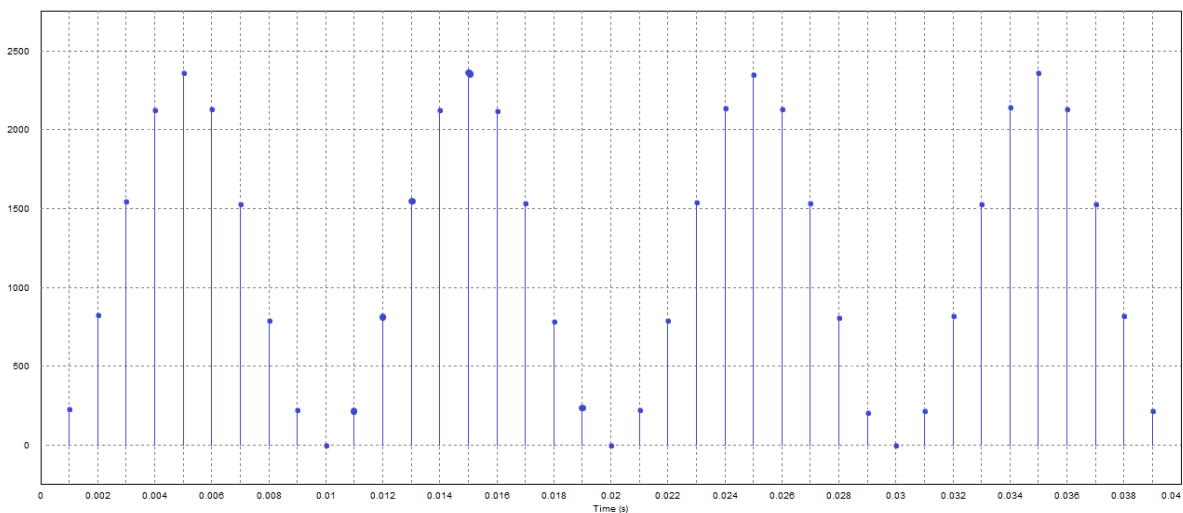


Figura 16. Cuadrados de corriente

7.3.5. CÁLCULO DEL VALOR EFICAZ DE LA CORRIENTE

Antes de mostrar la función que calcula el valor eficaz de la corriente en el programa, es necesario explicar cómo se va a realizar el cálculo del valor eficaz. El valor eficaz de la corriente queda definido de la siguiente manera:

$$I_{valor\ eficaz} = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} i(t)^2 dt} \quad \text{[Ecuación 1]}$$

Donde: $i(t)$ = valor instantáneo de la corriente
 T = periodo de la onda (20 ms)

La parte de la definición en la que se realiza el cuadrado de la corriente instantánea se lleva a cabo en la última función explicada. Para realizar la integral se utiliza el método de los trapecios (se suman todos los valores del cuadrado de corriente que se van obteniendo cada milisegundo y luego

se dividen por el número de muestras cogidas cada milisegundo). Esta fórmula de cálculo de la corriente eficaz en el programa queda definida de la siguiente manera:

```
def ValorEficaz(Sumadecorriente, contador, places) :
    ValorEficaz = math.sqrt(Sumadecorriente/contador)
    ValorEficaz = round(ValorEficaz, places)
    return ValorEficaz
```

Al dividir la suma de cuadrados de corriente con el número de sumas realizadas, ya no es necesario dividirlo por el periodo de la onda.

7.3.6. PASO DE LA HORA DEL DÍA A SEGUNDOS

Esta función pasa la hora que se indica a una acumulación de segundos. Por ejemplo las 00:00 y 0 segundos serán 0 segundos, y las 02:00 y 20 segundos serán 7220 segundos. Al fin y al cabo, la duración del día es como un contador de segundos que va desde 0 segundos hasta 86400 segundos. Cuando llegue a ese valor, el contador vuelve otra vez a cero (ha pasado un día). Las líneas de código del programa que realizan esta operación son las siguientes:

```
def getSec(s) :
    l= s.split(':')
    return int(l[0])*3600 + int(l[1]) * 60 + int(l[2])
```

En “s” se coloca la variable en la que se guarda la hora que se quiere pasar a segundos. Para eso habrá que calcular primero la hora que es en el momento de hacer la conversión y guardarlo en dicha variable. [9]

7.4. BUCLE INFINITO

Una vez que se han definido todas las funciones y las variables, el programa entra en este bucle, el cual es ejecutado continuamente. Como ya se ha comentado antes, este bucle deberá ir midiendo la corriente instantánea cada 1 ms, y cada 10 minutos deberá realizar el cálculo del promedio del valor eficaz de la corriente que se ha consumido durante esos 10 últimos minutos. Para ello, este bucle realiza los siguientes pasos:

1. Se ejecutan las funciones de lectura del dato proporcionado por el convertidor analógico-digital, obtención del valor de tensión que proporciona la sonda, medida de la corriente instantánea y cálculo del cuadrado de la corriente instantánea definidas anteriormente, y los valores obtenidos son guardados en las variables `current_level`, `current_volts`, `current` y `corrienteber2` respectivamente, y posteriormente mostrados en la consola.
2. El valor del cuadrado de la corriente que se obtiene en cada ejecución del bucle infinito (`corrienteber2`) se va sumando a los valores obtenidos de esta variable en anteriores ejecuciones del bucle, y se guardan en la variable `Sumadecorriente`. A continuación, se incrementa en uno el valor de la variable `contador` (variable que nos dirá cuántas muestras de corriente hemos cogido para después poder hacer el cálculo del valor eficaz).

3. Seguidamente, el programa obtiene la hora de ese instante con el formato horas : minutos : segundos, y después se pasa a segundos con la última función definida en el programa.
4. Se realiza la división de la hora en formato de acumulación de segundos por 600 (10 minutos). Si el resto de la división es cero querrá decir que la hora es en punto, y diez, y veinte, y media, menos veinte o menos 10, lo que indicará que han pasado 10 minutos. Si no se cumple esta condición, el programa esperará un 1 ms y comenzará otra vez el ciclo descrito. En el caso que se cumpla la condición, es decir han pasado 10 minutos, el programa en primer lugar accederá a la hoja de cálculo que se haya creado mediante las credenciales y calculará el valor de la corriente eficaz. Este valor también será mostrado en la consola. Inmediatamente, se vuelve a obtener la hora pero con el formato día/mes/año horas : minutos : segundos. Una vez obtenidos el valor eficaz de la corriente y la hora, estos dos se suben a Google-Drive. Lo último que debe realizar el programa antes de empezar a obtener datos cada 1 ms es poner a cero el valor de las variables Valor_eficaz_corriente, Sumadecorriente y contador, ya que si no, el siguiente dato de valor eficaz de corriente que calculase el programa estaría mal, y esperar 1 ms.

A continuación, se muestra el bucle infinito que realiza todas las acciones descritas en los cuatro pasos anteriores:

```
while True:

    current_level = ReadChannel(current_channel)
    current_volts = ConvertVolts(current_level,3)
    current = ConvertCurrent(current_level,3)
    corrienteber2 = CorrienteCuadrado(current)
    print("Corriente: {} ({}V) {}A
    {}A**A".format(current_level,current_volts,current,corrienteber2))

    Sumadecorriente = Sumadecorriente + corrienteber2
    contador = contador + 1

    muestrahora = time.strftime("%H:%M:%S",
time.localtime(time.time()))
    hora = getSec(muestrahora)
    decimal = hora%600
    if decimal == 0:

        gc = gspread.authorize(credentials)

        wks = gc.open("consumoRPI").sheet1

        Valor_eficaz_corriente =
ValorEficaz(Sumadecorriente,contador,3)
        print("Valor eficaz de corriente:
        {}A".format(Valor_eficaz_corriente))
        localtime = time.strftime("%d/%m/%y %H:%M:%S",
time.localtime(time.time()))
        datos = [localtime,Valor_eficaz_corriente]
        wks.append_row(datos)
```

```
Valor_eficaz_corriente = 0
Sumadecorriente = 0
contador = 0

time.sleep(delay)
```

En este apartado, se ha explicado el funcionamiento del programa mostrando el programa a pedazos. En los anexos se muestra el programa completo.

8. ENSAYO FINAL

Tras terminar con el montaje completo del contador y haber programado la Raspberry Pi, se ha realizado una prueba para comprobar su funcionamiento. Dicha prueba ha constado en conectar el contador desarrollado a una de las salidas del inversor de la microrred de la Universidad Pública de Navarra, con el fin de medir la corriente que circula por la misma. La salida del inversor a la que ha sido conectada es la número 2, y los datos obtenidos por el contador han sido comparados con los que proporciona la microrred.



Figura 17. Producto final

El ensayo se realizó el día 18 de junio de 2015, en el que se estuvo tomando datos durante 3 horas. En el anexo III se muestra una tabla con la hora a la que se realizaron las medidas, las medidas que realizó la Raspberry Pi y las medidas proporcionadas por la microrred. En el anexo III, también se realiza un cálculo del error de precisión que posee la Raspberry Pi (se supone que las medidas de la microrred son exactos).

Como se puede apreciar en la tabla del anexo III, el error de precisión de la Raspberry Pi es de 0.027 A. es decir, cada medida que proporciona la Raspberry Pi es 0.027 A mayor que el dato real.

Con los datos obtenidos en la tabla del anexo III, se realiza un gráfico. En éste se muestran las corrientes medidas tanto por la Raspberry Pi, como por la microrred.

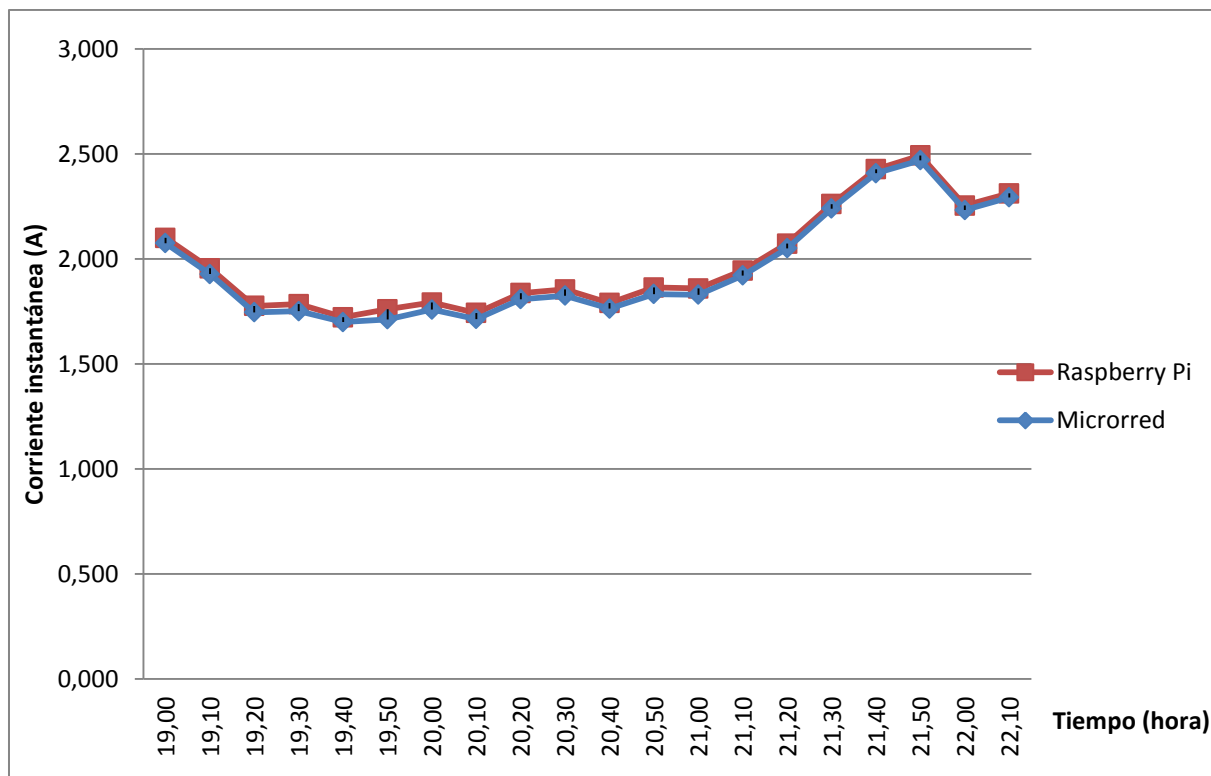
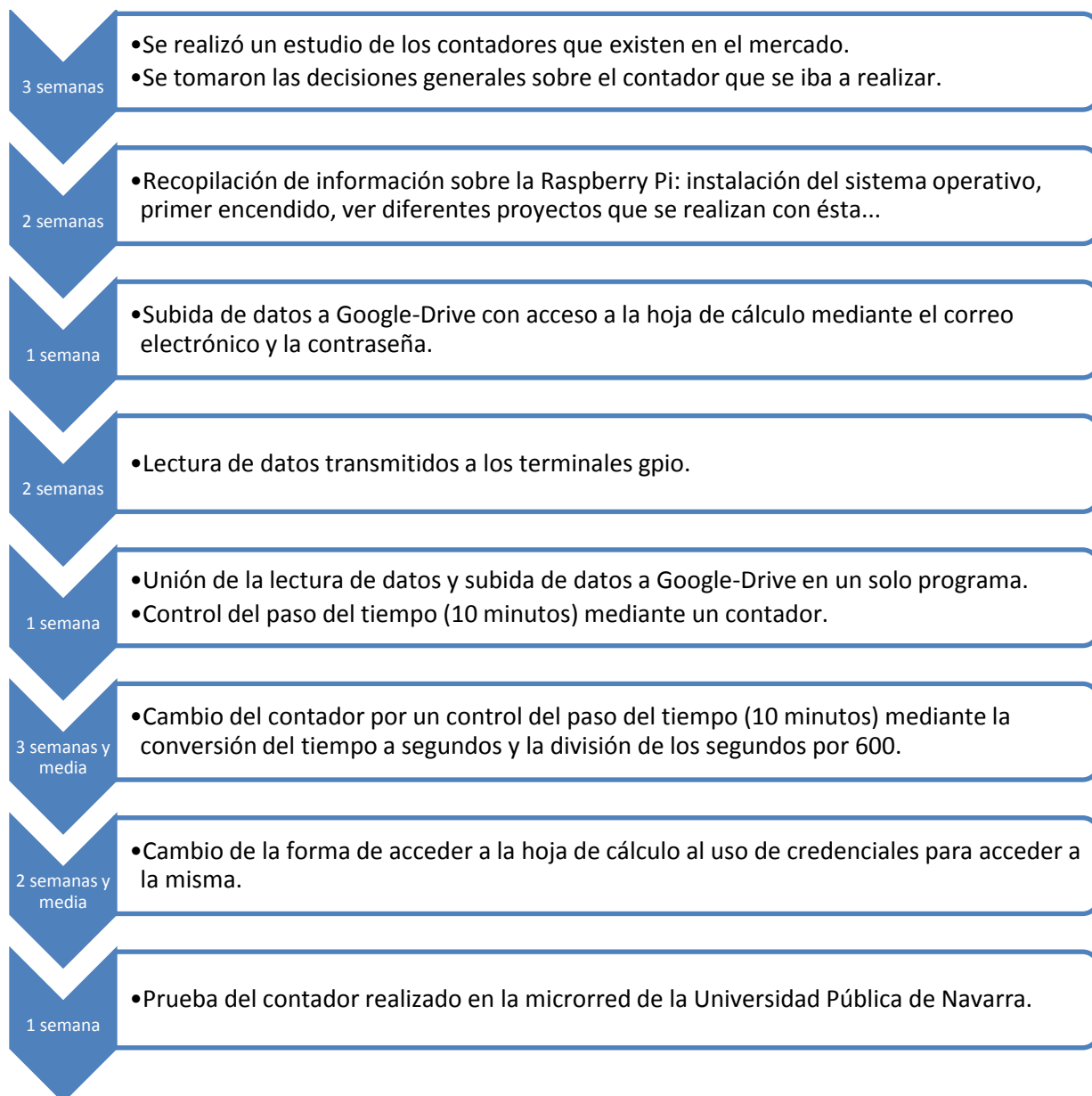


Figura 18. Representación de los datos del ensayo

9. LÍNEA TEMPORAL

En este apartado se transmite información acerca del tiempo que se ha destinado en la realización de diferentes tareas para el desarrollo del contador.



10. PRESUPUESTO

En este apartado se muestran todos los elementos necesarios para la realización del monitor de consumo, así como su coste. El apartado está compuesto por dos tablas diferentes, en la primera de ellas, la Tabla 1, muestra los elementos necesarios para cubrir las necesidades de este trabajo.

Artículo	Cantidad	Precio unitario (€)	Precio total (€)
Raspberry Pi 2 Model B	1	41.2	41.2
Cargador (5V, 2A)	1	8.99	8.99
Tarjeta microSD	1	2.9	2.9
Adaptador WiFi	1	20.37	20.37
TOTAL			73.46

Tabla 1

En la Tabla 2, se muestran la totalidad de los elementos necesarios para el montaje completo del contador inteligente que se ha desarrollado en el conjunto de los tres proyectos, así como el coste total del mismo.

Artículo	Precio unitario (€)
Circuito de sensado	4.2
Caja Raspberry Pi	11.60
Convertidor analógico-digital (Mcp-3008)	2.20
Hardware de la Raspberry Pi	73.46
TOTAL	91.46

Tabla 2. Presupuesto total

11. PROBLEMAS ENCONTRADOS

El problema más importante que se ha encontrado en la realización del programa es que la forma de acceder a las hojas de cálculo de Google-Drive ha cambiado durante el tiempo en el que se ha realizado este trabajo. Google ha deshabilitado la forma de autenticar mediante el correo electrónico y contraseña del usuario. A partir del 27 mayo esta forma de autenticar el acceso a la hoja de cálculo creada para subir el dato dejó de funcionar. Esto ha obligado a cambiar toda la parte del programa relacionada con la subida de los datos a la hoja de cálculo de Google-Drive.

Para solucionar dicho problema, se ha utilizado la estructura OAuth2 que permite el acceso a la hoja de cálculo de Google-Drive mediante el uso de credenciales. Es una forma más segura de acceder, ya que no es necesario escribir en el programa ni el correo electrónico ni la contraseña del usuario, las que de la forma anterior podían quedar fácilmente a la vista de terceras personas.

Otro inconveniente que ha ocurrido en la realización del trabajo ha sido que la Raspberry Pi no lee los datos exactamente cada un milisegundo, sino que le cuesta algo más. Por lo que el cálculo del valor eficaz de la corriente mediante el método de los trapecios se ha modificado un poco. Tras la suma de todos los cuadrados de la corriente, en lugar de multiplicarlo por el tiempo de muestreo (1 ms) y después dividirlo por el periodo de la onda (20 ms), se divide directamente por el número de muestras que se han tomado. De esta forma, aunque no muestree exactamente cada milisegundo, se obtiene la información suficiente sobre la onda y el cálculo se soluciona de una forma sencilla.

Por último, comentar también que al ejecutar el programa, hay veces en que éste se para debido al siguiente error que se muestra en la pantalla:

```
raise HTTPError (response) gspread.httpsession.HTTPError
```

Este error ha aparecido la última semana de realización del proyecto, por lo que, aunque se ha intentado solucionarlo, no ha habido mucho tiempo para arreglarlo. Se cree que puede ser debido a que el número de veces que se puede acceder a la hoja de cálculo mediante el método utilizado es limitado, y que cuando se excede de ese número de accesiones el programa se para, mostrando dicho error en la consola.

12. COMPARATIVA DE RASPBERRY PI CON ARDUINO

A continuación, se realiza una comparativa el procesamiento de la señal mediante arduino y el procesamiento mediante Raspberry Pi. Para ello se comparan diferentes características que aportan el empleo de cada alternativa:

- El empleo de Arduino emplea dos módulos diferentes, uno para el sensado y otro para la subida del dato a la hoja de cálculo almacenada en Google-Drive. Estos módulos se comunican mediante radiofrecuencia. Por el contrario, la Raspberry Pi únicamente necesita un módulo, el cual realiza ambas operaciones. Esto conlleva a que el empleo de Arduino requiera mayor espacio, aunque éste esté distribuido en diferentes zonas.
- En cuanto a la conexión a internet, se realiza por medio de WiFi por parte de la Raspberry Pi, y mediante conexión de cable Ethernet por parte de Arduino. En el caso de la Raspberry Pi, la distancia a la que se podrá colocar el contador del router dependerá del alcance del WiFi. Con Arduino, el módulo que habrá que conectar a internet se podrá colocar cerca del router, por lo que el cable Ethernet necesario será corta longitud. Pero la distancia a la que se podrá colocar un módulo del otro, y por lo tanto, la distancia del contador con el router quedará limitada por la capacidad de la radiofrecuencia.
- A pesar de que el Arduino utiliza dos módulos que se deben alimentar por separado, el consumo de la Raspberry Pi es mayor que el de los dos módulos de Arduino juntos. Esto es debido a que el adaptador WiFi utilizado en la Raspberry Pi para conectarnos a internet obliga a la utilización de un cargador de un mínimo de 1 A (en el trabajo se ha usado un cargador de 2 A) para su correcto funcionamiento. Por otro lado, el módulo de sensado de Arduino consume 40 mA y se emplea un cargador con capacidad de proporcionar 1.3 A; el módulo de envío de datos consume una media de 220 mA, y se empleará un cargador similar al otro módulo.
- Debido a que el convertidor analógico-digital utilizado con Arduino procesa peor la señal comparado con el convertidor analógico-digital empleado con la Raspberry Pi, en la Raspberry Pi se obtiene una mejor precisión en los datos. También se ha contemplado la opción de que la diferencia de error es debida a la forma de calcular el valor eficaz de la corriente. Ya que con la Raspberry Pi se utiliza la definición de la corriente eficaz (ecuación 1), y con Arduino el cálculo dividiendo el máximo por raíz de dos.
- A la hora de subir datos, Arduino tiene un error acumulativo de 42 ms, lo que se traduce en 6 segundos al día. Por el contrario, la Raspberry Pi cumple el objetivo de subir los datos cada 10 minutos, pero en ocasiones se produce un error que para el sistema.

- Por último, desde un punto de vista económico, el empleo de arduino en un principio resulta más económico que el de Raspberry Pi, aunque la diferencia de precio no es muy alta, ya que sus precios son 69.18 € y 73.46 € respectivamente. Pero si tomamos en cuenta el coste de los circuitos de sensado el precio de Arduino es mayor que el de Raspberry Pi. [10]

Para concluir, se emplea La tabla 3 para poder analizar de forma rápida las ventajas y desventajas de cada sistema de procesado de datos:

	Arduino	Raspberry Pi
Tamaño	Mayor	Menor
Tipo conexión	Ethernet	WiFi
Alimentación	<500 mA	>1 A
Precisión	Menor	Mayor
Tiempo de subida	No exacto	Se interrumpe
Precio	109,18 €	91,46 €

Tabla 3. Comparativa

13. CONCLUSIONES Y LÍNEAS FUTURAS

13.1. CONCLUSIONES

Se puede decir que en principio se ha logrado el objetivo propuesto al principio del proyecto de diseñar e implementar un contador inteligente, siendo su funcionamiento muy similar al de los contadores que podemos encontrar en el mercado. En cuanto al precio, los contadores del mercado similares al desarrollado en este proyecto (los que transmiten los datos vía web) poseen un precio de entorno a los 120€ (ejemplo: mirubee mirubox V.1.0 cuesta 119.5 € i.v.a. incluido). El contador desarrollado en este proyecto tiene un coste de 91.46 € sin incluir el i.v.a. Por lo tanto, el producto desarrollado no despuntaría en diferenciación, ya que no ofrece mejoras sustanciales respecto a los existentes en el mercado, ni sería muy competitivo en precio, ya que al aplicarle el i.v.a., el precio del contador, aunque sería un poco más barato que los existentes en el mercado, sería similar a estos.

Por otro lado, es muy importante analizar el consumo que tendrá el contador diseñado, ya que no tendría ningún sentido construir un contador inteligente con la finalidad de mejorar la eficiencia energética y disminuir el consumo, que consumiese mucha energía eléctrica. Desde este punto de vista, la utilización de una Raspberry Pi para la implementación del contador es un claro acierto, porque una Raspberry Pi conectada a internet por medio de WiFi consume una media de 2.5 W, cuando un ordenador en standby consume entre 2.2 y 3.6 W.

En cuanto al procesamiento de los datos, se ha conseguido que los datos se suban a Google-Drive en los instantes de tiempo requeridos. Ya que, a pesar de que en ocasiones cueste incluso hasta 5 ó 6 segundos subir el dato a Google-Drive, el siguiente dato lo sube en el instante preciso, sin que se den retrasos acumulativos. Por otro lado, la precisión de la medida de la corriente se considera que es buena, tal y como se aprecia en la figura 13, teniendo un error de precisión de 0.027 A en las medidas. Sin embargo, el no haber podido corregir el error comentado en el apartado

problemas encontrados impide el uso continuado del contador, cuando en principio el contador está pensado para ser utilizado de forma ininterrumpida.

13.2. LÍNEAS FUTURAS

Como aspectos de mejora que se pueden seguir realizando, lo primero sería corregir el error que ocurre de vez en cuando, el cual impide que se siga ejecutando el programa. Para ello, una posible solución podría ser desvincularse totalmente de la subida de datos a una hoja de cálculo de Google-Drive, y en lugar de ello, utilizar "OwnCloud". Esto es una nube de datos como Dropbox disponible para Raspberry Pi, en la que el límite de almacenamiento lo impone el usuario.

Por otro lado, en caso de que se quisiera llevar el producto al mercado, antes habría que realizar mejoras en el mismo. Por el lado del precio, sería difícil abaratar costes del producto, a no ser que la producción en grandes cantidades del mismo facilitara una reducción de los costes. Por lo que, el producto debería guiarse por la senda de la diferenciación sobre el resto de productos. Una mejora con la que se podría lograr esto, podría ser la comunicación vía whatsapp entre la Raspberry Pi y el usuario, en la que el usuario pudiera mandar órdenes a la Raspberry Pi desde cualquier lugar. Esto podría ser una mejora importante a tener en cuenta debido al auge que tiene dicha aplicación de móvil.

14. REFERENCIAS

- [1] NAHIA BARRIOLA HERNANDORENA, *“Monitorización del consumo eléctrico de un hogar: visualización vía web”*. Universidad Pública de Navarra, 2015.
- [2] <http://www.raspberrypi-shop.es/>
- [3] <https://www.youtube.com/watch?v=bTnLFU01kd4>, *“Iniciando tu Raspberry Pi Type B+ por primera vez”*.
- [4] <http://electronilab.co/tutoriales/tutorial-3-de-raspberry-pi-configuracion-de-la-red-y-acceso-a-internet/>
- [5] JOSEBA REVUELTA IRISARRI, *“Monitorización del consumo eléctrico de un hogar: sensado y acondicionamiento de la corriente”*. Universidad Pública de Navarra, 2015.
- [6] http://tightdev.net/SpiDev_Doc.pdf
- [7] <http://gsread.readthedocs.org/en/latest/oauth2.html>
- [8] <http://www.raspberrypi-spy.co.uk/2013/10/analogue-sensors-on-the-raspberry-pi-using-an-mcp3008/>
- [9] <http://stackoverflow.com/questions/6402812/how-to-convert-an-hmmss-time-string-to-seconds-in-python>
- [10] VÍCTOR ERICE CARBONERO, *“Monitorización del consumo eléctrico de un hogar: procesado de datos mediante Arduino”*. Universidad Pública de Navarra, 2015.

ANEXOS

ANEXO I

Programa realizado en Python.

```
import spidev
import time
import os
import math
import sys
import datetime
import json
import gspread
from oauth2client.client import SignedJwtAssertionCredentials

spi = spidev.SpiDev()
spi.open(0,0)

json_key = json.load(open('monitordeconsumo-31caf8460822.json'))
scope = ['https://spreadsheets.google.com/feeds']

credentials = SignedJwtAssertionCredentials(json_key['client_email'],
json_key['private_key'], scope)

def ReadChannel(channel):
    adc = spi.xfer2([1, (8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data

def ConvertVolts(data,places):
    volts = (data*3.3)/float(1023)
    volts = round(volts,places)
    return volts

def ConvertCurrent(data,places):
    current = (((data*3.3)/float(1023))-1.65)*(1000/34)
    current = round(current,places)
    return current

def CorrienteCuadrado(current):
    corrienteber2 = current**2
    return corrienteber2

def ValorEficaz(Sumadecorriente,contador,places):
    ValorEficaz = math.sqrt(Sumadecorriente/contador)
    ValorEficaz = round(ValorEficaz,places)
    return ValorEficaz

def getSec(s):
    l= s.split(':')
```

```

return int(l[0])*3600 + int(l[1]) * 60 + int(l[2])

print 'Press Ctrl-C quit.'

current_channel = 0
delay = 0.001

Sumadecorriente = 0
contador = 0

while True:

    current_level = ReadChannel(current_channel)
    current_volts = ConvertVolts(current_level,3)
    current = ConvertCurrent(current_level,3)
    corrienteber2 = CorrienteCuadrado(current)
    print("Corriente: {} ({}V) {}A
    {}A**A".format(current_level,current_volts,current,corrienteber2))

    Sumadecorriente = Sumadecorriente + corrienteber2
    contador = contador + 1

    muestrahora = time.strftime("%H:%M:%S",
time.localtime(time.time()))
    hora = getSec(muestrahora)
    decimal = hora%600
    if decimal == 0:

        gc = gspread.authorize(credentials)

        wks = gc.open("consumoRPI").sheet1

        Valor_eficaz_corriente =
ValorEficaz(Sumadecorriente,contador,3)
        print("Valor eficaz de corriente:
        {}A".format(Valor_eficaz_corriente))
        localtime = time.strftime("%d/%m/%y %H:%M:%S",
time.localtime(time.time()))
        datos = [localtime,Valor_eficaz_corriente]
        wks.append_row(datos)

        Valor_eficaz_corriente = 0
        Sumadecorriente = 0
        contador = 0

    time.sleep(delay)

```


ANEXO II

Datasheet del convertidor analógico digital mcp-3008

<http://docs-europe.electrocomponents.com/webdocs/1380/0900766b81380acd.pdf>

ANEXO III

Datos medidos en el ensayo

HORA	MEDIDA CON RASPBERRY PI (A)	MEDIDA DE LA MICRORRED (A)	ERROR DE PRECISIÓN DE LA RASPBERRY PI(A)
19:00	2,100	2,075	0,025
19:10	1,955	1,928	0,027
19:20	1,776	1,746	0,030
19:30	1,785	1,751	0,034
19:40	1,722	1,699	0,023
19:50	1,760	1,712	0,048
20:00	1,792	1,759	0,033
20:10	1,743	1,714	0,029
20:20	1,837	1,808	0,029
20:30	1,855	1,825	0,030
20:40	1,790	1,763	0,027
20:50	1,864	1,833	0,031
21:00	1,859	1,830	0,029
21:10	1,945	1,921	0,024
21:20	2,073	2,051	0,022
21:30	2,261	2,240	0,021
21:40	2,427	2,408	0,019
21:50	2,493	2,470	0,023
22:00	2,254	2,233	0,021
22:10	2,312	2,293	0,019
VALOR MEDIO DEL ERROR DE PRECISIÓN			0,027