secuCam

User Manual

Version 1.1.x

Table of Contents

Introduction

This is a best effort to implement a home made security camera for your private use and without any commercial purpose.

- Server and Client solution to build your security camera using a Raspberry Pi and a few hardware accessories.

- The server side strives to be as simple as possible to be installed and configured in your Raspberry Pi.

- Activate or deactivate the camera remotely, allowing to disarm the system temporary when you are not at home.

- Setup and change your notification parameters from your mobile application.

- All communication between your mobile application and the camera is encrypted with your own password. Therefore, no one would be able to tap your system.

- Unlimited mobile devices can be configured to access your security camera.

- All instructions to your camera are registered and they can be accessed at any time from your mobile application.

- The camera is activated when motion is detected and the event is videotaped and recorded. Each event will finish after 60 seconds of inactivity.

  In case there is a subsequent movement before the 60 seconds of inactivity, it will be recorded in the same video file as part of the initial event.

  On the other hand, if no further movement is detected after 60 second of inactivity, the event will be closed.

  Afterwards, another event can be triggered and it will generate a new event and a new video file will be recorded.

- Each event will generate an email notification with the date and time when it happened.

- The maximum length for a video file is set to 30 minutes. In case an event last longer, the video associated to this event will be split in one or more video files.

- Secure access to the videos automatically recorded in each event from any browser, FTP client or media player.

- Secure access to live stream from any browser or media player.

- You can attach servos to the GPIOs of the the Raspberry Pi to move the camera remotely. You have 2 sliders in the mobile application to control vertical tilt and horizontal pan.

Legal note: Raspberry Pi is a trademark of the Raspberry Pi Foundation. All information in relation to Rapberry Pi and Raspbian can be found in the official site: www.raspberrypi.org.

Building blocks and terminology

Before to start with the explanation of how to install and configure the various pieces of hardware and software, I will explain some terminology to be followed along this manual.

There are 4 different players in this solution with a clear and defined role:

Security camera: Hardware and software which will implement your camera.

Mobile application: Software to communicate with your security camera and access all features and settings.

Router: This is your own router which will need to be configured to access your private network from the outside world.

Dynamic DNS: Third party service where you will get a unique host name instead of dealing with your external IP address. It will be required if you don't have a static IP assigned to your router.

I will refer to the security camera as a whole most of the time, but there might be some instances when I need to indicate the specific service or application running in your Raspberry Pi to make the explanation more accurate.

Each time a a service or application is mentioned, it will be highlighted in ***italic bold***.

All command line instructions, references to files and directories will be in `mono blue`.

Setting up the security camera

As explained before, I will call "security camera" to the hardware and software that make possible the implementation of this device.

The security camera is composed by the following:

- Raspberry Pi (Raspberry Pi 2 is tested, but it might work in previous models)
- Raspberry Pi camera module
- Servos (optional)
- secuCam-server
- FTP server utility
- Motion utility

The rest of this chapter will explain how to install and configure each of the building blocks.

Prepare your Raspberry Pi

You will need to install Raspbian in your Raspberry Pi as per instructions in the official site: www.raspberrypi.org/downloads.

This installation is very well documented above, so I will skip this section.

Once the installation is completed, it is recommended to update and upgrade your packages to the latest version:

```
~$ sudo apt-get update
~$ sudo apt-get upgrade
```

Now that all packages are upgraded, you can also update your kernel as follow:

```
~$ sudo rpi-update
```

Finally, you will need to setup Raspbian to customize the OS for the purpose of this project:

```
~$ sudo raspi-config
```

It will open the configuration window and the recommended settings are detailed below:

```
Expands files [Make sure your OS is using the total SD card space]
Change user password [Recommended for security]
Enable camera [Necessary to use the camera module]
Disable boot desktop [Desktop is not required in this project and it will free resources]
Internationalization [Set your locale including time zone]
Memory Split [You won't need to run a desktop, so keep GPU to the minimum 16]
```

In case you have problems with the package source, you might need to change the mirror in your /etc/apt/sources.list as described in Raspbian's site:

www.raspbian.org/RaspbianRepository

Install and configure secuCam-server

Once Raspbian OS is loaded in your Raspberry Pi, you can startup the system.

Download **secuCam-server** (scserver-XXX_arm.tar.gz) from our site (www.darhon.com/secucam) and copy to your Raspberry Pi's home directory.

Execute the following command in the terminal to extract the files:

```
~$ tar -xf scserver-XXX_arm.tar.gz
```

As a result, you will have a directory sc in your home [/home/pi/sc] containing all binaries and configuration files.

The assumption from now is that you have installed **secuCam-server** as per above directory tree.

Finally, you will need to change some parameters from motiona.conf and motionp.conf inside [/home/pi/sc/config]

```
~/sc/config$ nano motiona.conf
~/sc/config$ nano motionp.conf
```

The following line will set the username and password to access the live stream when the camera is ON.

```
stream_authentication username:password
```

The following line will set the username and password to access the configuration of **motion** remotely.

```
webcontrol_authentication username:password
```

The configuration file motiona.conf is loaded when the camera is working in active mode and motionp.conf is loaded when the camera is working in passive mode. Therefore, to change

any of the configuration parameters for **motion**, you will need to make sure to update both files. The rest of the parameters have been preset for you and usually there is no need to change them, but if you feel to customize them further, you can get some guidance in the chapter "*Install and configure Motion*".

All executables will and should be run as normal user without root privileges, except **scgpio**. This executable will command the GPIOs of your Raspberry Pi to control the servos and enable the Tilt and Pan capabilities. However, the only way to accomplish this task is running with root privileges to access the hardware.

The safest way to go along this restriction is having an independent and standalone executable (*scgpio*) running as root with only one role: commanding the GPIOs, while the rest of the functionality is implemented in the main application.

Now in order to increase the privileges of this executable, you will need to run the following instructions from [/home/pi/sc]:

```
sudo chown root scgpio
sudo chmod 4755 scgpio
```

Install and configure the FTP server

You will need a running **FTP server** to access the video events generated when motion is detected.

Execute the following command in the terminal to install the utility:

```
~$ sudo apt-get install vsftpd
```

Now, you will need to customize the configuration file to match your requirements.

Open the file from the terminal:

```
~$ sudo nano /etc/vsftpd.conf
```

Customize the parameters based on the guidelines below:

Listen connections.

```
listen=YES
``` (required)

Accept anonymous connection. It means that anyone who knows your host (IP) address will be able to access your **FTP server** without a password.

```
anonymous_enable=NO
``` (default is YES, but it is not recommended)

The root directory for an anonymous connection. In other words, a user connecting to your server anonymously will land in this directory and he won't have access to directories above.

```
anon_root=/home/pi/sc/event-videos
```

Accept connections for users registered in your Raspbian OS. The default user is **pi**.

```
local_enable=YES
``` (recommended, but make sure to change the default password)

This is the root directory when you connect using your **pi** username and password.

```
local_root=sc/event-videos
```

Allow writing privileges for all FTP actions. It will allow you to upload files if required.

```
write_enable=YES
```

Present the time of files based on the server's current time. Useful when you are in a different time zone.

 `use_localtime=YES`

Accept connections from usual port.

 `connect_from_port_20=YES`

After the above modifications are completed you will need to re-start the **FTP server**.

 `~$ sudo service vsftpd restart`

Install and configure Motion

Motion is used by **secuCam-server** in the background to interact with the camera. **Motion** will generate events and action other parts of the server when it detects movements.

Execute the following command in the terminal to install the utility:

 `~$ sudo apt-get install motion libjpeg62`

This command will install all necessary dependencies for **motion** to work. However the current version of **motion** in the repository doesn't support the new Raspberry Pi camera module. As a result, I have included a modified version as part of the package `motion-mmal` located in the `[/home/pi/sc]` directory.

Motion is completely controlled by a configuration file. As explained before, the configuration file `motiona.conf` is loaded when the camera is working in active mode and `motionp.conf` is loaded when the camera is working in passive mode. Therefore, to change any of the configuration parameters for **motion**, you will need to make sure to update both files. Additional information to configure **motion** is found in the developer's website:

 http://www.lavrsen.dk/foswiki/bin/view/Motion/WebHome

Note however that by the time this manual was written, the guide available in this website relates to an older version of **motion** and as a result, some of the parameters are named differently. But the essence is the same.

Start secuCam-server during startup

The next step is to make sure **secuCam-server** will start when your Raspberry Pi is rebooted. Therefore, you won't have to do anything when power is gone and the system reboots.

Execute the following command in the terminal to edit `rc.local` configuration file:

 `~$ sudo nano /etc/rc.local`

Add the following line before the `exit` command as below:

 `sudo -u pi /home/pi/sc/scserver.sh &`
 `exit 0`

The ampersand at the end is mandatory to keep this command in a different process and the OS continue booting.

Advanced parameters

When **secuCam-server** is called without arguments, it will use the default configuration to set all parameters for you. However, if you need to change any of the default values, you can apply the following arguments after `scserver.sh` :

```
-h  Show this help.
-v  Version of secuCam server.
-p  IP port where the server will listen.
-i  Time interval refresh for your dynamic DNS (in miliseconds).
-s  Maximum size of disk space to store video events (in bytes).
-w  Enable servo capability.
-r  Range of servo's pulse width. Max Width - Min Width (in microseconds).
-d  Debug mode to get additional info in the terminal.
```

Example:

```
scserver.sh -w -p 8000 -i 600000 &
```

The above example will enable the GPIOs to connect servos, start the server to listen in port 8000 and it will refresh the dynamic DNS every 10 minutes.

Mobile application

secuCam is available for Android in the Play store. You can download the application and install it for free.

The first time you start the application, it will request for you to create a password.

Initial screen

The initial screen will allow you to access all the functionality in this application.

The first time you start, the *Communication Parameters* section will be opened to fill out the required values to communicate with the security camera (**secuCam-server**).

The next time you reach this screen, this section might be closed. But it can be opened with *Param* button.

URL: This is the IP address (eg. 192.168.0.2) or host address for your security camera. The first time you access, you will use the local IP assigned to the Raspberry Pi device by your router.

Later you can change and use the public IP address assigned to your router by your Internet provider and configure your router to forward port requests. Finally, if you don't have a static IP, you can use a domain registered in a Dynamic DNS service. These options are discussed with more detail later.

The *Default* button will retrieve the URL previously saved in *Settings*.

Communication password: The communication between **secuCam-server** and **secuCam-client** is encrypted. This feature gives you two main advantages:

1. Ensure the security and privacy of your communication. It makes virtually impossible for

malicious users to hack your system and disarm the camera. All communication not recognized by the server will be discarded and the connection aborted immediately.

2. You can have more than one device to control your security camera. Just make sure that all devices have the same communication password. And you can change the password at any time from your mobile. And of course you will need to advise the rest of the users about the change.

The default communication password is: *password*

You need to change it and create your own as soon as it is possible.

Not confuse the communication password and the password to open the application. The former is unique for all the mobile devices and your security camera, while the latter is specific for your mobile and each mobile can have a different one.

The *Default* button will retrieve the password previously saved in *Settings*.

Port: This is the port where **secuCam-server** is listening. The default number is 50000, unless you change this parameter with the -p argument.

Control Panel

This screen allows you to control the Tilt/Pan and turn the camera ON and OFF. There are two modes of operation:

Passive: The camera will turn on, but the motion detection will be disabled. You will be able to watch live video stream only.

Active: The camera will turn on and the motion detection is activated. The system is armed and ready to detect and advise about any movement in its view range.

In addition, you can also access the live video stream at any time. This is the mode to be used when you are away from home and intend to use your security camera to watch for you.

When the mode is active, your security camera will send an email notification with the date and time of the event. A video will be recorded every time an event is triggered. This video will last as long as the event last. Each event will finish only after 60 seconds of inactivity, otherwise the camera will keep videotaping and the video file updated.

Note that each video file has a maximum length of 30 min. Therefore, longer events will create more than one video file.

There is one slider to control the vertical axis (Tilt) and another for the horizontal axis (Pan). The range of both axis is 180°, moving from -90° to +90°. This is the standard range for servos.

Settings

All parameters for secuCam are monitored and updated in this screen.

Communication parameters

Username: Local parameter stored in your device and particular to your mobile phone. It will help to identify all requests coming from this phone.

Password: Communication password to access **secuCam-server**. Once this is changed, you will need to update all mobile phones linked to your security camera.

URL: Local parameter stored in your device. This is the IP address or host name for your security camera as already explained in the *Initial screen*.

DDNS: Remote parameter stored in the security camera. This is the URL required to update your Dynamic DNS provider with your current public IP.

Notification parameters

All notification parameters are remote and stored in the security camera. These parameters are used by **secuCam-server** to send email notifications when an event is triggered.

Email: Your email address where you want to be notified.

Password: Your email password.

Server: The SMTP server for your email provider.

Port: The port number for your SMTP server.

Connection: The type of connection you will use, which is related to the port number.

Authorization: The type of authorization you will use.

Once you are happy with the changes, click on *Apply* to save the local parameters and send the remote parameters to the security camera.

History

This is a read only screen to access and review all requests sent to your security camera.

The table will show the date and time of each event, the user who did the request and the description of the request.

Here is when it makes sense to assign a useful username to each of the mobile devices linked to your security camera.

Possible actions:

Setup update: Registered every time you change the setup parameters.

Camera ON [active]: Registered when the camera is turned ON in active mode.

Camera ON [passive]: Registered when the camera is turned ON in passive mode.

Camera OFF: Registered when the camera is turned OFF.

Change Tilt xx: Registered when the camera's tilt is changed.

Change Pan xx: Registered when the camera's pan is changed.

Security

The application requires a password to enter. It will protect your system against unwanted access.

Note that you log out by simply going back to the initial screen.

Setting and changing a Password: Click on *Pass* button in the Initial screen to open the Password screen and change your password.

You will find the necessary instructions on top of the screen to guide you through basic steps to

set or change your password.

For your own peace of mind, the application will log out automatically after 10 minutes of inactivity.

Access to video files

You can access the video files recorded in each event thanks to the **FTP server**.

You will need a FTP client for your mobile or laptop device. The parameters to set it up are explained below:

URL: IP address or host name for your security camera

User: pi (unless you change the user in your Raspbian OS)

Password: The password for your Raspbian OS.

Access to live stream

You can access live stream when your camera is ON from your mobile or laptop. The easiest way is using a browser compatible with Multipart JPEG and go to the following link:

   http://username:password@host:8081

host: IP address or host name for your security camera

port: 8081

user:password: username and password as per configuration files: motiona.conf and motionp.conf. The following line in those files will determine the current configuration:

   stream_authentication username:password

Another option to watch live stream would be using VLC and setting the parameters as above.

Access to control motion

There are two configuration files for **motion**: motiona.conf and motionp.conf

The first one is used when the camera is turned on in active mode and the second for passive mode.

You can access the configuration files from your browser at any time using the following link:

   http://username:password@host:8080

host: IP address for your security camera

port: 8080

user:password: username and password as per configuration files: motiona.conf and motionp.conf. The following line in those files will determine the current configuration:

```
webcontrol_authentication username:password
```
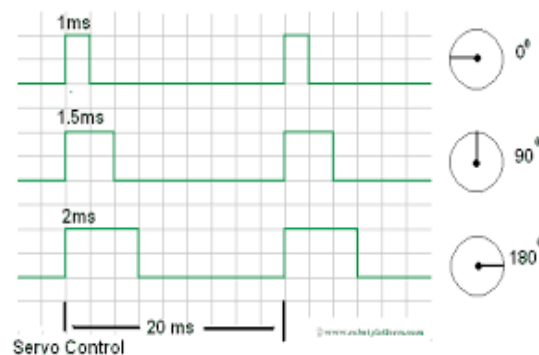
Install and configure servos

The security camera and the mobile application comes prepared to incorporate Tilt and Pan capabilities. You will just need to connect the servos and enable **secuCam-server**.

The default parameters should work without change in most of the cases.

How do servos work

The angle of a servo is determined by the width of an electrical pulse. The servo expects to see a pulse every 20ms. The width of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position (neutral position).

Similarly, 2.0ms pulse will make a clock-turn to 180°and 1ms pulse will make an anti clock-turn to 0°.

For convenience the above convention is translated in the applications as follows:

```
-90° (1.0ms)   — Full anti clock-turn
  0° (1.5ms)   — Neutral position
+90° (2.0ms)   — Full clock-turn
```

The servos should be powered from an independent +5v regulated source. It is highly recommended not to use the inbuilt +5v pins in your Raspberry Pi, because servos can draw peaks of current when operating in certain conditions like being forced and it can damage your device.

Setup the software

SecuCam-client is already enabled for this capability.

SecuCam-server needs to be enabled and configured at start up using two arguments.

> -w It will activate this capability
> -r Range of servo's pulse width

The range of the pulse width [maximum width – minimum width] is commonly 1ms. The neutral position is established by a 1.5ms pulse. A narrower pulse will turn the servo to the left and a wider pulse to the right.

Standard servos will turn to -90° with 1ms pulse and +90° with 2ms pulse. This is the default configuration of 1ms range.

Some other examples below:

-r 1200 [0.9ms to 2.1ms]
-r 800   [1.1ms to 1.9ms]

Be aware that going over the limits can damage the servo. Make sure to read the specifications before changing the parameters.

Setup the hardware

The block below (elinux.org) shows the RPi GPIO configuration for GPIO 40-pin header. Take in consideration that all GPIOs provide 3.3V.

```
            _____
            +3V3   1   2    +5V
GPIO2   SDA1   3   4    +5V
GPIO3   SCL1   5   6    GND
GPIO4   GCLK   7   8    TXD0   GPIO14
            GND   9   10   RXD0   GPIO15
GPIO17  GEN0  11  12   GEN1   GPIO18
GPIO27  GEN2  13  14   GND
GPIO22  GEN3  15  16   GEN4   GPIO23
            +3V3  17  18   GEN5   GPIO24
GPIO10  MOSI  19  20   GND
GPIO9   MISO  21  22   GEN6   GPIO25
GPIO11  SCLK  23  24   CE0_N  GPIO8
            GND  25  26   CE1_N  GPIO7
EEPROM  ID_SD 27  28   ID_SC  EEPROM
GPIO5         29  30   GND
GPIO6         31  32          GPIO12
GPIO13        33  34   GND
GPIO19        35  36          GPIO16
GPIO26        37  38          GPIO20
            GND  39  40          GPIO21
```

Connect the servos as follows:

- Tilt control: PIN16 / GPIO23

- Pan control: PIN18 / GPIO24

- The power to the external source or PINs 2 and 4.

- Ground to any of the GND PINs in the board.

When the source to power the servos is other than the Raspberry Pi, you might need to connect the grounding point of the external source to any of the GND PINs in the board. It will ensure all voltages refer to only one common ground.

Access from outside your private network

Your security camera will be living inside your private network and it will only be visible for you when you are connected to this network.

In order to access your private IP from outside your network, you will need to set the URL pointing to your public IP. This is the unique IP assigned to your router from your Internet provider.

But this is not all, your services are running in one of the internal private IPs, so the requests to your router will need to be forwarded to the correct internal IP based on the port.

The following example shows a scenario to explain the procedure:

Public IP: 114.78.50.50

Internal IP for your Raspberry Pi: 192.168.0.2

Setting your router to Port Forwarding:

Service	External Port	External IP	Internal Port	Internal IP
SecuCam	50000	Any	50000	192.168.0.2
FTP	20-21	Any	20-21	192.168.0.2
Live stream	8081	Any	8081	192.168.0.2

Consequently, you will need to set the URL for your mobile application, FTP client and the browser as follows:

URL: 114.78.50.50

Port: 50000

It will allow you to access your security camera from everywhere in the world.

Dynamic DNS

The solution in the previous section will work well as far as you have an static IP in your router. However, this is not common and it might comes with an additionally fee.

The usual situation is that your public IP will change many times per day, or a few times per week or when your router is rebooted. Anyhow, you will need to keep track of it.

The best way to solve this problem is registering to a Dynamic DNS service that will update your public IP and you will be given a host name which is going to be unique. The DDNS service will take care of tracking your public IP.

There are many DDNS services, some of them charge a fee and some are free. For the purpose of this manual, I will use DHIS.org to create an scenario and explain the configuration parameters to setup your application.

All parameters in relation to your DDNS are configured in your mobile application under

Settings and they will be sent to **secuCam-server** when you *Apply* changes.

Let's create an scenario to understand each parameter:

DHIS.org host: YourHost.dhis.org

Host ID: 45129

Password: password

The service provider (DHIS.org) will update your IP address every time your security camera sends the following request:

```
http://HostID:password@is.dhis.org
```

Your IP address is taken from the request's source arriving at the server. As a consequence, your DHIS host (YourHost.dhis.org) is brought online and it will remain online for 6 minutes only. After this period has elapsed if there haven't been any more recent updates the host is brought offline.

In order to keep your host online, your security camera needs to make an HTTP request every so often in order to refresh the online timer.

So, as explained before, **secuCam-server** will update your DDNS once every 5 minutes as default, which works well for this service in particular. If you need to change the refresh interval, you will need to instruct the server using the `-i` argument.

Finally, for the current scenario, you will need to configure your mobile application as follows:

DDNS: `http://45129:password@is.dhis.org`

Host: `YoutHost.dhis.org`

Click *Apply* and you are all done.

Mounting the camera

After all software configuration is in place, you will need to work your hardware and mount the camera to the destination point.

Some helpful hints:

- Be sure to place the power supply in a dry and safe place. Raspberry Pi 2 requires 5V and a minimum of 1.2A.

- There are many servos in the market, but you don't need any fancy unit for this simple application, so you could use the low cost 9g Micro Servo A0090 which was successfully tested. It is available in SparkFun (ROB-09065).

- SparkFun also offers some other accessories to implement your hardware which are listed below as a complement:

    - Pan/Tilt bracket suitable for the Micro Servo ROB-10335.

    - Camera case suitable for the above bracket PRT-12846.

    - Rasperry Pi enclosure DEV-12997.

    - Jumper wires M/F PRT-12794

References

www.raspberrypi.org

Raspberry Pi foundation provides the hardware (mini computer) and some of the documentation required to develop this project.

www.qt.io

Qt is the leading independent technology for cross-platform development. Most of the software implementation in this project (client & server) has been developed using Qt framework.

www.lavrsen.dk/foswiki/bin/view/Motion/WebHome

Motion is a program that monitors the video signal from cameras. It is able to detect if a significant part of the picture has changed; in other words, it can detect motion.

elinux.org/RPi_Low-level_peripherals

General Purpose Input/Output (GPIO) is a generic pin on a chip whose behavior (including whether it is an input or output pin) can be controlled (programmed) through software.

abyz.co.uk/rpi/pigpio/

pigpio is a C library for the Raspberry Pi which allows control of the general purpose input outputs (GPIOs).

www.sparkfun.com

www.rs-components.com

Electronic and hobby stores to get Raspberry Pi related products and accessories.