



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

**Sistema de vigilancia de bajo coste energético con Raspberry Pi.**

Proyecto Final de Carrera  
ITIS

Autor: Alberto Toledo Escrihuela  
Director: Antonio Martí Campoy  
24 de Septiembre de 2015

## Índice de contenido

1.- Introducción.....	6
2.- Objetivo.....	8
3.- Materiales.....	12
3.1.1.- Raspberry pi.....	13
3.1.2.- Sensor movimiento.....	16
3.1.3.- Relés, Resistencias, Condensadores, Placas Prototipo, etc.....	19
3.1.4.- Cámara (Webcam).....	28
3.1.5.- Sistema 3G, conexión Internet.....	29
3.1.6.- Batería.....	31
3.1.7.- Bombilla.....	33
4.- Software.....	36
4.1.- Sistema operativo.....	37
4.2.- Instalación Periféricos.....	38
4.3.- Configuración para el servidor “no-ip”.....	39
4.4.- Programa para la obtención de imágenes “Motion”.....	40
4.5.- Lenguaje de programación “Python”.....	41
4.6.- Programa de comunicación usuario-alarma “Telegram”.....	42
5.- Construcción.....	44
5.1.- Conexión periféricos.....	45
5.1.1.- Esquema conexión batería raspberry pi.....	46
5.1.2.- Esquema conexión batería, bombilla y raspberry pi.....	47
5.1.3.- Esquema conexión sensor volumétrico raspberry pi.....	48
5.2.- Instalación y configuración de software.....	49
5.2.1.- Instalación y configuración del Sistema Operativo.....	49
5.2.2.- Instalación de periféricos.....	52
5.2.3.- Configuración para el servidor no-ip.....	55
5.2.4.- Programa para la obtención de imágenes “Motion”.....	56
5.2.5.- Lenguaje de programación “Python”.....	57
5.2.6.- Programa de comunicación usuario-alarma “Telegram”.....	59
6.- Montaje y pruebas.....	64
6.1.- Pruebas laboratorio.....	68
6.2.- Pruebas de campo.....	75

7.- Mediciones consumos.....	82
8.- Conclusiones.....	84
9.- Agradecimientos.....	86
10.- Bibliografía.....	88

## Índice de ilustraciones

Ilustración 1: Raspberry pi Modelo B.....	2
Ilustración 2: GPIO's raspberry pi B+.....	4
Ilustración 3: Sensor presencia.....	5
Ilustración 4: Sensor presencia Mixto.....	6
Ilustración 5: Características sensor de presencia del proyecto.....	7
Ilustración 6: Sensor de presencia vista conexiones.....	7
Ilustración 7: Sensor presencia vista frontal.....	7
Ilustración 8: Componentes electrónicos para montaje.....	9
Ilustración 9: Placa prototipo para montaje.....	10
Ilustración 10: Placa prototipo soldadura.....	10
Ilustración 11: Condensador de 100nF.....	11
Ilustración 12: Condensador de 220µF.....	11
Ilustración 13: Resistencia de 10KΩ.....	13
Ilustración 14: MOSFET.....	14
Ilustración 15: Regulador MC7805CT.....	15
Ilustración 16: Cargador de coche.....	16
Ilustración 17: Cargador de coche desmontado.....	16
Ilustración 18: Componente interno del cargador de coche.....	17
Ilustración 19: Características de la webcam.....	18
Ilustración 20: Mapa mundial de la señal GSM.....	20
Ilustración 21: Módem 3G delantera.....	20
Ilustración 22: Módem 3G trasera.....	20
Ilustración 23: Batería.....	21
Ilustración 24: Batería a utilizar.....	22
Ilustración 25: Símbolo diodo.....	23
Ilustración 26: Símbolo led.....	23
Ilustración 27: Bombilla 220v.....	24
Ilustración 28: Bombilla 12v vista lateral.....	24
Ilustración 29: bombilla 12v vista frontal.....	24
Ilustración 30: Esquema eléctrico general.....	33
Ilustración 31: Esquema eléctrico conexión batería.....	34
Ilustración 32: Esquema eléctrico conexión bombilla.....	35
Ilustración 33: Esquema eléctrico conexión sensor volumétrico.....	36
Ilustración 34: Captura de pantalla inicio sistema “rasbian”.....	38
Ilustración 35: Captura pantalla configuración módem 3G.....	42
Ilustración 36: Captura pantalla 2 configuración módem 3G.....	43
Ilustración 37: Montaje en placa de prototipos.....	52
Ilustración 38: Montaje placa prototipos completo.....	52
Ilustración 39: Montaje placa prototipo soldadura.....	53

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

Ilustración 40: Montaje placa prototipo soldadura con los puntos de soldadura.....	54
Ilustración 41: Montaje soldadura GPIO.....	54
Ilustración 42: Montaje del sistema con sensor y bombilla.....	55
Ilustración 43: Montaje del sistema con sensor y bombilla, activado de sensor.....	55
Ilustración 44: Montaje del sistema con sensor y bombilla, bombilla encendida.....	56
Ilustración 45: Prueba conexión “Telegram”.....	57
Ilustración 46: Prueba detección presencia con “Telegram”.....	58
Ilustración 47: Montaje del sistema con sensor y bombilla, 2.....	59
Ilustración 48: Montaje del sistema con sensor y bombilla, 3.....	59
Ilustración 49: Prueba detección presencia con imagen en “Telegram”.....	60
Ilustración 50: Prueba del sistema completo.....	61
Ilustración 51: Prueba del sistema completo con bombilla encendida.....	61
Ilustración 52: Prueba de petición de imagen del usuario “Telegram”.....	62
Ilustración 53: Prueba vídeo en “streaming”.....	63
Ilustración 54: Prueba de vídeo en “streaming” 2.....	63



## 1.- Introducción

Valencia es una zona con muchas explotaciones agrarias, la mayoría de ellas son minifundios, casi 7.000, donde se cultivan muchos cítricos y en menor cantidad melocotones, uvas, ciruelas, albaricoques. Muchas de estas explotaciones disponen de una caseta de aperos o caseta de campo, donde se guarda la maquinaria, o bien son un lugar de escapada.

Cada vez son más numerosos los robos en las zonas agrícolas, lo que supone un importante trastorno económico tanto para los productores agrícolas como para los propietarios de casetas, en los que no sólo se roban las cosechas, también metal, contadores eléctricos, canalizaciones de riego, etc. Según la ASAJA (Asociación Agraria de Jóvenes Agricultores), por provincias, Valencia es la zona más castigada, con 6 robos diarios.

No hay que olvidar que la mayoría de estas explotaciones agrícolas no disponen de corriente eléctrica ni servicios de telefonía en las dependencias donde se guarda la maquinaria y materiales. Es pues para esta tipología de entorno que tiene difícil o ningún acceso a corriente eléctrica y servicio de internet, para el que está diseñado el proyecto.

El abastecimiento eléctrico en estas zonas suele realizarse a través de baterías, grupos electrógenos, placas solares y/o generadores eólicos, etc., y la conexión a Internet se realiza a través de un módem 3G, o si la zona está cerca de población, cabe la posibilidad de tener conexión a través de algún servicio de wifi.



## 2.- Objetivo

El objetivo del proyecto es diseñar, construir, probar y poner en funcionamiento, un sistema de detección de intrusión en zonas rurales, de manera que el propietario esté en todo momento informado de la situación de la zona controlada.

Tiene que ser un sistema lo más autónomo posible, capaz de detectar la intrusión e iluminar la zona, enviar un mensaje instantáneo a través de correo electrónico o sms y almacenar en un disco virtual en la nube o un servidor FTP determinado, una imagen o un vídeo, con un consumo mínimo de energía. En detalle, para que el sistema de detección de intrusión sea efectivo, debe ser capaz de:

- Detectar una intrusión.
- Encender una bombilla.
- Hacer una foto y enviarla por mensaje a un móvil u otro dispositivo.
- Grabar vídeo durante un tiempo determinado siempre que no se vuelva a detectar una intrusión, sino volverá a hacer otro vídeo o continuará con el mismo.
- Al terminar de hacer el vídeo por detección de intrusión debe subirlo a un disco virtual en la nube o a un servidor ftp, de manera que se quede una copia o el original en la nube, por si el dispositivo es sustraído.
- El usuario debe poder ver en cualquier momento lo que está siendo captado por la cámara.
- Poder hacer una foto a voluntad o un vídeo.
- Apagar o encender el sistema de alarma de manera que no grabe vídeo ni envíe foto al detectar movimiento, pero sí poder ver en tiempo real la imagen de la cámara y hacer foto o vídeo a voluntad.
- Apagado total del sistema, de manera que el usuario para encenderlo debe hacerlo físicamente y no remotamente.

El programa que se utilizará en el proyecto para la captura de imágenes será "Motion". Aunque por sí solo es capaz de detectar cambios en las imágenes y hacer una grabación al detectar movimiento, requiere de un análisis de imágenes continua. Esto supone un esfuerzo para la CPU y el correspondiente consumo eléctrico, por lo que se ha decidido utilizar "Motion" sólo para generar el vídeo en "streaming" y la toma de fotos y vídeos cuando el sistema lo requiera o se demande por parte del usuario.



También se tendrá en cuenta el coste del sistema ya que puede ser susceptible de rotura o robo por parte de los malhechores. Así pues, el conjunto deberá ser lo más sencillo de reponer y económicamente viable.

Actualmente las soluciones disponibles en el mercado no contemplan todos estos requerimientos y las que más se aproximan son muy costosas, además de no cumplir con el requisito de bajo consumo energético.

La solución planteada pasa por implementar un hardware de bajo coste ligado a un software libre o gratuito. Para ello se debe realizar un estudio de mercado para saber qué dispositivos son los que se adecuan al producto final.

Las opciones para realizar el proyecto son:

- Ordenadores personales de sobremesa y los portátiles. Aunque hoy en día los portátiles han mejorado mucho el consumo energético, los precios tanto de portátiles como de sobremesa no son muy asequibles y las prestaciones de éstos están por encima de nuestras necesidades. No son adecuados para el proyecto, ya sea por su consumo, precio y aunque no tan importante, también por sus dimensiones ya que no deben ser muy vistosos.
- Sistemas microcontroladores de diseño ad-hoc o diseño propio. Para ello se necesita de un proceso de diseño del hardware para más tarde construir el equipo y generar el software para dicho hardware. Todo ello hace que sea demasiado complejo y costoso económicamente. Es una opción muy interesante en el caso de hacer muchas unidades, pero para el proyecto no es el caso.
- Sistemas microcontrolador o microprocesador de propósito general: Computadores en una placa basados en microprocesador o microcontrolador, que incluyen conexiones para periféricos similares a los de los ordenadores personales. En un principio los dispositivos de menor coste energético y económico del mercado son los sistemas “*Arduino*” , “*Raspberry Pi*” y los derivados o competencia de este último como por ejemplo el “*Banana Pi*”.

En un primer estudio, se descarta el sistema “*Arduino*” ya que necesita de componentes adicionales que encarecen el producto final y carece de potencia suficiente para realizar todas las tareas estipuladas en el proyecto. También se desestima el sistema “*Banana Pi*” por su inestabilidad ya que se necesita un reinicio físico no compatible con las necesidades del proyecto.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

El sistema "*Raspberry Pi*" es el más adecuado y el elegido por su relación entre el hardware y el software ya que no suele requerir reinicio físico por cuelgues del sistema en comparación con sus competidores más próximos. Además, dispone de un gran apoyo de la "*Comunidad Raspberry Pi*". Otros componentes que deberá tener son:

- Un sistema de conexión a internet.
- Un dispositivo que detecte la intrusión.
- Una cámara para grabar vídeo o realizar fotografías.
- Un sistema de aviso luminoso al intruso.



## Sistema de vigilancia de bajo coste energético con Raspberry Pi

### **3.- Materiales**

Será necesario para construir el sistema los siguientes componentes que deberán cumplir con los requisitos de bajo coste energético y económico:

- Raspberry pi, encargada de gestionar todos los dispositivos conectados y realizar las funciones deseadas.
- Sensores de movimiento volumétricos que detectan la presencia de personas, y que deben ser antisabotaje.
- Relés, resistencias, condensadores, placas prototipo, etc....
- Cámaras para la realización de foto o vídeo. (Cámara webcam usb).
- Dispositivo lumínico. (Bombilla led).
- Conexión a Internet a través de 3G. (Módem usb 3G).
- Batería de 12V

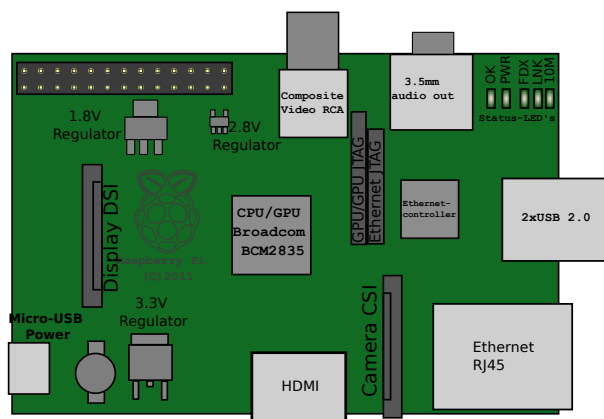
### 3.1.1.- Raspberry pi

“Raspberry Pi” es un ordenador de bajo coste integrado en una placa con procesador ARM diseñado por la fundación que lleva el mismo nombre. Estas placas empezaron su desarrollo en el 2006 y no fue hasta agosto de 2011 cuando se pudieron adquirir, pero eran más grandes que las que podemos conseguir hoy en día ya que necesitaban dicho tamaño para poder realizar la depuración de la misma. Pero no fue hasta finales de febrero de 2012 cuando nació la “Raspberry Pi” que conocemos en la actualidad con la finalidad de estimular el aprendizaje en programación y acercar el mundo de la informática a las escuelas.

Aunque la primera intención de la Fundación “Raspberry Pi” era introducir la programación en las escuelas con su ordenador de bajo coste, ésta se vio desbordada por usuarios particulares que empezaron a adquirirlo para otros usos.

En la breve historia de la placa se han desarrollado diversos modelos que han ido evolucionando con sutiles mejoras. Después de las primeras placas beta, el 29 de febrero de 2012, se pusieron a la venta 10.000 unidades de los modelos A y B, agotando las existencias al poco de su lanzamiento.

Esquema básico del modelo B



*Ilustración 1: Raspberry pi Modelo B*

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

Tabla raspberry pi consumos:

Producto	Alimentación recomendada	Máximo consumo de los periféricos de usb conectados	Consumo típico del dispositivo
Raspberry Pi Model A	700mA	500mA	200mA
Raspberry Pi Model B	1.2A	500mA	500mA
Raspberry Pi Model A+	700mA	500mA	180mA
Raspberry Pi Model B+	1.8A	600mA/1.2A (switchable)	330mA
Raspberry Pi 2 Model B	1.8A	600mA/1.2A (switchable)	

Tabla raspberry pi Modelos

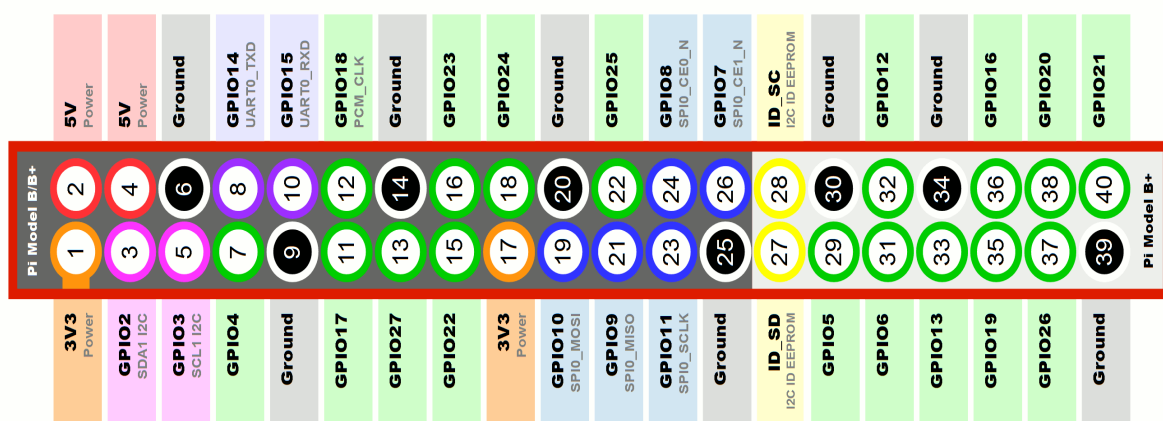
	Modelo A	Modelo B	Modelo B+
SoC:5	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM + puerto USB)		
CPU:	ARM 1176JZF-S a 700 MHz (familia ARM11)		
Juego de instrucciones:	RISC de 32 bits		
GPU:	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia), 1080p30 H.264/MPEG-4 AVC		
Memoria (SDRAM):	256 MiB (compartidos con la GPU)	512 MiB (compartidos con la GPU) desde el 15 de octubre de 2012	
Puertos USB 2.0:	1	2 (vía hub USB integrado)	4
Entradas de vídeo:	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la RPF		
Salidas de vídeo:	Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), Interfaz DSI para panel LCD		
Salidas de audio:	Conector de 3.5 mm, HDMI		
Almacenamiento integrado:	SD / MMC / ranura para SDIO		MicroSD
Conectividad de red:	Ninguna	10/100 Ethernet (RJ-45) via hub USB	
Periféricos de bajo nivel:	8 x GPIO, SPI, I <sup>2</sup> C, UART		
Reloj en tiempo real:	Ninguno		
Consumo energético:	500 mA, (2.5 W)	700 mA, (3.5 W)	600 mA, (3.0 W)
Fuente de alimentación:	5 V vía Micro USB o GPIO header		
Dimensiones:	85.60mm × 53.98mm (3.370 × 2.125 inch)		
Sistemas operativos soportados:	GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux. RISC OS		

### Consumos eléctricos:

- La alimentación de corriente "Raspberry Pi" es de 5v y el consumo de los amperios depende del modelo y de los periféricos que conectemos a la raspberry pi.
  - El Modelo B puede consumir entre 700-1000mA dependiendo de los periféricos
  - El Modelo A puede consumir desde los 500mA sin periféricos conectados hasta los 1000mA como máximo con periféricos
- Las conexiones por GPIO para los dos modelos, pueden consumir 16mA cada uno con un total máximo de 50mA entre todos.
- La conexión HDMI son 50 mA, no es nuestro caso ya que no se va a utilizar en el uso normal del proyecto, solamente al principio para configurar el sistema operativo y la red para poder conectarnos por ssh y hacer el resto de la configuración.
- La cámara propia de raspberry pi presenta un consumo de 250mA, pero utilizaremos otra por el puerto usb con un consumo de 43mA.

Debido a que previamente ya se disponía de un dispositivo "Raspberry Pi" B+, será el utilizado para el proyecto, aunque cabe recordar, tal y como se recoge en la tabla superior, que el modelo 2A tiene unos consumos energéticos y económicos menores.

Para las conexiones para la GPIO utilizaremos el esquema siguiente, correspondiente a la raspberry pi 35B+:



www.raspberrypi-spy.co.uk

Ilustración 2: GPIO's raspberry pi B+

### 3.1.2.- Sensor movimiento

Existen varios tipos de sensores volumétricos:

Los detectores por infrarrojos, los detectores por microondas y los que utilizan los dos sistemas: infrarrojos y microondas.

Los detectores de infrarrojos: utilizan la variación de temperatura del ambiente y la de cualquier cuerpo, de manera que si se produce una diferencia de grados centígrados en el campo de trabajo del sensor, lo detecta y envía la señal de alarma.

Puede dar falsos positivos por el cambio brusco de temperatura en el ambiente, por aparatos de aire acondicionado, ventanas abiertas, etc.



*Ilustración 3: Sensor presencia*

Los de microondas: utilizan el efecto Doppler, que consiste en que un emisor emite una señal con una frecuencia determinada y los objetos presentes la reflejan y si estos se desplazan, la frecuencia recibida es distinta a la emitida, por lo que detecta el movimiento y envía la señal de alarma. En desuso por la gran cantidad de falsos positivos. Además los inconvenientes son varios, no se pueden utilizar en dirección a las ventanas ya que las microondas atraviesan el cristal, no puedes tener dos sensores enfrentados ya que si se utiliza más de uno, deben trabajar en frecuencias distintas.



Alberto Toledo Escrihuela.

Los Mixtos: utilizan las dos tecnologías descritas anteriormente combinadas y sólo envía la señal de alarma en el caso de que las dos tecnologías detecten la intrusión, de esta manera se evitan muchos falsos positivos.



*Ilustración 4: Sensor presencia Mixto*

Para el proyecto desestimamos los sensores por microondas debido a que están en desuso por su alto número de detecciones erróneas y los mixtos por su alto coste económico y energético.

La elección pasa por los sensores de movimiento por infrarrojos debido a su coste económico y a su consumo que varía según modelo de 10 a 120mA. Es la mejor opción, y se ha optado por el modelo RXC-ST PIR QUAD cuyo consumo energético es de 11mA como máximo según el fabricante, frente a otros cuyo consumo varía entre 18mA a 120mA.

Como norma general se cablea con un cable multihilos apantallado de 6 hilos, dos hilos para la alimentación, dos para la señal de alarma y dos para el sistema antisabotaje (tamper), de manera que si se abre la tapa del sensor éste envía la señal de alarma.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

En la siguiente imagen se pueden ver las instrucciones de montaje del sensor que se va a utilizar:

En la imagen se puede apreciar cuatro pares de conexiones de izquierda a derecha son:

- Entrada de corriente de 12 voltios positivo y negativo,
- Un par de conectores libres que no están conectados a nada,
- Otro par de conectores para la señal de alarma
- Y el ultimo par de conectores para el sistema antisabotaje o también conocido como Tamper.

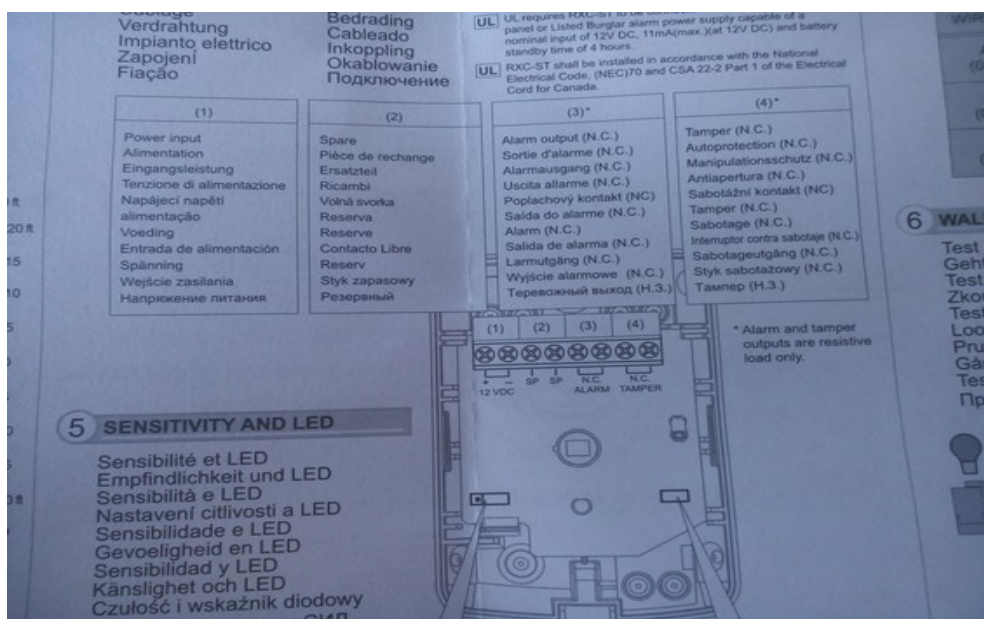


Ilustración 5: Características sensor de presencia del proyecto

En estas imágenes podemos apreciar el dispositivo y sus conexiones reales:



Ilustración 6: Sensor de presencia vista conexiones



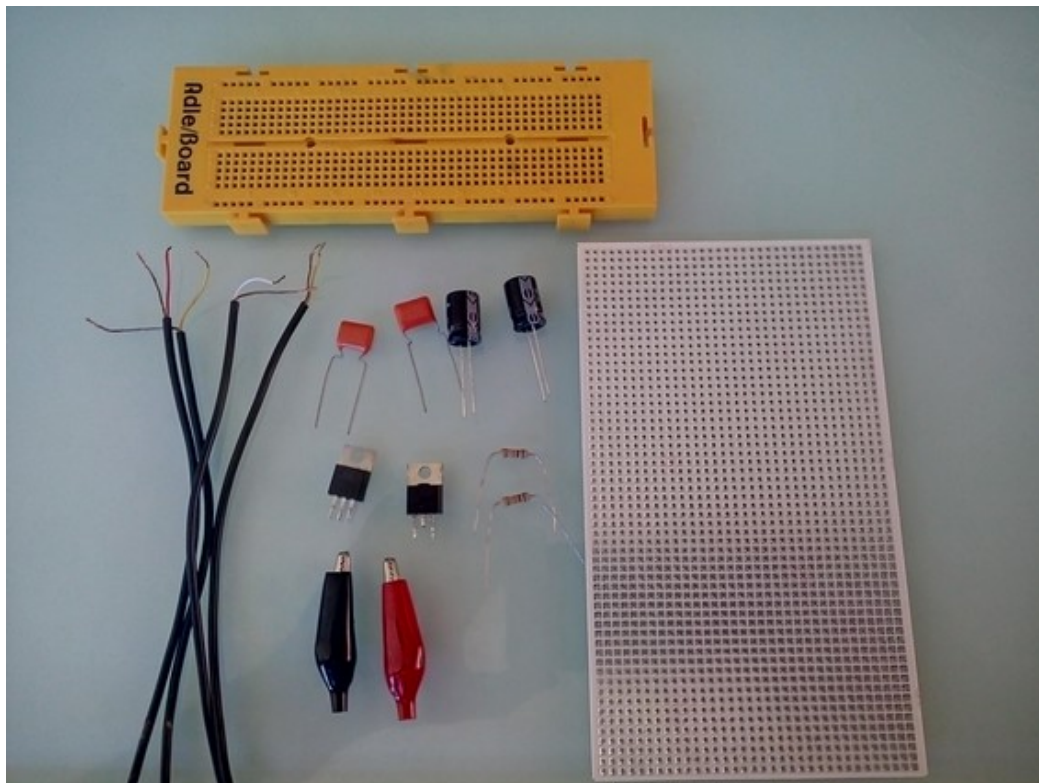
Ilustración 7: Sensor presencia vista frontal

### **3.1.3.- Relés, Resistencias, Condensadores, Placas Prototipo, etc....**

Componentes electrónicos necesarios para realizar el proyecto son:

- Placa prototipo de soldadura.
- Dos condensadores de 100 nano faradios.
- Dos condensadores de 220 micro faradios.
- Dos resistencias de 10 kilo Ohmios.
- Cables de cobre.
- MOSFET IRZ24N
- Regulador MC7805CT.
- Cargador de móviles para el coche.
- Pinzas de cocodrilo.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

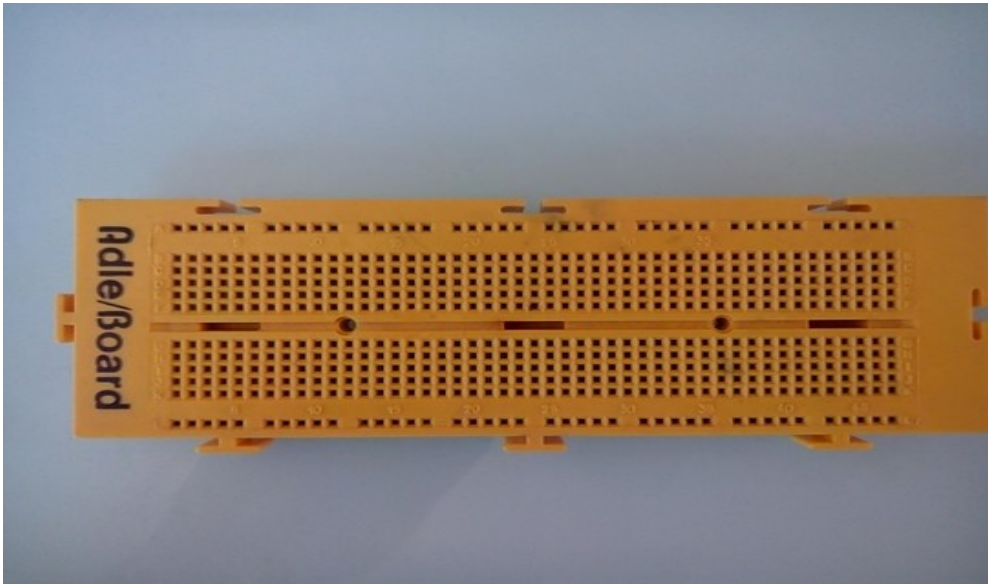


*Ilustración 8: Componentes electrónicos para montaje*

- Las pinzas de cocodrilo y los cables se utilizarán para hacer las conexiones.
- Placas de prototipo:

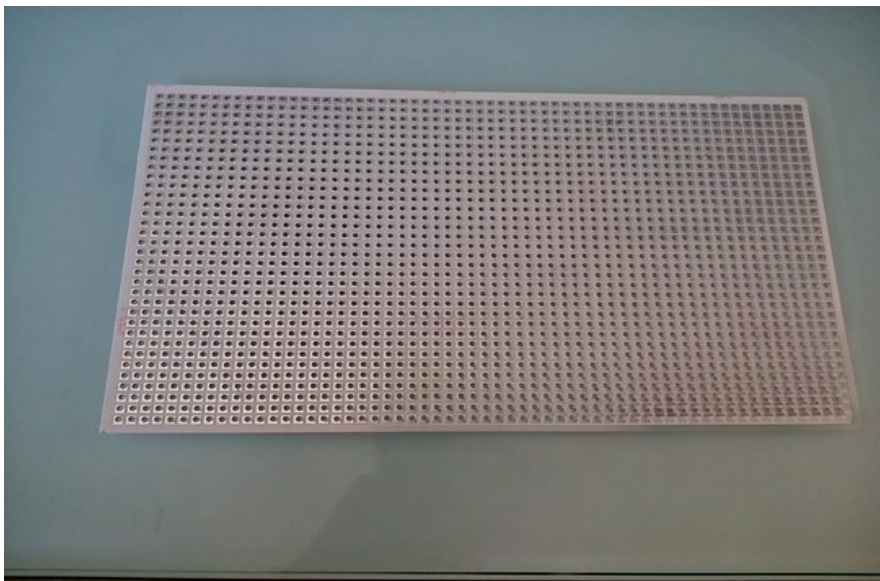
Son aquellas en las que podemos montar fácilmente un circuito eléctrico, en la imagen podemos observar la placa de prototipo de color amarillo, la otra placa de prototipo de soldadura es la de color blanco con muchos taladros rebordeados de estaño. Se usan básicamente para hacer prototipos ya que su coste es reducido y permite ensamblarla muchas veces en comparación con una placa de impresión fotosensible.

Imagen placa prototipo montaje rápido:



*Ilustración 9: Placa prototipo para montaje*

Imagen placa prototipo soldadura:

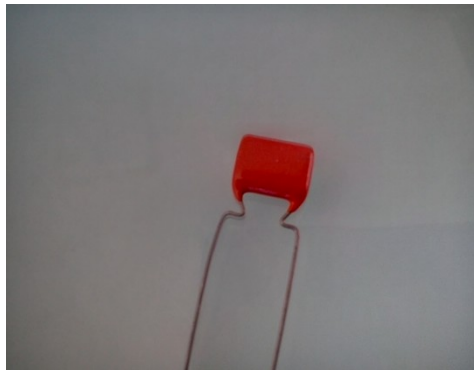


*Ilustración 10: Placa prototipo soldadura*

- Condensadores de 100nF.

Son condensadores cerámicos que no tienen polarización por lo que se pueden utilizar para circuitos de corriente alterna o polarización variable.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi



*Ilustración 11: Condensador de 100nF*

- Condensadores de 220 $\mu$ F.

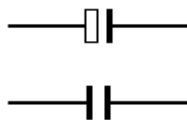
Los condensadores eléctricos o capacitores son dispositivos capaces de almacenar una pequeña cantidad de electricidad, son de corriente continua y contienen un dieléctrico que les hace tener mayor capacidad, entre estos existen muchos tipos dependiendo sobretodo del material del dieléctrico. En nuestro caso los condensadores de 220 $\mu$ F son los de la siguiente figura:



*Ilustración 12: Condensador de 220 $\mu$ F*

Son electrolíticos y se utilizan para circuitos de corriente continua.

Símbolos electrónicos:



- Resistencias de 10K $\Omega$ .

Como su nombre indica es un componente el cual ejerce una resistencia al paso de

la corriente eléctrica. Su resistividad viene dada por ohmios y el símbolo es la letra griega omega. Podemos saber los ohmios de una resistencia por la franja de colores que lleva impresa y para saber los valores de una resistencia necesitamos saber tres valores: la resistencia, la tolerancia y disipación máxima:

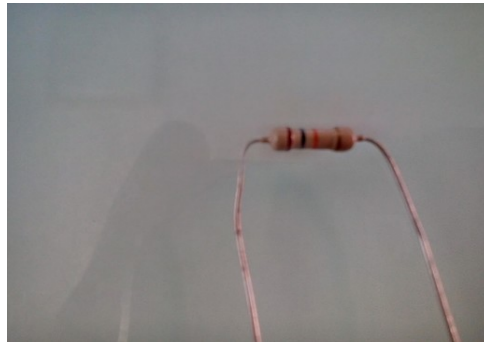
Color de la banda	Valor de la 1ª cifra significativa	Valor de la 2ª cifra significativa	Multiplicador	Tolerancia	Coefficiente de temperatura
Negro	0	0	1	-	-
Marrón	1	1	10	±1%	100ppm/°C
Rojo	2	2	100	±2%	50ppm/°C
Naranja	3	3	1 000	-	15ppm/°C
Amarillo	4	4	10 000	±4%	25ppm/°C
Verde	5	5	100 000	±0,5%	20ppm/°C
Azul	6	6	1 000 000	±0,25%	10ppm/°C
Morado	7	7	10 000 000	±0,1%	5ppm/°C
Gris	8	8	100 000 000	±0.05%	1ppm/°C
Blanco	9	9	1 000 000 000	-	-
Dorado	-	-	0,1	±5%	-
Plateado	-	-	0,01	±10%	-
Ninguno	-	-	-	±20%	-

Para leer el código de colores hay que mirar de izquierda a derecha, pueden haber tres, cuatro o cinco bandas de colores, por lo general suelen ser de cuatro bandas o líneas

- la primera línea o banda representa a las decenas o centenas.
- la segunda línea o banda representa a las unidades o decenas.
- la tercera línea o banda representa bien a una unidad o al multiplicador para la potencia de base 10.
- la cuarta línea o banda representa bien el multiplicador en cuyo caso suele ser de color dorado o plateado, para la potencia de base 10 o bien la tolerancia.
- la quinta línea o banda representa la tolerancia o en su caso el coeficiente de temperatura.

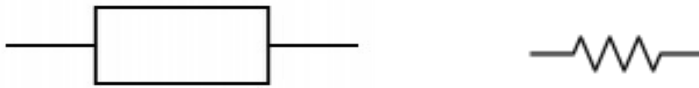
Imagen de las resistencias que se utilizaran en el proyecto

## Sistema de vigilancia de bajo coste energético con Raspberry Pi



*Ilustración 13: Resistencia de 10KΩ*

Símbolos electrónicos:



- MOSFET IRZ24N

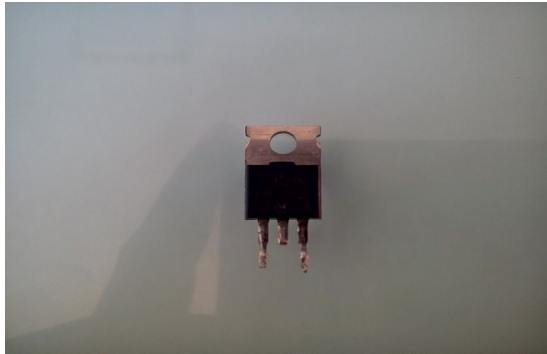
Se trata básicamente de un transistor MOSFET de enriquecimiento, que al aplicarle una tensión en uno de sus conectores conocido como la base, atrae a cierta cantidad de portadores de carga incrementando la conductividad entre los otros dos conectores, llamados emisor y colector, de manera que se genera un cierre de circuito entre el emisor y el colector, dejando así que pase la corriente entre estos dos.

Los transistores MOSFET de enriquecimiento, a diferencia de los de empobrecimiento, tienen el circuito cerrado sin aplicarle corriente. Estos últimos se abren al aplicarle corriente en la compuerta, de manera que si suministramos tensión en la misma, el circuito entre emisor y colector se abre cortando el paso de corriente entre estos. Por el contrario, el enriquecido tiene el circuito abierto en reposo, y cerrado al aplicarle corriente a la base, por lo que es más provechoso para el proyecto ya que mantiene abierto el circuito sin necesidad de consumo eléctrico y lo cierra para que pase la electricidad al aplicarle una pequeña corriente al conector base.

Para nuestro proyecto este transistor nos permitirá desde la GPIO de la "Raspberry Pi", con una baja tensión, controlar el cierre y apertura del circuito de alimentación entre la batería y la bombilla de 12 voltios.

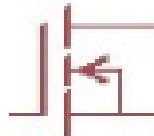


Imagen del MOSFET que se utilizara en el proyecto.



*Ilustración 14: MOSFET*

Símbolo electrónico:



- Regulador MC7805CT

Aunque no se va a utilizar en el proyecto, en un principio era una pieza fundamental para el convertidor de corriente de 12 voltios de la batería a 5 voltios para la entrada de corriente de la “Raspberry Pi”. Más tarde se sustituyó por otro sistema que era energéticamente más eficiente aunque un poco más caro. Aun así la relación eficiencia - coste, fue la que empujó a hacer el cambio.

Este regulador tiene tres conectores o terminales, de los cuales uno es de entrada, otro común o de masa y el tercero de salida. Es un regulador lineal lo que significa que está en serie con la carga, por lo que en la caída de tensión produce una gran disipación de potencia.



*Ilustración 15: Regulador MC7805CT*

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

- Cargador de móviles para el coche.

Hoy en día es muy común tener un cargador de coche para nuestro móvil. Estos dispositivos al ser tan comunes han reducido su coste drásticamente y para el proyecto resultan muy útiles ya que trabajan con la corriente que proporciona el coche que son 12 voltios al igual que la fuente de alimentación que va a ser utilizada en el proyecto. Éste la transforma en 5 voltios que es lo que se necesita para cargar el teléfono móvil, igual que la “Raspberry Pi” para su funcionamiento, por lo que este dispositivo, con una pequeña modificación, se puede adaptar para que funcione directamente en la batería que se utilizará.

Este dispositivo sustituirá al Regulador MC7805CT, ya que es mucho más eficiente energéticamente y su coste no es desproporcionado para el proyecto.

Internamente lleva un convertidor MC34063 el cual a diferencia del MC7805 lineal, que se tenía previsto utilizar en un principio, puede trabajar con tensiones de entrada desde los 3 voltios hasta los 40 voltios y la disipación es mucho menor, por lo que la eficiencia del MC34063 es mayor.



*Ilustración 16: Cargador de coche*

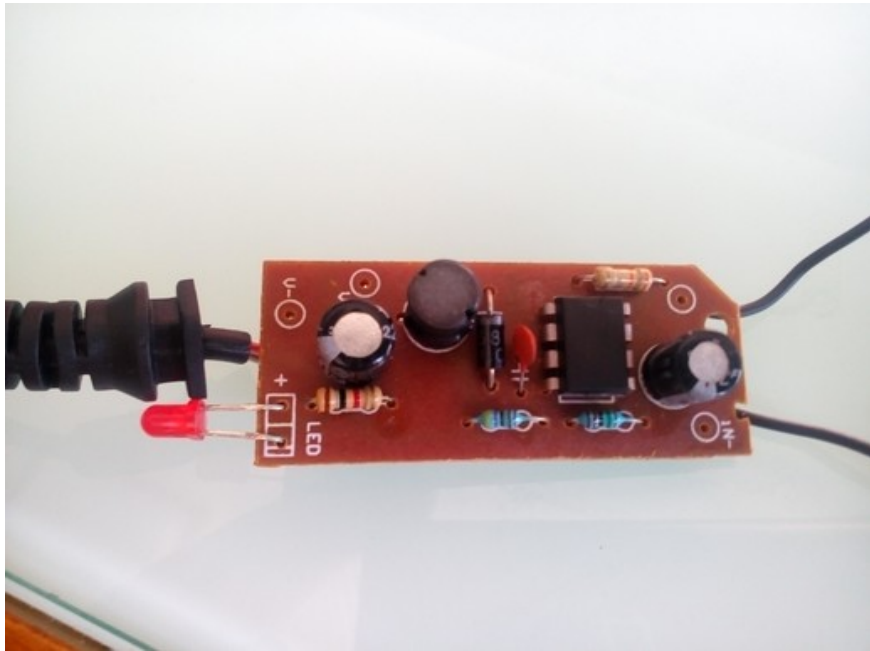
### Dispositivo cargador de móviles para coche



*Ilustración 17: Cargador de coche desmontado*

Alberto Toledo Escrihuela.

Desmontaje del dispositivo para poder incluirlo en el esquema eléctrico final.



*Ilustración 18: Componente interno del cargador de coche*

Este es el circuito que se extrae para la fuente de alimentación con el que pasaremos de 12 voltios a 5 voltios.

### 3.1.4.- Cámara (Webcam)

Hoy en día hay infinidad de cámaras, con infinidad de precios, y el mundo de las webcam no es menos, incluso podemos encontrar cámaras webcam con infrarrojos para la visión nocturna. Para el proyecto con una cámara sencilla nos basta, por lo que se ha optado por una webcam por USB de bajo coste, por debajo de los 10€, con una resolución de 640x480 que es suficiente para lo deseado. Especificaciones de la cámara:

**PC CAMER**  
**Mini packing**

**Features:**

CMOS chip Type: Color CMOS image sensore  
Hi-resolving power: 640x480  
Video Format. 24Bit RGB  
Interface: USB  
Frame Rate: 320x240 up to 30 frames/sec(cif)  
640x480 up to 15 frames/sec(VGA)  
S/N ratio: 48dB  
Dynamic range: 72dB  
Focus range: 3cm-infinity  
Buil-in image compresion  
Automatic white balance  
Automatic color compensated  
Dynamic image E-mail  
Manual focus  
Shutter key

*Ilustración 19: Características de la webcam*

### 3.1.5.- Sistema 3G, conexión Internet

Para la conexión a Internet se utilizará un módem 3G usb , el cual proporciona una conexión a Internet a través de datos móviles, por el puerto usb.

Para el proyecto se hará uso del modelo k3765 de HUAWEI TECHNOLOGIES CO. Cuyas características son:

Soporta los siguientes estándares: HSUPA, HSDPA, UMTS, EDGE, GPRS, GSM. HSUPA, HSDPA y UMTS para navegar por Internet de forma rápida.

EDGE y GPRS para navegar por Internet si no hay cobertura 3G.

GSM para enviar mensajes SMS.

Ranura para SIM: SIM de tamaño estándar.

Ranura para tarjeta microSD de hasta 32GB de capacidad.

Interfaz: USB 2.0 para PC y Mac con los correspondientes drivers y software de Vodafone. Alimentación: 5V de tensión de corriente y 0,5A de intensidad de corriente, especificaciones estándar de los USB.

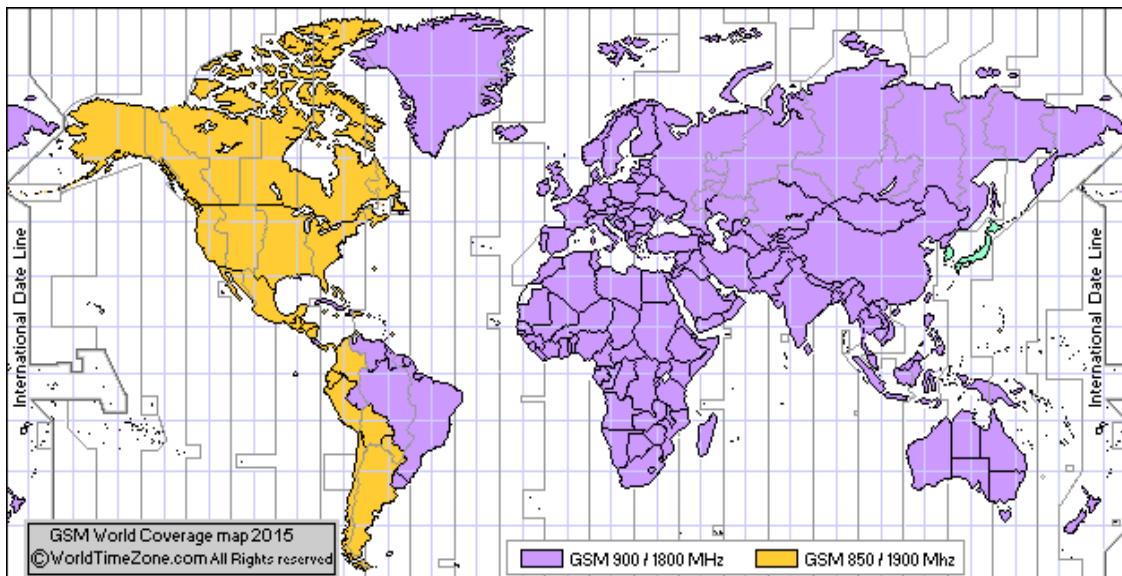
Las frecuencias del módem 3G Huawei k5765 son:

- Para redes 3G las bandas HSDPA y UMTS de frecuencias: 2100MHz y 900MHz.
- Para redes 2G las bandas EDGE, GPRS, GSM de frecuencias: 1900MHz, 1800MHz, 900MHz y 850MHz.

Dichas frecuencias son las que se trabaja en Europa, para GSM 900MHz y 1800MHz, mientras que para HSDPA son la 2100MHz y 900MHz, como se muestra en la imagen para GSM:

imagen GSM Mundial:

## Sistema de vigilancia de bajo coste energético con Raspberry Pi



*Ilustración 20: Mapa mundial de la señal GSM*

La velocidad de transmisión depende en la banda con la que se puede conectar y son:

- HSDPA hasta 7,2Mbps de descarga y HSUPA hasta 5,76Mbps de subida.
- EDGE hasta 326,8Kbps.
- GPRS hasta 85,6Kbps.

Imagen del modulo 3G



*Ilustración 22: Módem 3G trasera*



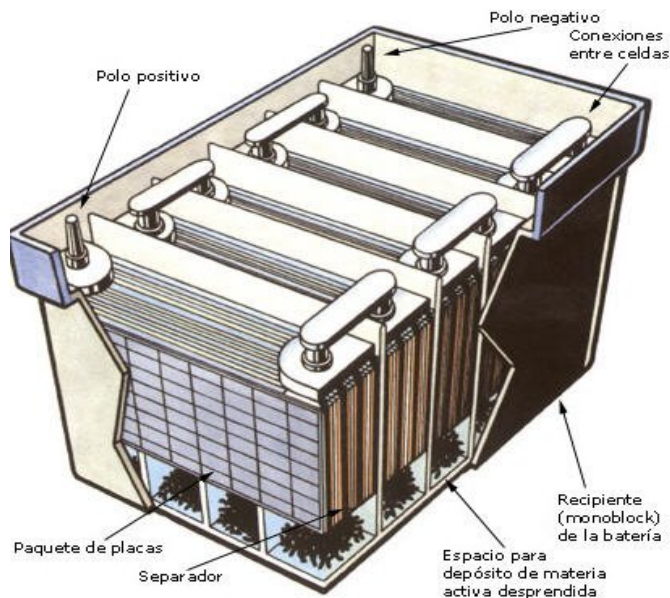
*Ilustración 21: Módem 3G delantera*

### 3.1.6.- Batería

Hoy en día las baterías de pequeño tamaño han aumentado espectacularmente su capacidad de carga y han disminuido también el tiempo de recarga con poco efecto memoria y bajo peso. Son las llamadas baterías de ion-litio, que aunque son de gran capacidad, no es suficiente. Es por ello que nos decantamos por las baterías de plomo, que aunque son más grandes y pesadas, la capacidad de almacenar energía es mucho mayor y económicamente son viables para el proyecto.

Las baterías convencionales de los coches están constituidas por acumuladores de ácido-plomo, los cuales están compuestos por láminas de plomo que son el polo positivo, láminas de dióxido de plomo que son el polo negativo y bañadas por una disolución de agua destilada y ácido sulfúrico. De esta forma puede generar una diferencia de potencial de 2,12 voltios cada acumulador, con lo que si se juntan 6 acumuladores obtenemos la diferencia de potencial de 12 voltios, de las baterías convencionales.

Imagen batería:



*Ilustración 23: Batería*

Los datos a tener en cuenta son: voltaje, amperios y amperios – hora.

Voltaje (V): en nuestro caso será de 12 voltios que es lo común en estas baterías.

Amperios (A): Potencia que puede dar la batería.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

Amperios – hora (AH): Capacidad de energía de la batería. Esta es la característica que más tenemos que tener en cuenta ya que es la que nos limitará el tiempo que puede estar el dispositivo sin cambiar o recargar.

En cuanto a la duración de la batería, podemos hacer el cálculo aproximado de las horas que podemos tener en marcha el dispositivo. Por ejemplo: tenemos una batería de 100Ah de 12v y todos los dispositivos conectados tienen un consumo de 10W, calculamos  $100\text{Ah} \times 12\text{v} = 1200\text{Wh}$  acumulados, por lo que los multiplicamos por el consumo total 10W y obtendremos 120h, por lo que tendremos 120 horas de duración de la batería aunque no la podemos consumir toda ya que la estropearíamos, por ello mantendremos la carga entre un 50% y un 75% por lo que nos quedarían entre 60 y 90 horas de consumo.

Para las pruebas de laboratorio se utilizará una batería pequeña de ácido-plomo de la marca Energivm modelo MVH1272F2 que es de 12 voltios con un amperaje máximo de 2,16A que aunque esté un poco desgastada (con muchos ciclos de carga descarga) es suficiente para realizar las pruebas.

Imagen batería pruebas:



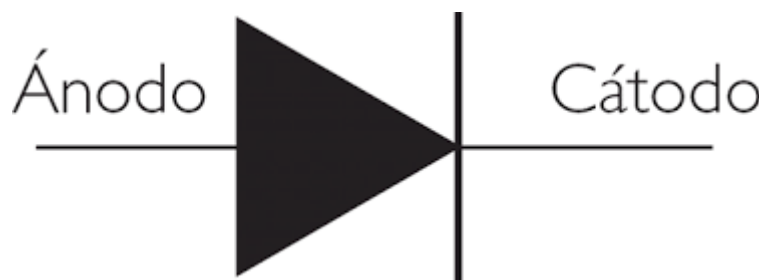
*Ilustración 24: Batería a utilizar*



### 3.1.7.- Bombilla

LED (light emitting diode) siglas en inglés que significan diodo emisor de luz. Sus terminales son el ánodo para la parte positiva y cátodo para la parte negativa del diodo, sentido de circulación de la corriente permitido es de ánodo a cátodo.

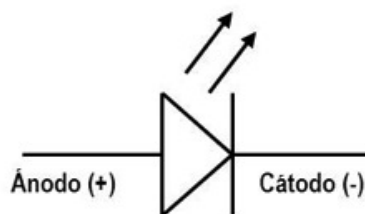
Simbolo eléctrico del diodo:



*Ilustración 25: Símbolo diodo*

El diodo emisor de luz no es más que un diodo que al cruzar la corriente desde el ánodo al cátodo, produce una emisión de fotones y dependiendo del material con el que esté fabricado emitirá un tipo de luz u otro, es decir, emitirá en una longitud de onda específica para cada compuesto.

Símbolo del led.



*Ilustración 26: Símbolo led*

Las bombillas led están formadas por varios leds ya que estos por si solos no son capaces de generar suficiente luz para emular una bombilla incandescente o fluorescente. Para ello se unen grupos de leds para generar más o menos luminiscencia según lo deseado.

Como se puede apreciar en las imágenes, los leds en los dos tipos de bombillas expuestos.

# Sistema de vigilancia de bajo coste energético con Raspberry Pi

Bombilla led 1.



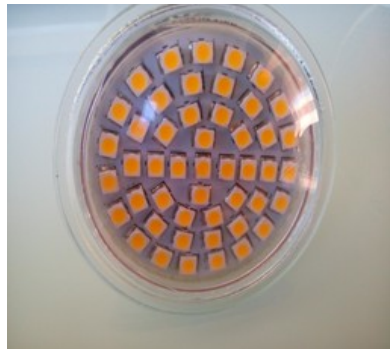
*Ilustración 27: Bombilla 220v*

Bombilla led 2.1



*Ilustración 28: Bombilla 12v vista lateral*

Bombilla led 2.2



*Ilustración 29: bombilla 12v vista frontal*



Sistema de vigilancia de bajo coste energético con Raspberry Pi

## **4.- Software**

El software necesario para el proyecto es el siguiente:

- Sistema Operativo.
- Instalación de los periféricos.
- Configuración para el servidor “no-ip”.
- Programa para la obtención de imágenes “Motion”.
- Lenguaje de programación “Python”.
- Programa de comunicación usuario-alarma “Telegram”.

## 4.1.- Sistema operativo.

El software más importante que se utilizará es el sistema operativo. Uno de los aconsejados en la propia web oficial de la fundación “Raspberry Pi”, es el sistema operativo “*rasbian*” basado en “*Debian*”, una de las más importantes distribuciones *Linux*. Al ser un sistema operativo completo, nos da más opciones para instalar programas ya desarrollados y al ser una distribución “*debian*” posee una de las mayores listas de los mismos y un gran apoyo de personas que se dedican a desarrollar o a responder dudas sobre el software *Linux*. El grupo conocido como comunidad linuxera puede responder a dudas que se puedan plantear en el proyecto.

Para instalar el sistema operativo hay que descargar la imagen desde la web de [www.raspberrypi.org](http://www.raspberrypi.org). Todo el proceso que se requiera de un segundo ordenador se hace desde un sistema Linux ya que éste no requiere de ninguna instalación adicional. Una vez descargada se procede a insertar la imagen en una tarjeta de memoria SD desde un equipo con Linux. Dicha imagen se inserta mediante la siguiente instrucción y se configura tal y como se recoge más adelante en el punto 5.2.1:

`dd if=./rasbian.img of=/dev/sdx`, donde `./` es la dirección completa donde se encuentra la imagen descargada y la “x” de “sdx” es el punto de montaje de la tarjeta SD y suele ser b.

## 4.2.- Instalación Periféricos.

La cámara modelo standard de bajo coste utilizada en el proyecto es detectada automáticamente por el sistema operativo “raspbian”, por lo que no se procede a hacer nada, simplemente se comprueba que la ha detectado, bien comprobando que existe el archivo video0 en /dev, o bien con la instrucción `lsusb` que imprime por pantalla un listado de todos los componentes USB conectados. En el punto 5.2.1 se detalla el proceso de instalación y los elementos hardware utilizados temporalmente. Si fuese un modelo o marca específica de cámara se deberá buscar la compatibilidad con la “Raspberry Pi”, a través de esta web: [http://elinux.org/Rpi\\_VerifiedPeripherals](http://elinux.org/Rpi_VerifiedPeripherals), o bien ver si es capaz de detectarla. Si no fuese así se buscaría por internet añadiendo en el buscador el modelo de cámara más “Raspberry Pi” o “raspbian”.

Para el módem 3G se debe instalar el “*point-point protocol*” (ppp), el driver “Usb-modeswitch” y el programa de conexión “wvdial” o bien el “sakis3g”. Nos declinamos por el “wvdial” debido a que para ponerlo en marcha una vez configurado es una simple instrucción, mientras que el “sakis3g” es en modo gráfico a través de la consola, aunque es posible encontrar la instrucción para ponerlo en marcha sin necesidad del entorno gráfico, pero como no es la finalidad estricta de este proyecto, se ha decidido por el que sí se conoce la instrucción.

### **4.3.- Configuración para el servidor “no-ip”.**

Para acceder a la cámara y ver el video en “streaming”, en tiempo real, desde internet se debe disponer de una IP fija o bien saber en todo momento la IP que se le ha asignado al dispositivo módem 3G, por lo que se debe configurar el sistema para utilizar el servidor gratuito no-ip. Para ello hay que crear una cuenta en noip.org, instalar el programa y hacer la configuración de éste con la cuenta creada en noip.org.

Así mismo, cuando cambia la IP pública del módem 3G, el servidor de no-IP puede tardar un cierto tiempo en cambiar sus registros, por lo que puede que en un momento dado no se tenga acceso al sistema de “streaming” de la “Raspberry Pi”, desde internet.

Al ser gratuito el servicio, hay algunas restricciones, como que cada mes hay que activar el host de la cuenta en la web de Noip para que no se elimine el registro del host. También está el límite de sólo tres host por cuenta, aunque para el proyecto sólo hace falta un host.

Hay una opción en la que se puede prescindir del servicio de Noip. Dicha opción pasa por preguntar a la “Raspberry Pi” la IP pública que tiene con el comando “curl ifconfig.me”, por medio del programa que se utilizará para el proyecto y del cual se detalla más adelante. Una vez responda, se sabrá la IP pública del sistema “Raspberry Pi”, y ya se puede acceder a dicha IP por el puerto fijado para ver el vídeo en “streaming” que emite el programa “Motion”.

#### **4.4.- Programa para la obtención de imágenes “Motion”.**

Para la obtención de imágenes se utilizará el programa “Motion” que es capaz de generar vídeo en “streaming”, grabación de vídeos y toma de fotos desde la cámara USB destinada al proyecto. Como ya se ha dicho al inicio de la memoria, “Motion” sólo se utilizará para generar el vídeo en “streaming” y la toma de fotos y vídeos cuando el sistema lo requiera o se demande por parte del usuario.

El programa “Motion” es fácil de programar para que antes o después de hacer una grabación de vídeo, o al detectar movimiento por imágenes que no es el caso del proyecto, pueda ejecutar cualquier instrucción que se desee; por lo que puede enviar un mensaje, ejecutar instrucciones directamente o a través de “scripts”, o enviar el vídeo que se acaba de grabar, al momento de terminar la grabación.

Si se quisiera ampliar el área de vigilancia, el programa Motion también es capaz de trabajar con “cámaras IP” que son aquellas cámaras que vienen conectadas a través de red no directamente al dispositivo “Raspberry Pi”, por lo que podría ser interesante para conectar un cierto número de cámaras IP a un “switch” que también esté conectada la “Raspberry Pi”. Así se podrían controlar todas dependiendo de la potencia de la CPU del ordenador.



## **4.5.- Lenguaje de programación “Python”.**

Para la detección de la intrusión se utilizará el sensor volumétrico. Con el lenguaje de programación “Python” se controlará las GPIOs de la “Raspberry Pi” detectando la intrusión. En ese momento la función pertinente hará que se encienda la bombilla a través de las conexiones GPIO y se encargará de hacer el aviso al usuario enviándole una foto o texto, a algún dispositivo como puede ser el móvil.

Python fue creado a finales de los 80 por Guido Van Rossum en los Países Bajos. En 1991 se publicó el código de la versión 0.9.0, cuya finalidad es hacer un código legible, pensado para ser usado tanto como lenguaje de programación orientado a objetos, como a programación imperativa o programación funcional. Es multiplataforma, por lo que el código desarrollado en un sistema operativo es perfectamente funcional en otro diferente.

#### **4.6.- Programa de comunicación usuario-alarma “Telegram”.**

Para el manejo del sistema desde otro dispositivo se utilizará una aplicación de mensajería instantánea: “Telegram”. Es muy parecido al ya conocido “Whatsapp”, pero con la diferencia que la parte del cliente es de código abierto, y se pueden hacer aplicaciones a medida de mensajería instantánea utilizando este programa.

Así pues se podrá configurar para asignarle instrucciones a ciertas frases o palabras que deseemos, de manera que cuando la “Raspberry Pi” reciba una palabra clave desde el “Telegram” pueda responder al mensaje de “Telegram” enviado una imagen al usuario que envió la palabra clave, o bien realizar una operación específica previamente configurada, como por ejemplo el reinicio o apagado de la “Raspberry Pi”. También se puede configurar para cuando la “Raspberry Pi” detecte una intrusión, envíe una imagen o un aviso a uno o varios usuarios previamente configurados. También brinda la oportunidad de tenerlo en varios dispositivos a la vez, como teléfonos móviles, tabletas, ordenadores de sobremesa o portátiles entre otros, de manera que el usuario puede enviar instrucciones a la “Raspberry Pi” desde cualquier sitio y en cualquier dispositivo que tenga instalado el “Telegram”, e incluso ser manejado por varios usuarios a la vez.

Hay que tener en cuenta que para utilizar el “Telegram” es necesario un número de teléfono móvil. Podemos utilizar el número de la SIM que se utilizará para el módem 3G, pero para ello habrá que poner la SIM en un móvil para poder recibir el sms con el código de activación o para que nos puedan llamar para decirnos el código. También cabe la posibilidad de adquirir gratuitamente un número de teléfono móvil virtual. Los podemos conseguir desde internet en numerosas páginas habilitadas para ello. En el proyecto se utilizará el número de teléfono móvil del módem 3G.



## 5.- Construcción

En esta sección se explica cómo se deben hacer las conexiones a los diferentes dispositivos que van a ser utilizados, instalación de software y configuración.

- Conexión periféricos.
  - Esquema conexión batería raspberry pi.
  - Esquema conexión batería, bombilla y raspberry pi.
  - Esquema conexión sensor volumétrico raspberry pi.
- Instalación y configuración de software.
  - Instalación y configuración del Sistema Operativo.
  - Instalación periféricos.
  - Configuración para el servidor no-ip.
  - Programa para la obtención de imágenes "Motion".
  - Lenguaje de programación "Python".
  - Programa de comunicación usuario-alarma "Telegram".

## 5.1.- Conexión periféricos

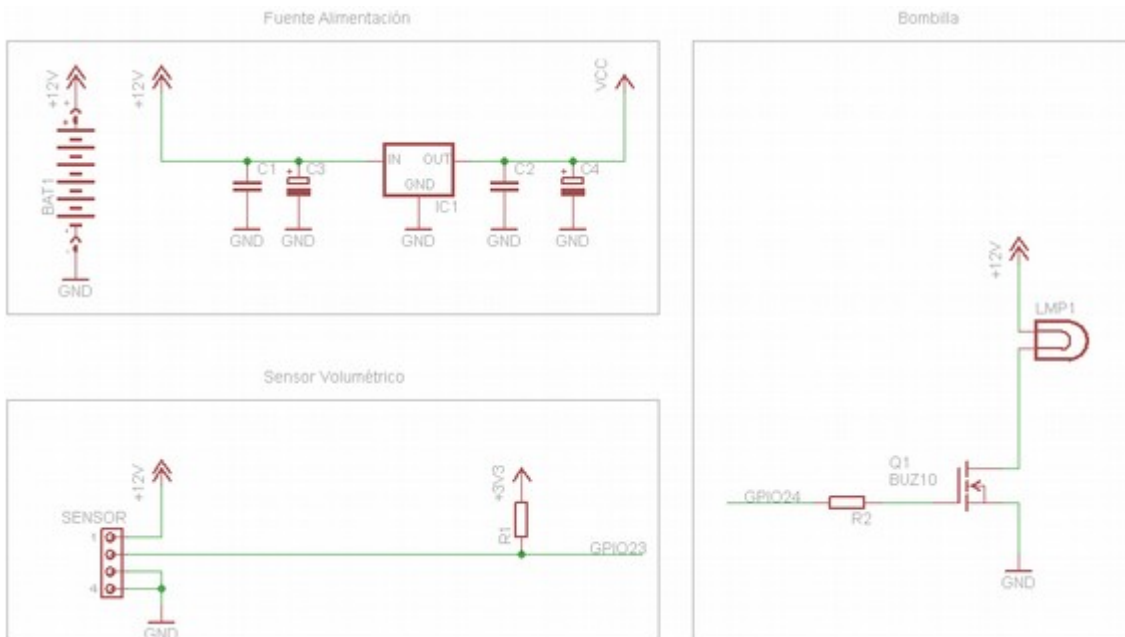
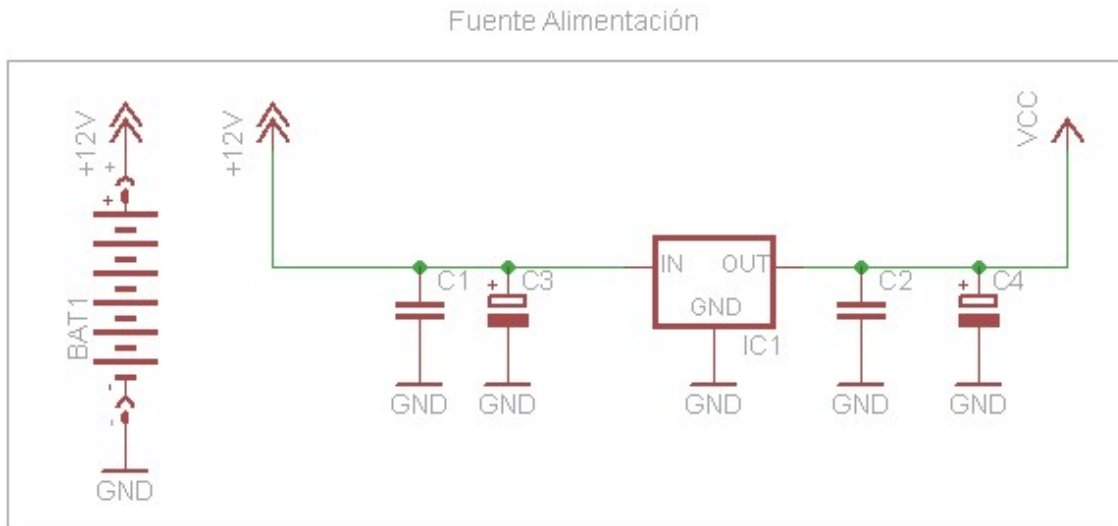


Ilustración 30: Esquema eléctrico general

### 5.1.1.- Esquema conexión batería raspberry pi



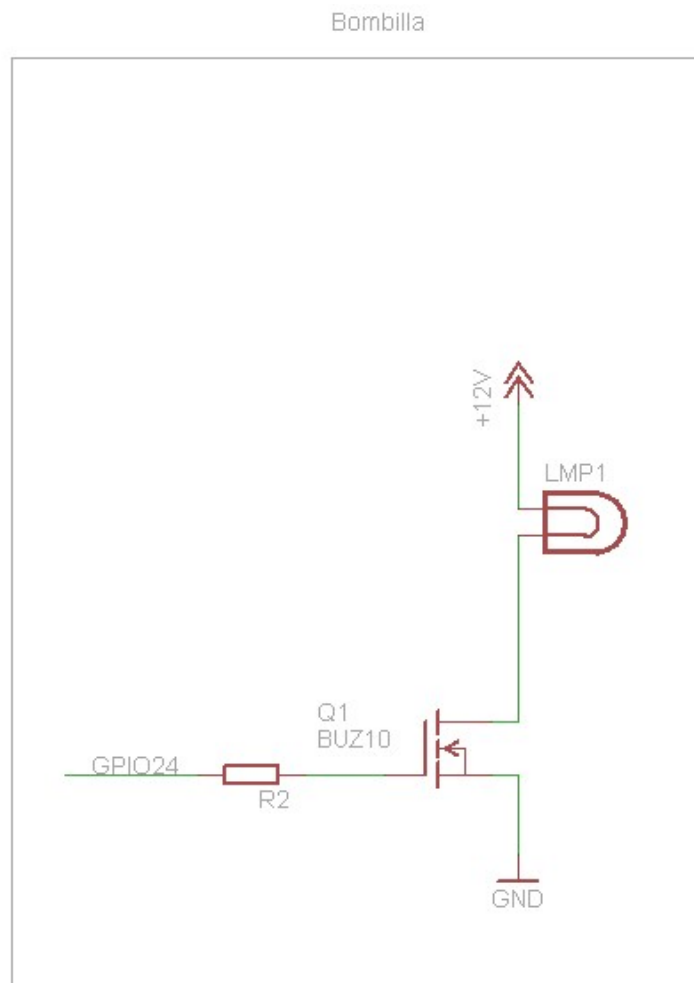
*Ilustración 31: Esquema eléctrico conexión batería*

Bat1 representa una batería real. El voltaje de entrada es de 12v que tiene que ser convertido para poder ser utilizado por la “Raspberry Pi”. En el esquema IC1 hace referencia a un convertidor smps, que tiene una eficiencia muy alta comparado con convertidores de tipo lineal. Se han añadido condensadores para mejorar la estabilidad y reducir el ruido de la fuente de alimentación.

Las C1, C2 equivalen a los condensadores de 100 nano faradios y las siglas C3 y C4 son los condensadores de 220 micro faradios.

El componente “IC1” es un convertidor MC34063 que es el que sustituirá al MC7805 lineal que se tenía previsto utilizar en un principio.

### 5.1.2.- Esquema conexión batería, bombilla y raspberry pi



*Ilustración 32: Esquema eléctrico conexión bombilla*

La alarma está representada como una bombilla que se maneja mediante un transistor de potencia de tipo MOSFET, modelo IRZ24N. Estos transistores pueden manejar grandes cargas pidiendo muy poca energía a la salida GPIO de la “Raspberry Pi”.

La LMP1 representa a la bombilla LED de 12v que se utilizará para el proyecto.

La R2 es la resistencia de 10 kilo Ohmios para que limite la corriente que le llega desde la GPIO 24 en el momento al MOSFET IRZ24N.

### 5.1.3.- Esquema conexión sensor volumétrico raspberry pi



*Ilustración 33: Esquema eléctrico conexión sensor volumétrico*

En la parte del sensor volumétrico los contactos 1 y 4 son alimentación que es a 12V y los contactos 2 y 3 son una salida a relé que mediante una resistencia de pull-up de 10 kilo ohmios representada como R1 en el gráfico, proporciona los estados lógicos 1 y 0.



## **5.2.- Instalación y configuración de software.**

Una vez esté el sistema operativo en la microsd, se procederá a hacer la configuración básica para el sistema operativo. Para ello será necesario adicionalmente un teclado por conexión USB, un cable de red RJ45 para la conexión temporal a internet y para la instalación de programas y un cable HDMI para conectarlo a una televisión con entrada HDMI o bien un cable Jack a RCA para conectarlo a la entrada RCA de la televisión.

Se inserta la microsd en la "Raspberry Pi" que tendrá conectado un teclado USB, y el cable ya sea HDMI como el RCA y el cable RJ45 con conexión a internet. Con esto estará preparada la "Raspberry Pi" para configurar el sistema.

### **5.2.1.- Instalación y configuración del Sistema Operativo.**

Al suministrarle corriente la primera vez a la "Raspberry Pi", saldrá una pantalla azul con un cuadro para seleccionar ciertas opciones:

- Se selecciona primero la primera opción "Expand Filesystem". Una vez concluido reiniciamos el sistema.
- Al reiniciar vuelve a salir la misma pantalla, se selecciona "Change User Password" y se configura.
- El siguiente paso "Enable Boot to Desktop/scratch" y se selecciona la primera opción "Console Text console...".
- Una de las más importantes es la configuración del lenguaje, ya que sin esta configuración se pueden presentar problemas a la hora de instalar aplicaciones. Es por ello que se deberá seleccionar "Internationalisation Options". Dentro de ésta, se configuran las tres opciones que se muestran. Para España hay que configurar es\_ES, es\_ES.UTF-8 y es\_ES@euro.
- Para la conexión remota se utilizará ssh y se habilitará seleccionando la opción "Advanced Options" y dentro de ésta "SSH" la cual se activará.
- Por último se pulsa "finish" y se reinicia el sistema.

Al reiniciarse el sistema se podrá observar una imagen como esta, en la que se puede apreciar la IP que se le ha asignado a la "Raspberry Pi":

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

```
dhcpcd[2033]: if_address6: Operation not supported
dhcpcd[2033]: DUID 00:01:00:01:1c:dd:60:6b:b8:27:eb:fa:5
dhcpcd[2033]: eth0: IAD eb:fa:5f:86
dhcpcd[2033]: eth0: soliciting a DHCP lease
dhcpcd[2033]: eth0: offered 192.168.0.107 from 192.168.0.1
Starting dphys-swapfile swapfile setup ...
want /var/swap=100MByte, checking existing: keeping it
done.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting Avahi mDNS/DNS-SD Daemon: avahi-daemon.
[ ok ] Starting NTP server: ntpd.
[ ok ] Starting OpenBSD Secure Shell server: sshd.
dhcpcd[2033]: eth0: leased 192.168.0.107 for 7200 seconds
dhcpcd[2033]: eth0: adding route to 192.168.0.0/24
dhcpcd[2033]: eth0: adding default route via 192.168.0.1
dhcpcd[2033]: forked to background, child pid 2386
My IP address is 192.168.0.107

Raspbian GNU/Linux 7 raspberrypi tty1
raspberrypi login: _
```

*Ilustración 34: Captura de pantalla inicio sistema “rasbian”*

Con la IP asignada y SSH habilitado se puede desconectar el cable hdmi y el teclado USB ya que por comodidad los siguientes pasos se deberían hacer desde remoto por ssh, con la instrucción:

```
ssh 192.168.0.107 -l usuario
```

donde “-l” es una “-L” minúscula y usuario es el nombre de usuario que se ha configurado en la parte de configuración. Si no se ha cambiado el nombre de usuario es “pi”.

Se actualizará el sistema con las instrucciones:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Como el archivo temporal del sistema está ubicado en el disco interno el cual en la “Raspberry Pi” es la SD o microsd según modelo, el hecho de estar grabando y borrando muchas veces en los mismos sectores las puede estropear, ya que el número máximo de veces que puede escribirse en una memoria flash antes de que empiece a fallar oscila entre 1.000 y 50.000, dependiendo del fabricante y de la capacidad de memoria. Sin ponernos en el peor caso, actualizar el SO (update/upgrade) podría superar el máximo de ciclos en unos tres a seis meses, que es un tiempo demasiado corto para el tipo de instalación que se contempla en este proyecto. Gracias a una marca muy consolidada en el mercado de las memorias, podemos ver estos datos reflejados en el documento pdf realizado por ellos “Flash Memory Guide ES” el cual lo podemos descargar de la pagina web de kingston. Para evitar el fallo en la tarjeta de memoria en la medida de lo posible tenemos dos opciones:

- Utilizar un programa, que evita que siempre se escriba en los mismos sectores, alternando todos los sectores de la sd, de manera que cuando se va a volver a escribir en un sector, se habrá escrito en todos los sectores de la sd anteriormente.

- Y la que utilizaremos que es tener como archivo temporal la memoria ram, la cual es mucho más rápida y no le afecta el hecho de escribir muchas veces en los mismos bits de memoria. El único inconveniente es que es volátil, por lo que si se corta el suministro eléctrico se pierde toda la información allí almacenada. El otro inconveniente que se podría dar es que la capacidad de memoria RAM es limitada y el archivo temporal la puede sobrepasar fácilmente. Pero no es así ya que cuando la memoria RAM se agota pasa directamente a la memoria SWAP del sistema y ésta suele ser un 20% de la capacidad del disco duro. En el caso de la “Raspberry Pi” el disco duro es la sd, bastante más grande que la memoria RAM y es gestionada directamente el sistema operativo.

Como ejemplo podemos ver el código del /etc/fstab de la “Raspberry Pi”:

```
$ sudo editor /etc/fstab
```

Código:

```
proc          /proc        proc         defaults 0      0
/dev/mmcblk0p1 /boot        vfat         defaults 0      2
/dev/mmcblk0p2 /            ext2         defaults,noatime 0      1
tmpfs         /tmp         tmpfs        size=200M,defaults,noatime,mode=1777 0      0
tmpfs         /var/log     tmpfs        size=1M,defaults,noatime,mode=0755 0      0
tmpfs         /var/cache/apt/archives/ tmpfs        size=10M,defaults,noatime 0
0
tmpfs         /var/run     tmpfs        size=100M,noatime 0      0
```

Se debe tener en cuenta que los logs se guardan en la memoria RAM y ésta es volátil, de manera que si hay algún problema y se reinicia el sistema los logs no estarán y se tendrá que comentar la línea de log para que se almacene en disco y poder ver el contenido cuando se produzca el error.

Para hacer actualizaciones con la asignación de 10 Megas para el archivo “/var/cache/apt/archives/” según la configuración anterior, no habrá suficiente, por ello en el momento de hacer actualizaciones del sistema o instalaciones de programas siempre es mejor cambiar el campo “10M” por los megas que se cree oportuno, como puede ser “30M” que son 30 Megas asignados para tal o bien comentar con el símbolo de la almohadilla “#” la línea:

```
#tmpfs /var/cache/apt/archives/ tmpfs size=10M,defaults,noatime 0 0
```

También se puede quitar el journaling que es el encargado de restaurar los archivos dañados al hacer un apagado no programado por corte de luz. Con el journaling se reduce el tiempo de recuperación del sistema. Pero para mantener la sd más tiempo se puede optar por quitarlo. Para ello se debe montar la sd en otro sistema linux y modificarlo, en /dev/sdx sustuiremos la x por la letra correspondiente a la sd.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

Se comprueba que esta activo el journaling:

```
sudo dumpe2fs /dev/sdx | grep -i journ
```

Saldrá por pantalla algo parecido a esto:

```
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype
needs_recovery extent flex_bg sparse_super large_file uninit_bg dir_nlink extra_isize
```

```
Journal inode:          8
```

```
Journal backup:         inode blocks
```

```
Journal features:       journal_incompat_revoke
```

```
Journal length:         32768
```

```
Journal sequence:       0x0024fec5
```

```
Journal start:          8359
```

lo que se tiene que tener en cuenta es lo que esta en negrita.

Ahora se puede proceder a desactivar el journaling:

```
tune2fs -O ^has_journal /dev/sdx
```

se comprueba:

```
dumpe2fs /dev/sdx | grep -i features
```

Se puede apreciar que no aparece “journal”, y con esto ya se ha terminado la configuración del sistema operativo.

Para el proyecto no se quitará el “journaling”, ya que puede haber caídas de tensión si la batería no está lo suficientemente cargada, y para una pronta recuperación y estabilidad en el sistema de archivos del conjunto es necesario mantener el “journaling”.

### 5.2.2.- Instalación de periféricos.

Para la instalación del módem 3G se debe realizar las tareas siguientes:

- Se instalan los archivos necesarios con este comando:

```
sudo apt-get install ppp usb-modeswitch wvdial
```



## Sistema de vigilancia de bajo coste energético con Raspberry Pi

`DefaultVendor=0x19d2`

`DefaultProduct=0x2000`

`TargetVendor=0x19d2`

`TargetProduct=0x2002`

`MessageContent="55534243123456780000000000000061e0000000000000  
0000000000000000"`

`MessageContent2="55534243123456790000000000000061b0000000200000  
0000000000000000"`

`MessageContent3="55534243123456702000000080000c850101011801010  
1010100000000000"`

- Se crea el archivo `wvdial.conf` para hacer la conexión:

```
sudo leafpad /etc/wvdial.conf
```

- Hay que reemplazar el contenido del archivo por este otro:

```
[Dialer 3gconnect]
```

```
Init1 = ATZ
```

```
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
```

```
Init3 = AT+CGDCONT=1,"IP","internet"
```

```
Stupid Mode = 1
```

```
Modem Type = Analog Modem
```

```
ISDN = 0
```

```
Phone = *99#
```

```
Modem = /dev/gsmmodem
```

```
Username = { }
```

```
Password = { }
```

```
Baud = 460800
```

- Añadiendo o sustituyendo las variables en **negrita** o palabras subrayadas por los datos proporcionados por el proveedor del servicio, buscar por internet la

configuración de APN del proveedor correspondiente, como puede ser vodafone, movistar, orange, etc., internet: según proveedor.

\*99#: el número de teléfono de la tarjeta sim.

El contenido de la variable Username: según el proveedor del servicio.

El contenido de la variable Password: según el proveedor del servicio.

- Se cambia la configuración para que siempre arranque el dispositivo como módem y no como disco duro.

```
sudo usb_modeswitch -c /etc/usb_modeswitch.conf
```

- Ahora ya se puede poner en marcha con la instrucción:

```
wvdial 3gconnect
```

- Para que se ponga en marcha en el arranque del sistema, editaremos el archivo /etc/rc.local al cual le añadiremos antes del "exit 0" la siguiente línea:

```
sudo wvdial 3gconnect
```

### 5.2.3.- Configuración para el servidor no-ip.

Para la instalación y configuración del servicio seguimos los siguientes pasos:

- Lo primero será crear una cuenta en [www.noip.com](http://www.noip.com). Dentro de esta cuenta se dará de alta nuestro host, al cual le asignaremos el nombre que se desee y que este libre.
- El siguiente paso es la instalación de programa para "no-ip",

- Hay que conectarse por ssh a la raspberry pi.
- Crear una carpeta con nombre noip, "mkdir noip" y entrar en ella con: cd noip.
- Descargar el programa:

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```

- Descomprimir el archivo descargado.

```
tar -zxvf noip-duc-linux.tar.gz
```

- Hay que entrar en la carpeta descomprimida: cd nombre carpeta descomprimida.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

- Ejecutar: `make & sudo make install`
- En la instalación se pregunta el nombre del usuario, contraseña y el nombre del host que se haya puesto.
- Crear el script en `/etc/init.d/noip`, para que se ponga en marcha cada vez que inicie sistema operativo, que contendrá el siguiente código:

```
#!/bin/bash

### BEGIN INT INFO

### END INIT INFO

sudo /usr/local/bin/noip
```

- Le damos permisos de ejecución: `sudo chmod +x /etc/init.d/noip`
- Configurar para que se ejecute en el arranque del sistema con la instrucción: `sudo update-rc.d noip defaults`

### 5.2.4.- Programa para la obtención de imágenes “Motion”.

Para su instalación se deberá hacer uso de la instrucción:

```
sudo apt-get install motion
```

Antes se deberá estar conectado por ssh a la raspberry.

Se modificará el archivo de configuración con:

```
sudo editor /etc/motion/motion.conf
```

Cambiando las siguientes líneas por:

```
daemon off por daemon on

webcam localhost on por webcam localhost off

control_port 8081 por control_port 8085

control_authentication pi:pruebas
```

donde “pi” es el usuario para la web de “Motion” y “pruebas” el password.

cambiar permisos pertinentes:

```
sudo chmod 664 /etc/motion.conf

sudo chmod 755 /usr/bin/motion
```



```
sudo touch /tmp/motion.log  
sudo chmod 775 /tmp/motion.log
```

Ahora ya se puede ejecutar motion con:

```
sudo motion
```

Para el proyecto se necesitará que se envíe el vídeo creado a algún lugar en la nube. Utilizaremos la variable dentro del archivo de configuración motion.conf: on\_movie\_end, en la cual se puede ejecutar un script en el momento en que termine de hacer la grabación del vídeo, y este script sería el encargado de subir el archivo a la nube a un servidor ftp o bien enviarlo por “Telegram” al usuario pertinente.

### 5.2.5.- Lenguaje de programación “Python”.

La instalación de las librerías necesarias para controlar las GPIOs desde “Python” se hará descargandolas con la instrucción:

```
wget 'http://downloads.sourceforge.net/project/raspberry-gpio-python/RPi.GPIO-0.5.11.tar.gz'
```

Se descomprime el archivo que hemos descargado, se accede a la carpeta descomprimida y se procede a la instalación.

```
tar zxvf RPi.GPIO-0.5.11.tar.gz  
cd RPi.GPIO-0.5.11/  
sudo apt-get install python-dev  
sudo python setup.py install
```

Para que detecte la señal del sensor a través de la GPIO se debe crear un archivo ejecutable para “Python” con nombre sensor.py. Hay que tener en cuenta de poner los textos sin acentos para que no de error.

```
import RPi.GPIO as GPIO  
  
import os ## librerias para hacer llamadas al sistema  
  
GPIO.setmode(GPIO.BCM)  
  
GPIO.setup(17, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)  
  
while True:
```

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

```
GPIO.wait_for_edge(217, GPIO.FALLING)

print("Boton 1 presionado")

os.system("sudo python luz.py")

GPIO.cleanup()
```

Para enviar la señal y que la luz se encienda durante un tiempo determinado el archivo ejecutable al cual se pone como nombre luz.py y el contenido seria:

```
import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(27,GPIO.OUT)##GPIO 27 se pone como salida para encender la luz.

def encender():

    iteracion = 0

    print("Se inicia la aplicacion para encender la luz")

    while iteracion < 60: ## Segundos que durara la función

        GPIO.output(27, True) ## Se enciende el 27

        print("Luz encendida")

        time.sleep(1) ## Esperamos 1 segundo

        iteracion = iteracion + 1

    GPIO.output(27, False) ## Se apaga el 27

    print("Luz apagada")

    GPIO.cleanup() ## Hago una limpieza de los GPIO

    print("Se ha terminado la aplicacion para encender la luz")

encender() ## Llamada a la función "encender"
```

## 5.2.6.- Programa de comunicación usuario-alarma “Telegram”.

Para la instalación de “Telegram” se siguen los siguientes pasos:

- Se conecta a la “Raspberry Pi” por ssh.
- Tener instaladas ciertas librerías para ello se instalan con:

```
sudo apt-get install libreadline-dev libconfig-dev libssl-dev lua5.2 liblua5.2-dev libevent-dev make lua-lgi libnotify-dev
```

- Se descarga del repositorio la ultima versión:

```
git clone --recursive https://github.com/vysheng/tg.git && cd tg
```

- Se procede a la instalación con dos pasos.

```
./configure
```

```
make
```

- Se procede a la Ejecución de la aplicación “Telegram” para su configuración.

```
bin/telegram-cli -k tg-server.pub -W
```

- Se dispondrá de un móvil con la tarjeta SIM que se va a utilizar para el módem 3G. Se conectará la “Raspberry Pi” a internet mediante el cable de red ya que el módem 3G no dispone de la SIM y no puede realizar la función de conectarnos a internet. Al introducir el comando anterior se nos preguntará el número de teléfono desde el que se va a enlazar el “Telegram” y al cual se enviará el código de activación a través de un sms, por ello lo del móvil con la tarjeta SIM. Al activar por primera vez el “Telegram” en la “Raspberry Pi” nos pide un número para el cual se introduce el número de la SIM precedido por el código telefónico de España “+34”, y acto seguido nos pedirá el código que habremos recibido por sms en el móvil. En el caso de no recibirlo nos llamarán al mismo número para indicarnos el código mediante una teleoperadora digital. Una vez se haya recibido dicho sms con el código, se introduce en la “Raspberry Pi” y se pulsa “enter”. A partir de este momento ya podemos poner la SIM en el módem 3G y quitar el cable de red.
- Ya esta instalado el “Telegram” en la “Raspberry Pi”.
- Ahora se configura el archivo test.lua para poder responder a los mensajes del usuario, respondiendo un texto o bien ejecutando alguna instrucción. En este ejemplo responde automáticamente al texto “ping” respondiendo “pong”. Hay que hacer hincapié en que se diferencian las mayúsculas y las minúsculas al introducir

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

dentro de la sección, “function on\_msg\_receive (msg)”, el siguiente texto:

```
if msg.out then
    return
end

if (msg.text=='ping') then
    send_msg (msg.from.print_name, 'pong', ok_cb, false)
end
```

- Se va rellenando esta sección, del archivo test.lua para poder tener todas las instrucciones necesarias para el proyecto. Una instrucción más compleja sería la de responder a un mensaje con una imagen tomada desde la cámara, y al tener que hacerlo a través del programa “Motion”, lo complica un poco más, quedando de esta manera:

```
if (msg.text=='Foto') then
    if (msg.to.id == our_id) then
        os.execute('wget -S --http-user=usuario --http-password=contraseña
http://localhost:8085/1/action/snapshot &')
        -- Hay que cambiar las palabras usuario y contraseña por el nombre de
        -- usuario y la contraseña que se haya puesto en el archivo de configuración,
        -- para el acceso web del control del “Motion”, al igual que configurar el puerto
        -- si así se desea al 8085
        sleep(5) -- Para que tenga tiempo la aplicación “Motion” para hacer la
        -- captura de imagen
        send_photo (msg.from.print_name, '/home/pi/motion/lastsnap.jpg', ok_cb,
        false)
        -- sustituir /home/pi/motion/ por la dirección absoluta correspondiente, donde
        -- se guardan las imágenes del programa “Motion”, la ultima imagen tomada
        -- tendrá un enlace llamado lastsnap.jpg.
        send_msg (msg.from.print_name, '¡Foto enviada!', ok_cb, false)
    else
        send_msg (msg.to.print_name, '¡Foto no enviada!', ok_cb, false)
    end
end
return
```

end

- Tambien se han añadido otras funciones extra, que son:

Foto flash: enciende la bombilla hace la foto nos la envia y apaga la bombilla.

Reboot: reinicia la "Raspberry Pi".

Apagar: apaga la "Raspberry Pi".

Miip: devuelve la ip pública de la "Raspberry Pi".

Iplocal: devuelve la ip local de la "Raspberry Pi".

Alarma on: nos devuelve un texto como que la alarma esta activada, y activa la función para cuando detecte la intrusión.

Alarma off: nos devuelve un texto como que la alarma esta apagada y desactiva la función de detección de intrusión, por lo que al detectar la intrusión no hace nada.

Cualquier otra instrucción que no este contemplada, nos devuelve un texto predeterminado, como que la prueba de conexión "Raspberry Pi" y "Telegram" funciona.

- La instrucción para poner en marcha el "Telegram" en segundo plano y que podamos enviar mensajes desde consola o desde el propio archivo test.lua es a través de web por el puerto 2233, es:

```
/home/pi/tg/bin/telegram-cli -k /home/pi/tg/tg-server.pub -s  
/home/pi/tg/test.lua -d -W -I -P 2233 -L /var/log/telegram &
```

- Para ello hace falta el archivo en /var/log/telegram y cambiar los permisos:

```
touch /var/log/telegram
```

```
sudo chown -R pi:pi /var/log/telegram
```

- Para hacer las pruebas de que funciona bien mejor quitar el "-d" y el "&" de la instrucción.
- Para parar el programa se tiene que saber el pid del daemon para después matarlo con kill en el caso de no haber quitado el "-d" y el "&":

```
ps aux | grep telegram
```

- Sale una tabla donde esta el pid y el nombre del programa en ejecución:

```
pi          2355    0.3  46.5 396064 176820 ?        S    ago08  59:24  
/home/pi/tg/bin/telegram-cli -k /home/pi/tg/tg-server.pub -s /home/pi/tg/test.lua -d W
```

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

```
-l -P 2233 -L /var/log/telegram
```

```
pi      29933  0.0  0.4  3848  1648 pts/2    S+   13:20   0:00 grep --color=auto telegram
```

- El numero subrayado es el pid que se debe matar con kill:

```
sudo kill 2355
```

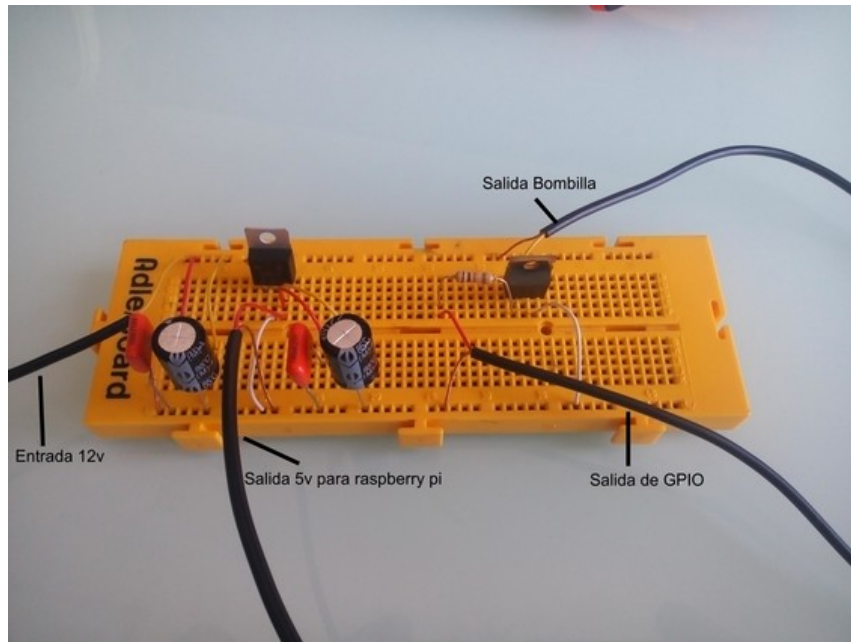
- Para que el arranque del programa se haga en el momento de iniciarse la raspberry pi se debera modificar el archivo /etc/rc.local, introduciendo la linea "/home/pi/tg/bin/telegram-cli -k /home/pi/tg/tg-server.pub -s /home/pi/tg/test.lua -d -W -l -P 2233 -L /var/log/telegram &" al final antes de la instrucción "exit 0".



Sistema de vigilancia de bajo coste energético con Raspberry Pi

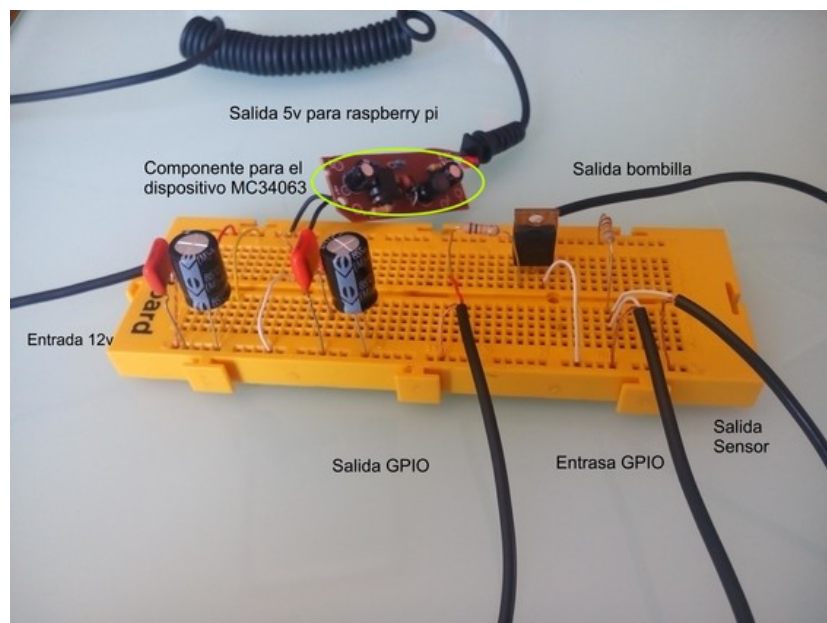
## 6.- Montaje y pruebas

Esquema montado en una placa de prototipos:



*Ilustración 37: Montaje en placa de prototipos*

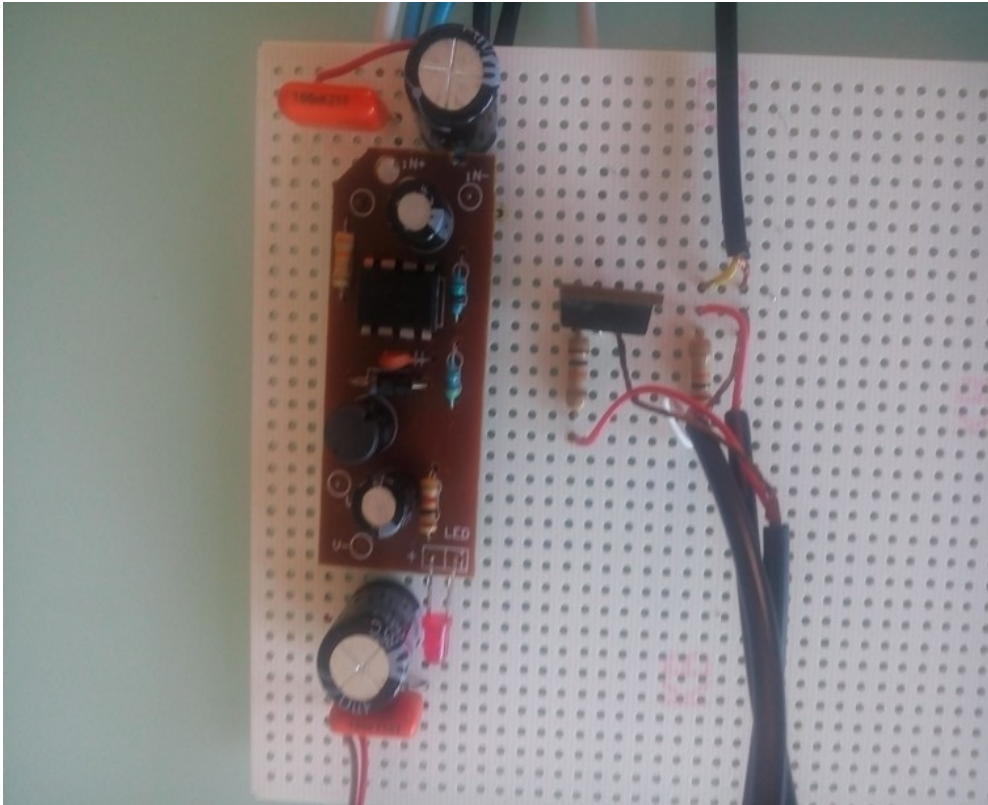
Montaje de salida de corriente de 5 voltios con el MC7805 y la salida de la GPIO al MOSFET para encender la luz de 12 voltios.



*Ilustración 38: Montaje placa prototipos completo*



Esquema montado en una placa de prototipos definitiva:



*Ilustración 39: Montaje placa prototipo soldadura*

Se ha sustituido el MC7805 por el dispositivo MC34063, cuyo cambio reduce el consumo eléctrico por pérdidas en disipación de calor.

Entrada de corriente a 12 voltios que alimenta a todo el circuito que se va a desarrollar.

Salida GPIO, para encender la bombilla activando el MOSFET.

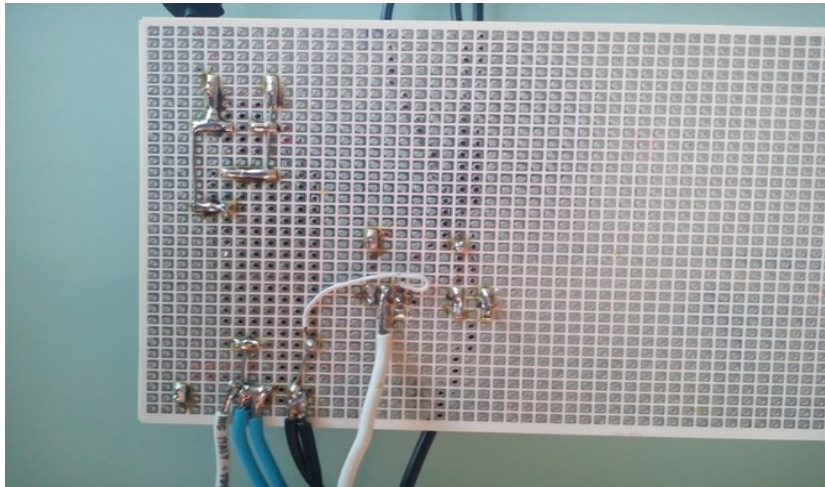
Salida a la bombilla desde el MOSFET, que es el encargado de cerrar el circuito para que se le suministren los 12 voltios necesarios para encender la bombilla.

Salida Sensor de presencia que envía la señal a la GPIO, al detectar la intrusión.

Salida GPIO es la señal que le llegara a la "Raspberry Pi" desde el sensor de presencia.

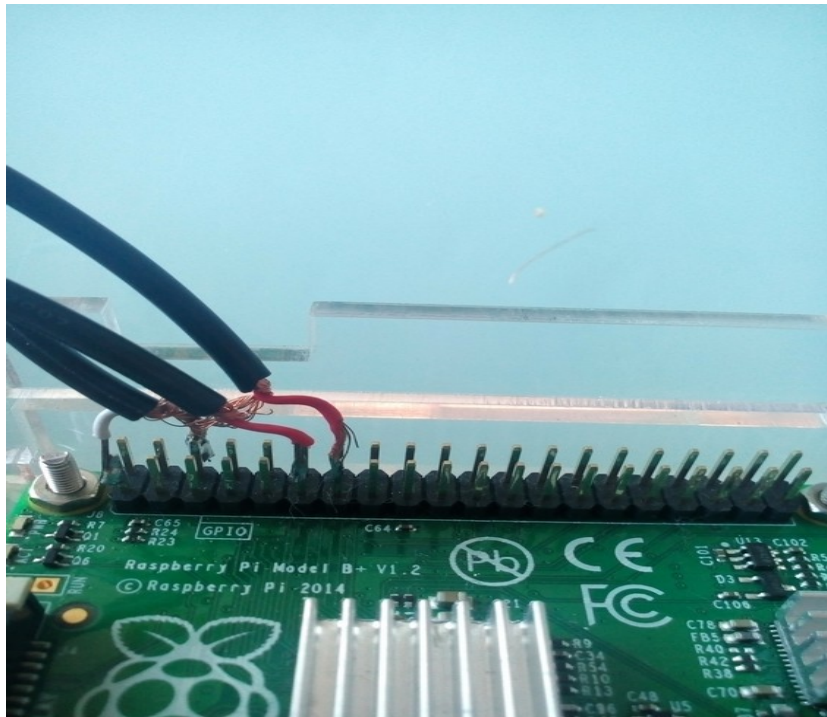
## Sistema de vigilancia de bajo coste energético con Raspberry Pi

La parte trasera donde se encuentran las soldaduras se puede ver en esta imagen:



*Ilustración 40: Montaje placa prototipo soldadura con los puntos de soldadura*

En la siguiente imagen se puede observar los puntos de soldadura con los GPIOs de la raspberry pi:

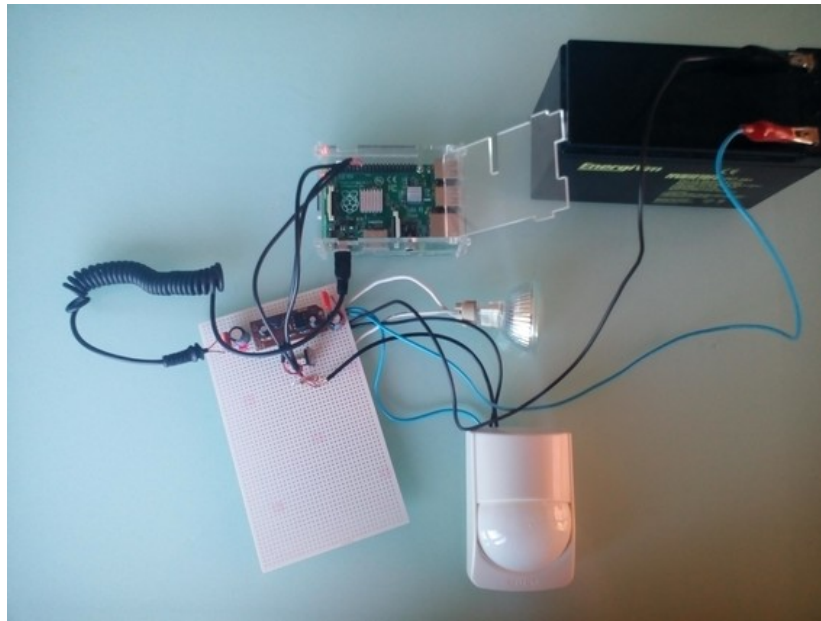


*Ilustración 41: Montaje soldadura GPIO*

Alberto Toledo Escrihuela.

Se puede observar que el pin número 6 “GND” se conecta la masa del conjunto de la conexión al sensor de presencia, junto con el pin número 1 “3v3” y el pin 11 “GPIO 17”. La otra conexión es la del relé para encender la bombilla pin 13 “GPIO 27”. La protección “tamper” del sensor se puede tratar como una conexión adicional igual a la del detector de presencia, conectándola a otra GPIO.

Sistema completo de conexiones:



*Ilustración 42: Montaje del sistema con sensor y bombilla*

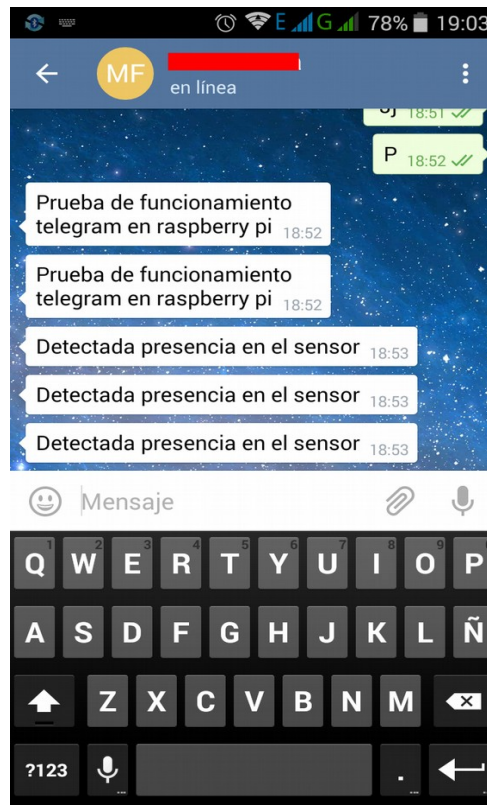
## 6.1.- Pruebas laboratorio

Una vez instalados todos los programas necesarios, se procede a su comprobación. Para ello se harán pruebas con el sistema “Raspberry Pi” y sus diferentes componentes más un móvil con la aplicación “Telegram” instalado y el “Net eye camera” que son gratuitos y los podemos descargar de la web de “Playstore”. Se puede hacer lo mismo con un equipo de sobremesa, cambiando a una página web las aplicaciones como “Telegram” o bien accediendo a la web de “Motion” de nuestra “Raspberry Pi” para ver el vídeo en “streaming”.



*Ilustración 43: Prueba conexión “Telegram”*

En esta captura de pantalla se puede observar que hay comunicación entre el dispositivo móvil y la “Raspberry Pi” con algunos comandos ya configurados como enviar “ping” y que devuelva “pung” o “PING” en mayúsculas ya que las diferencia de las minúsculas y devuelve “Mensaje enviado de (nombre del usuario que ha enviado el texto) PING”, con la primera en mayúsculas envía el texto “pong”, cualquier otro texto que no esté contemplado devuelve el texto: “Pruebas de funcionamiento telegram en “Raspberry Pi”.

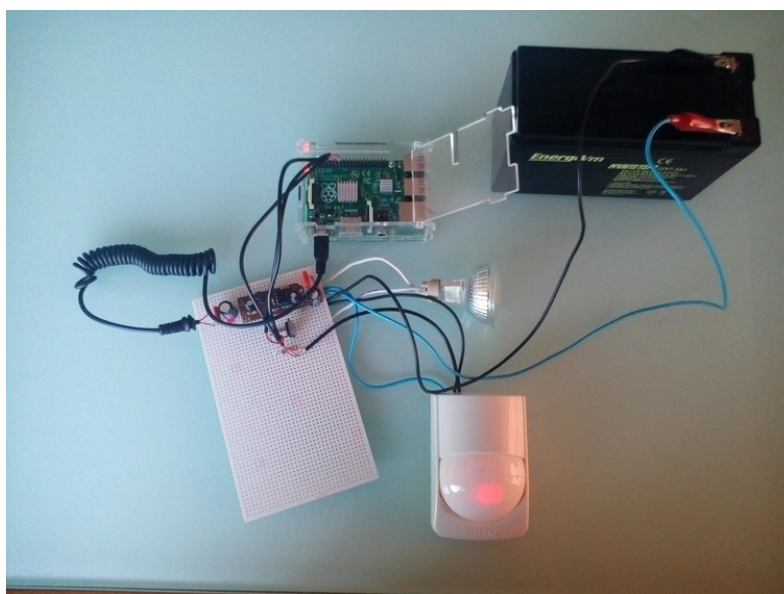


*Ilustración 44: Prueba detección presencia con “Telegram”*

En esta imagen se observa que el sistema “Raspberry Pi” con el sensor de presencia ya instalado y configurado, al detectar una intrusión ha enviado un mensaje de aviso de presencia detectada.

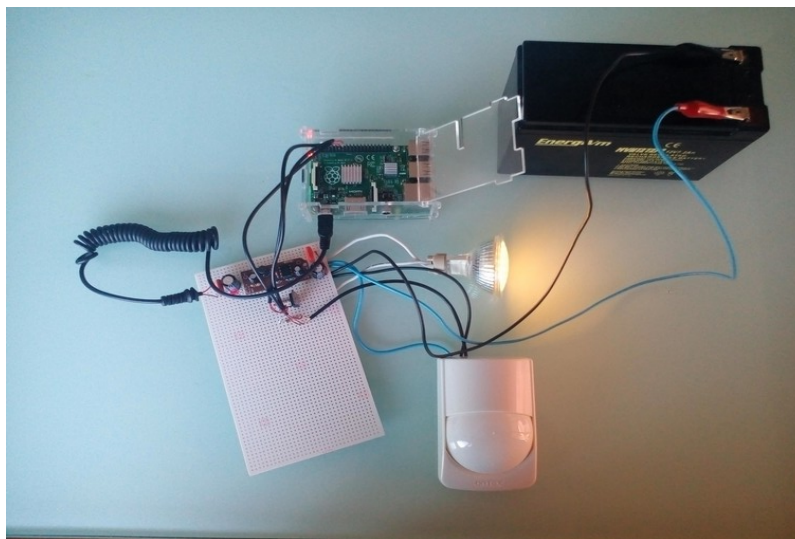
## Sistema de vigilancia de bajo coste energético con Raspberry Pi

En la siguiente imagen se puede observar el sistema “Raspberry Pi” con el sensor instalado y la bombilla.



*Ilustración 45: Montaje del sistema con sensor y bombilla, 2*

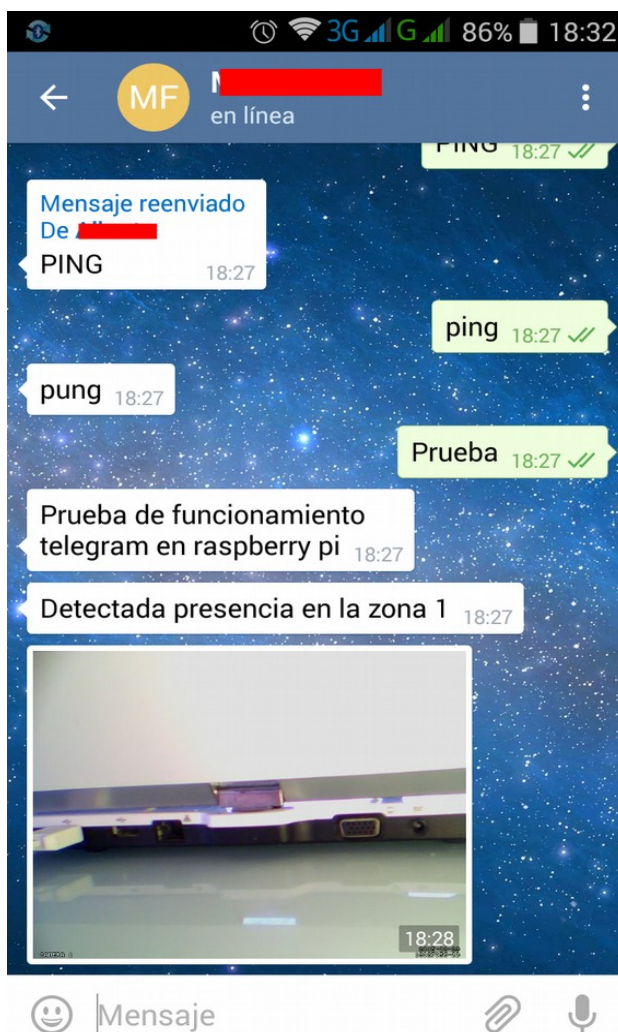
En esta otra se puede ver que el sensor ha detectado la presencia y si ha encendido la bombilla.



*Ilustración 46: Montaje del sistema con sensor y bombilla, 3*

Al añadir al sistema “Raspberry Pi” la cámara y configurar el programa “Motion”, junto con

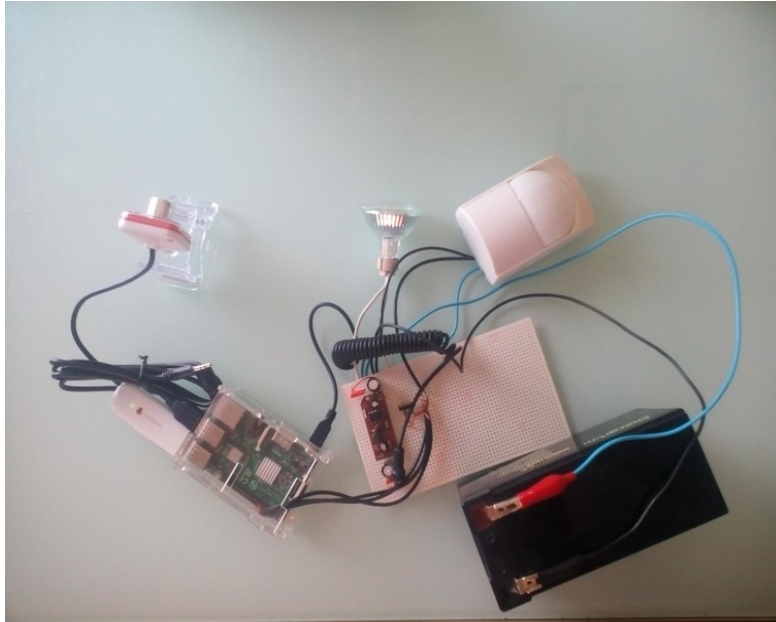
el sensor de presencia, además de enviar el aviso de texto envía una imagen capturada por dicho programa.



*Ilustración 47: Prueba detección presencia con imagen en “Telegram”*

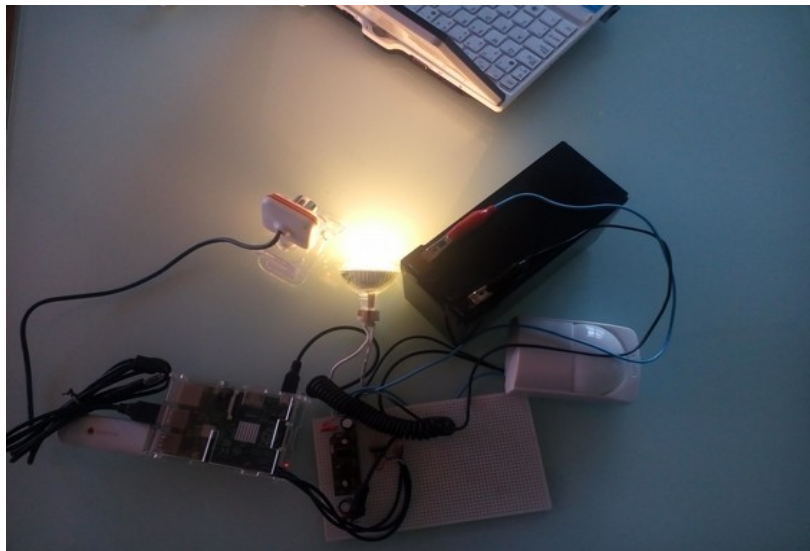
## Sistema de vigilancia de bajo coste energético con Raspberry Pi

En estas otras imágenes se aprecia todo el sistema antes de la detección de presencia y durante la ejecución al detectar la misma, en la que se enciende la bombilla y se hace la foto para ser enviada a la cuenta pertinente de “Telegram”.



*Ilustración 48: Prueba del sistema completo*

Imagen de todo el sistema encendido a la espera de detectar presencia.



*Ilustración 49: Prueba del sistema completo con bombilla encendida*

Sistema encendido con la presencia detectada y haciendo la foto para ser enviada al usuario configurado previamente de “Telegram”.



Alberto Toledo Escrihuela.

En esta otra imagen se puede observar que el usuario puede pedirle una fotografía a voluntad.



*Ilustración 50: Prueba de petición de imagen del usuario “Telegram”*

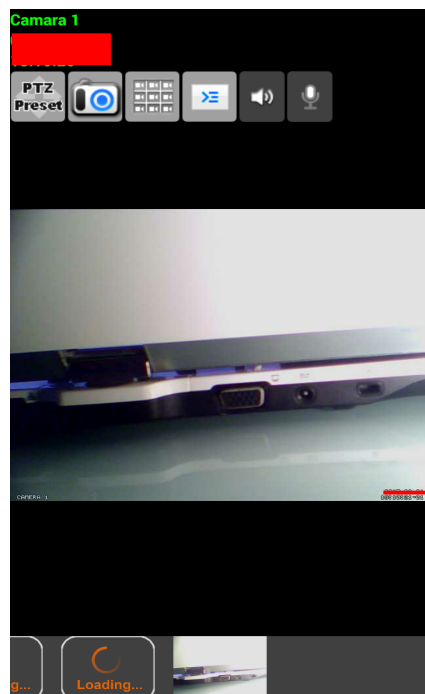
## Sistema de vigilancia de bajo coste energético con Raspberry Pi

En esta otra imagen se observa la captura de pantalla del programa “net eye camera” en la que se está viendo el vídeo en “streaming”. Una imagen es sin la luz de la bombilla y la otra con luz de la bombilla.



*Ilustración 51: Prueba vídeo en “streaming”*

Imagen sin luz de la bombilla:



*Ilustración 52: Prueba de vídeo en “streaming” 2*

Imagen con luz de la bombilla:

## 6.2.- Pruebas de campo

Una vez realizadas las pruebas de laboratorio sólo nos falta realizar una pequeña prueba en un entorno real en la que se simulará una intrusión. Para ello nos hemos desplazado a un campo con una caseta de aperos.

En dicha caseta hay instalado un sistema de control de riego por goteo y un generador de electricidad. Podemos ver las imágenes del interior de la caseta debido a que los propietarios desean la no difusión de imágenes del exterior ni la ubicación de la misma.

Montaje del sistema con el sensor de presencia y la bombilla encaradas a la entrada de la caseta, sin conectar a la batería.



*Ilustración 53: Pruebas de campo montaje imagen 1*

Montaje completo en el que se puede ver el sensor de presencia y la bombilla junto a la batería en la parte derecha de la imagen y la cámara en la parte izquierda de la misma.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi



*Ilustración 54: Pruebas de campo montaje imagen 2*

Imagen tomada desde un teléfono móvil para ver el ángulo en el que debe ir la cámara, para tomar las imágenes de la intrusión.



*Ilustración 55: Pruebas de campo montaje foto del ángulo*

Sistema completo visto desde la entrada.



*Ilustración 56: Pruebas de campo montaje foto sistema completo*

Sistema conectado a la batería y listo para las pruebas.



*Ilustración 57: Pruebas de campo montaje puesta en marcha*

Fotografía tomada desde fuera de la caseta hacia dentro cuando al sistema se le ha pedido que tomara una fotografía con la luz encendida.

## Sistema de vigilancia de bajo coste energético con Raspberry Pi



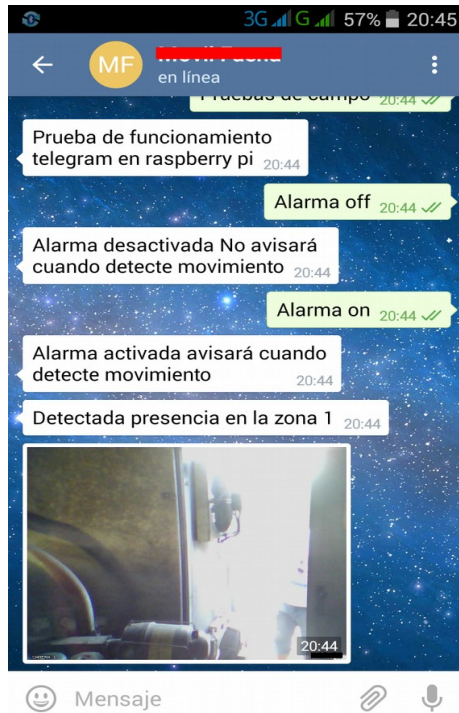
*Ilustración 58: Pruebas de campo montaje prueba funcionamiento 1*

La misma imagen sin que le pidamos nada y sin que el detector de presencia nos detecte.



*Ilustración 59: Pruebas de campo montaje prueba funcionamiento 2*

Aviso por el “Telegram” de la intrusión:



*Ilustración 60: Pruebas de campo aviso intrusión*

Captura de imagen al hacer la foto desde fuera de la caseta.



*Ilustración 61: Pruebas de campo foto por petición*

## Sistema de vigilancia de bajo coste energético con Raspberry Pi

Aviso por Telegram de una segunda intrusión al cabo de 60 segundos de la primera intrusión.



*Ilustración 62: Pruebas de campo aviso intrusión 2*

Captura de imagen de una foto con la luz de la bombilla.



*Ilustración 63: Pruebas de campo petición foto con luz*





## 7.- Mediciones consumos

Para realizar las mediciones se pueden observar que dependiendo de la tarea que realiza, el consumo puede variar. En la siguiente tabla se detallan los consumos de los diferentes dispositivos.

Dispositivo	Amperios	Voltios	Wattios
Dispositivo con MC34063	4,8mA	12v	57,6mW
Sensor al detectar	10.8mA	12v	129,6mW
Sensor en reposo	7,2mA	12v	87,12mW
Raspberry pi	0,27A	5v	1,35W
Cámara	0,043A	5v	0,215W
Módem 3G	0,199A	5v	0,995W
Bombilla	0,3A	12v	3,6W

El sistema encendido en espera de la detección de presencia, los consumos son:

0,524 Amperios, que son 524mA.

2,70472 vatios.

Con la presencia detectada pero sin haber encendido la bombilla, los consumos son:

0,5276 Amperios, que son 527,6mA.

2,7472 vatios.

Con la presencia detectada, pero el sensor de presencia ya en reposo y la bombilla encendida, los consumos son:

0,824 Amperios, que son 824mA.

6,30472 vatios.



## 8.- Conclusiones.

Este proyecto ha hecho posible la creación de un sistema viable de alarma autónomo de bajo coste energético y económico. Es controlado remotamente y nos avisa de la detección de una intrusión mediante mensajería instantánea con una imagen capturada en dicho momento. Gracias a la versatilidad de la placa "Raspberry Pi" se le ha podido conectar una cámara web, un sensor de presencia, un módem 3G, un cargador de móvil de coche, componentes electrónicos, una bombilla led de 12 voltios y una batería de 12 voltios y controlar todo el sistema desde la misma "Raspberry Pi". Todo el conjunto es económicamente asequible siendo la batería el elemento más costoso dependiendo de su capacidad.

En la realización del proyecto se han aplicado muchos de los conocimientos adquiridos durante los estudios de Ingeniería técnica informática de sistemas, pero además ha sido necesario adquirir nuevos conocimientos para poder hacer posible el propósito final. Entre las nociones previas y las adquiridas podemos destacar las siguientes:

- El hardware y conexiones con los diferentes componentes electrónicos.
- Conocimientos en sistemas operativos Linux.
- Configuración del sistema operativo de la "Raspberry Pi".
- Programación de "Python", "scripts".
- Configuración de "Telegram".
- Interrelación de los tres anteriores para llevar a cabo las tareas necesarias.

No está de más decir, la satisfacción que es para alguien del mundo de la informática, el haber construido un sistema y controlarlo desde otro dispositivo como un Smartphone, tableta o PC con acceso a internet, o que este mismo sistema pueda enviar información de un evento con imágenes en el instante en el que ocurre.

Este mismo conjunto se podría aplicar a otros proyectos con ligeras modificaciones y conseguir controlar por remoto desde nuestro Smartphone, todo un sistema de riego por goteo y saber en todo momento el estado de nuestros campos en tiempo real y desde cualquier lugar. Incluso se podría trasladar al domicilio particular realizando un sistema domótico de bajo coste.

Gracias al control que tenemos sobre la "Raspberry Pi" remotamente, se podría desarrollar un sistema que controlase remotamente todo un sistema de riego por goteo, poniendo en marcha si así se requiriera un grupo electrógeno, que alimentaría el sistema de riego por goteo y cargaría la batería de la "Raspberry Pi". Se puede llevar este método a casa pudiendo realizar un sistema completo doméstico de bajo coste.



Sistema de vigilancia de bajo coste energético con Raspberry Pi

## **9.- Agradecimientos**

Con la inestimable colaboración y apoyo del tutor del proyecto Antonio Martí Campoy y como no a la enorme comunidad de “Raspberry Pi” y todo el mundo Linux en general.



## 10.- Bibliografía

Ilustración 1:Raspberry pi Modelo B:

[www.raspberrypi.org](http://www.raspberrypi.org)

imagen símbolo eléctrico diodo:

<https://ditomasojv.wordpress.com/2012/06/25/como-probar-diodos-de-alternadores-1ra-parte/>

imagen símbolo led:

<http://www.elt-blog.com/iluminacion-led-lo-que-necesitas-saber/>

imagen de un led:

<http://www.monografias.com/trabajos60/diodo-led/diodo-led.shtml>

Imagen batería:

<http://blog.briconatur.com>

imagen GSM Mundial:

bibliografía:

<http://www.worldtimezone.com/gsm.html>

tabla raspberry pi consumos:

[www.raspberrypi.org](http://www.raspberrypi.org)

tabla raspberry pi Modelos:

[https://es.wikipedia.org/wiki/Raspberry\\_Pi](https://es.wikipedia.org/wiki/Raspberry_Pi)

Ilustración 2 GPIO's Raspberry Pi:

<http://www.raspberrypi-spy.co.uk/wp-content/uploads/2014/07/Raspberry-Pi-GPIO-Layout-Model-B-Plus-rotated-702x336.png>

Pdf de realizado por Kingston:

[http://media.kingston.com/pdfs/MKF\\_283.1\\_Flash\\_Memory\\_Guide\\_ES.pdf](http://media.kingston.com/pdfs/MKF_283.1_Flash_Memory_Guide_ES.pdf)