|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  | |  | | Imagen relacionada | |
| ALARMA MULTI-D  Alarma con sensor PIR multidispositivo escalable |
| PROJECT SPRINT #2. DATE: 18 Abril 2018  Emiliano Vázquez |

Table of Contents

[Project description 1](#_Toc513208595)

[Electronic components 2](#_Toc513208596)

[Scheme 2](#_Toc513208597)

[Extra components and 3D pieces 3](#_Toc513208598)

[Foreseen risks and contingency plan 4](#_Toc513208599)

[Create the Telegram API Chat 5](#_Toc513208600)

ALARMA MULTI-D

Alarma con sensor PIR multidispositivo escalable

# Project description

*Dispositivo controlable por móvil a través de Telegram. El código es escalable para múltiples dispositivos con un sensor, todos ellos controlados por un dispositivo base con un lector de tarjetas y un sensor.*

*Se conectan a través de red wifi, aunque no es necesario que se encuentren bajo el mismo paraguas de la misma red (diferentes ubicaciones un solo control). Dispositivos de conexión autoconfigurable (WifiManager) que crea un Access point al que podemos conectarnos con el móvil. Escanea las redes wifi y permite registrar login y pwd de una red wifi seleccionada. Tras reinicio se conecta a dicha red seleccionada.*

*Cada dispositivo puede activarse en conjunto o independientemente. Además existen dos modos de vigilancia:*

* *Modo normal: El dispositivo si detecta movimiento pasa a estado alarmado*
* *Modo vigía: Registra todos los movimientos que detecta el sensor y guarda la hora del último movimiento*

*Mediante el dispositivo base se pueden incorporar datos de configuración (Token y datos de Telegram) y hasta 6 fechas semanales de activación y desactivación automática del dispositivo (ejemplo, activación los lunes a las 8:10 de la mañana)*

*El dispositivo central controla un reloj para todos los procesos relacionados con fechas.*

*Mediante Telegram se pueden lanzar los comandos que regulan el funcionamiento de cada dispositivo, tanto la activación, como la desactivación, como la comprobación de los estados.*

# Electronic components

This is the list of the used components:

* *Arduino WebMos D1 R2*
* *Sensor PIR*
* *Dispositivo Reloj DS3231*
* *Dispositivo lectura/escritura de Micro SD*
* Cables y conexiones varias

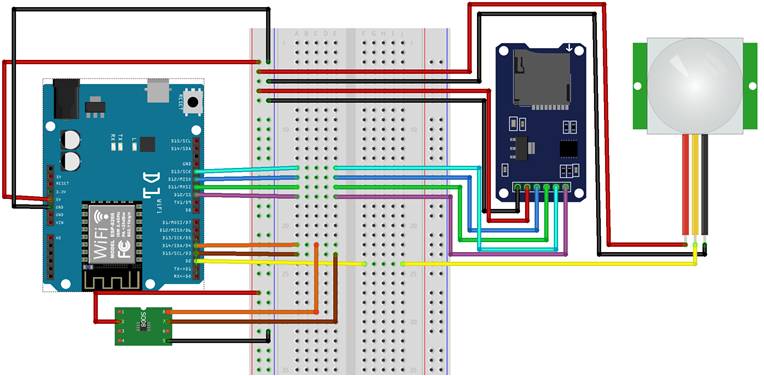
# Scheme

**Lector**

**SD**

**PIR**

**Sensor**

**

**WebMos D1 R2**

1. **-8**
2. **-7**
3. **-6**
4. **-5**

**-**

**-3**

**Reloj**

**1-2-3**

**1-2-3-4-5-6**

*PIR:*

1. *VCC (5v)*
2. *Sig – Webmos D0*
3. *GND*

*Reloj:*

*2- VCC (5V)*

1. *GND*
2. *SLC/D3 (WebMos)*
3. *SDA/D4(WebMos)*

*Lector SD*

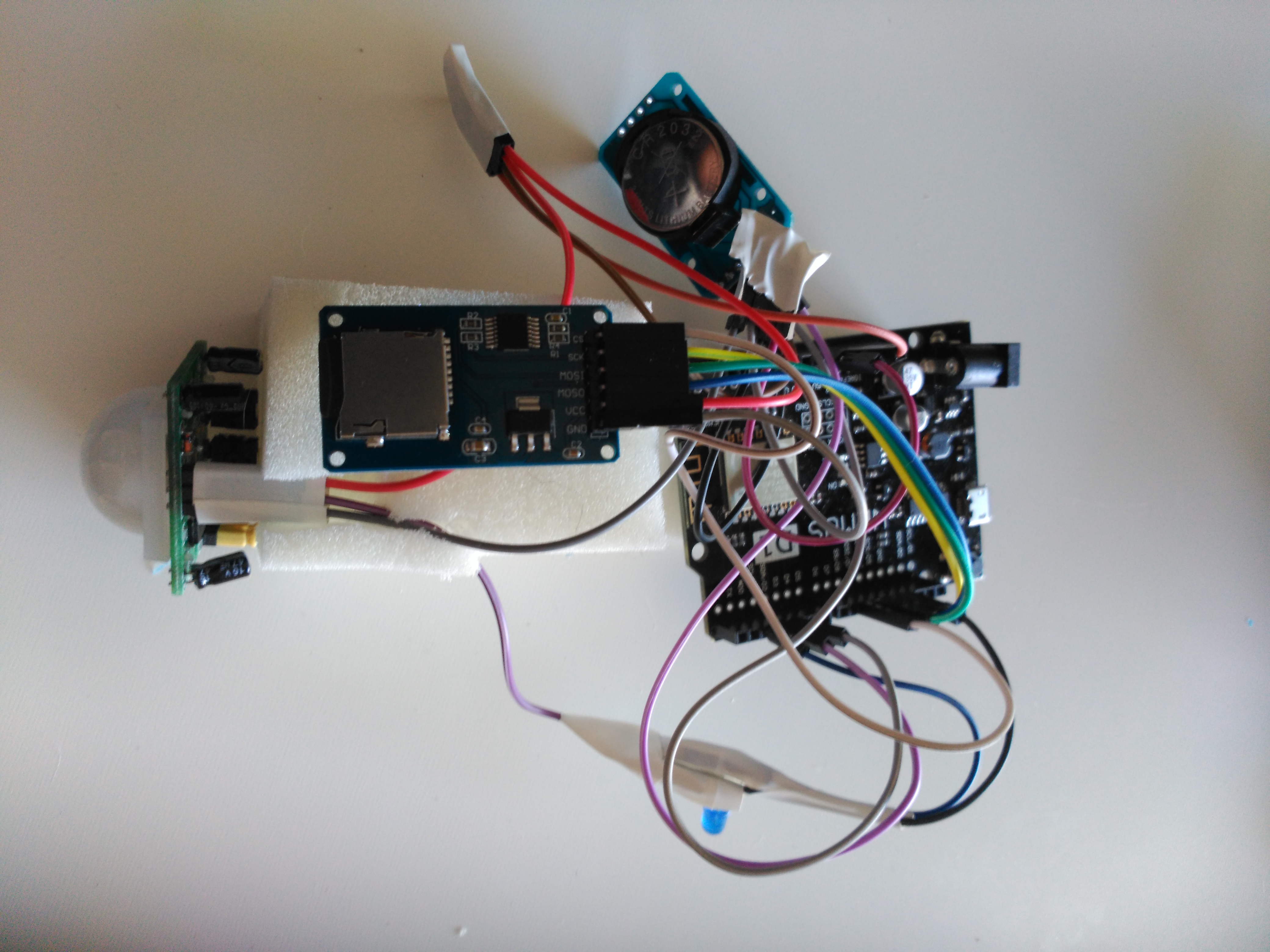
1. *GND*
2. *VCC (5V)*
3. *MISO (WebMos)*
4. *MOSI(WebMos)*
5. *SCK(WebMos)*
6. *SS(WebMos)*

# 

# Extra components and 3D pieces

* *Caja DUREX para soporte de dispositivo y sensores*

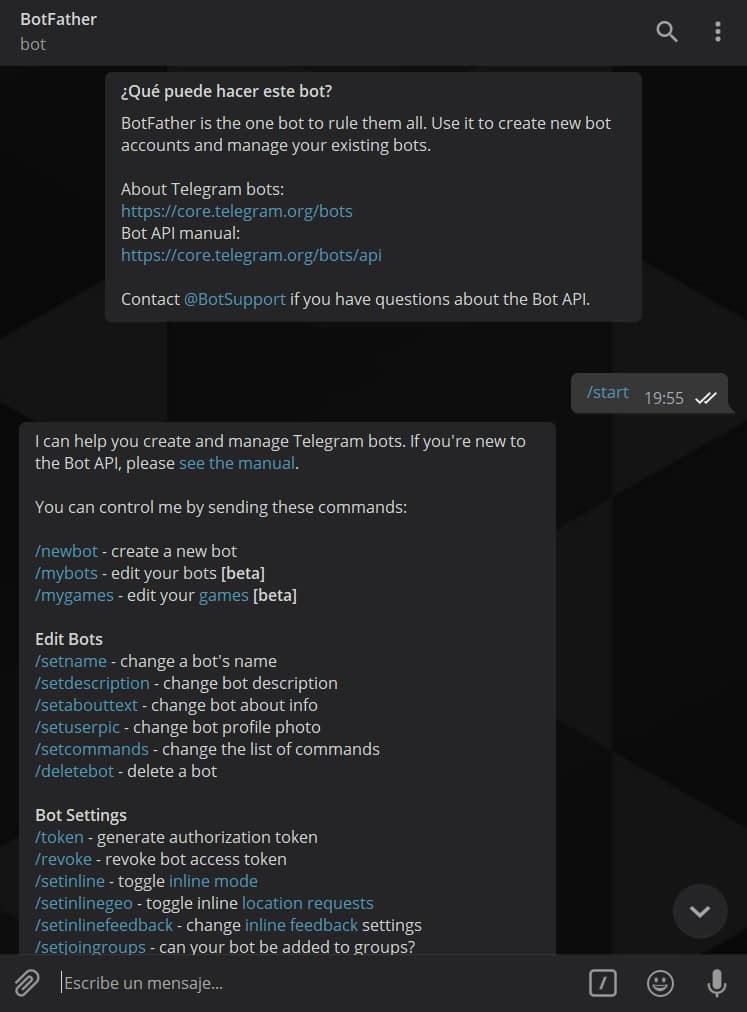
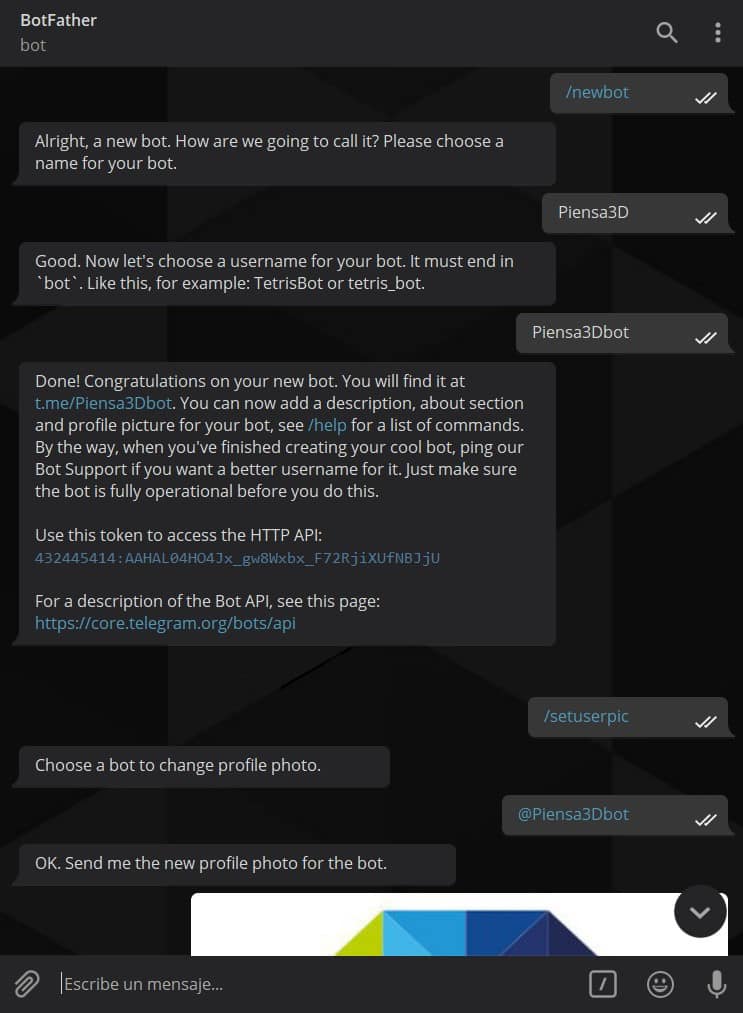
**

**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk #** | **Description** | **Probability** (High/Medium/Low) | **Impact** (High/Medium/Low**)** | **Contingency plan** |
| 1 | Energía | Low | High | SAI |
| 2 | Falta imagen | High | High | Añadir dispositivo con tratamiento de imagen |
| 3 | Interferencias IP | Low | High | Incorporar dispositivo Ethernet |
| 4 | Chequeo de dispositivo Vivo | Low | Low | Incorporar chequeo UDP desde el Base incompatible con librerías actuales. |

# Foreseen risks and contingency plan

# Create the Telegram API Chat

* Lo primero que hay que hacer es **instalar la aplicación de Telegram** en el Movil.
* Luego debemos dirigirnos a la **plataforma de Telegram** y buscar **@BotFather** para poder crear nuestro bot. Es aquí cuando le daremos a **iniciar** y nos saldrán una lista de **varios comandos** para poder personalizarlo a nuestro gusto, pero lo que nos interesa es darle a **/NewBot** para crearlo.  Nos debería aparecer una pantalla como esta:
* 
* Y simplemente debemos **seguir los pasos** que nos indica, como podemos ver en la imagen siguiente:
* 
* Una vez tenemos esto, ya podemos seguir con la parte importante: **La programación del bot**. Pero antes hay que decir que, con los comandos de **Edit Bots** y **Bot Settings**, podemos personalizar nuestro bot como queramos: la foto, descripción, privacidad en los grupos, si podemos añadirlo a grupos e infinidad de cosas, que más adelante iremos explicando con calma.
* El siguiente paso es preparar la zona de creación del bot. Este paso es lo más importante, ya que sin él, no tendríamos donde **almacenar los datos** del mismo. En cambio, dejaremos un poco a la elección de cada uno como hacerlo, ya que hay varias opciones disponibles.

En nuestro caso pondremos los siguientes datos:

* Nombre del bot: **Alarma\_CVC**
* Usuario del Bot: **alarmacvcbot** (el usuario quedara registrado como @alarmacvcbot)
* Por ultimo nos devolverá el valor del Token de la API donde pone “use this Token to access the **HTTP APÎ**: (el valor debe ser algo parecido a esto:

xxxxxxxxx:xxxxxxxxxxxxxxxxxxxxxxxx

introducimos este valor en la tarjeta SD

References

This project has been inspired by the following Internet projects:

<https://www.prometec.net/control-acceso-clave/>

<https://tzapu.com/esp8266-wifi-connection-manager-library-arduino-ide/>