Motion-Camera

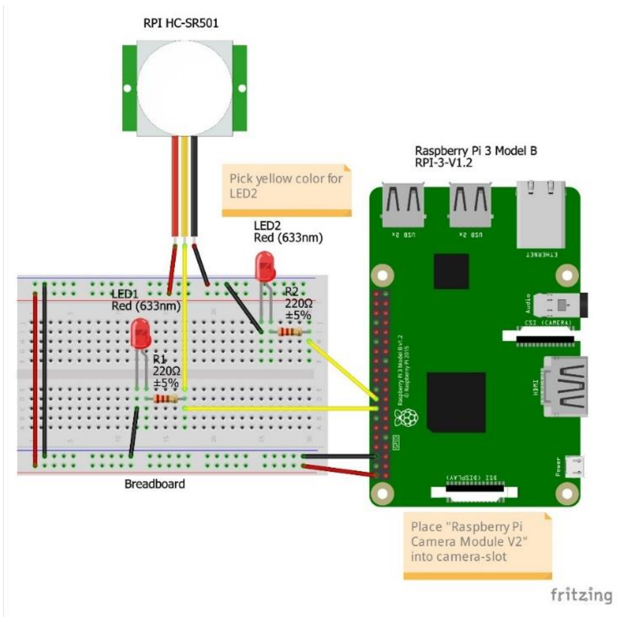
Introduction

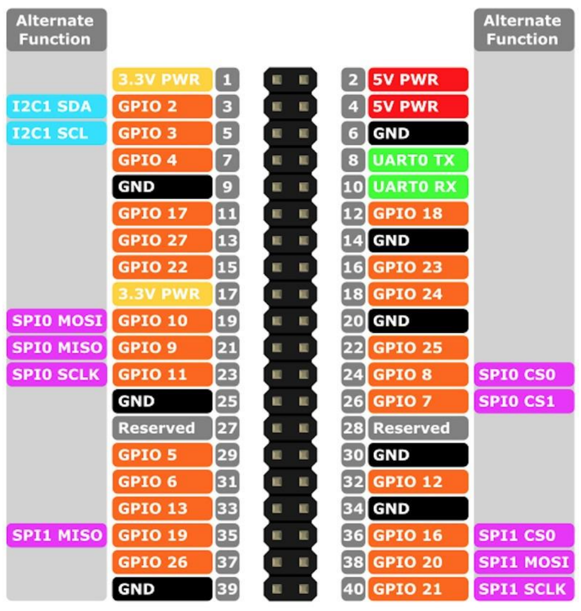
This project includes a surveillance camera system with any number of clients that can be switched on and off via a central interface. It should be possible to store recorded images on the network drive and, if necessary, to start a live stream from a targeted camera.

Hardware-Components

- Raspberry Pi 3 Model B
- RPI HC-SR501 (Motion-Sensor)
- Camera Module V2
- LED (1 x Red & 1 X Yellow)
- Resitors (2 x 220Ohm)
- Cables
- Micro-SD-Card (at least class 10 with UHS-I)

Hardware-Installation & Pins





Alternate Function					Alternate Function
	3.3V PWR	1	2	5V PWR	
I2C1 SDA	GPIO 2	3	4	5V PWR	
I2C1 SCL	GPIO 3	5	6	GND	
	GPIO 4	7	8	UART0 TX	
	GND	9	10	UART0 RX	
	GPIO 17	11	12	GPIO 18	
	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
	3.3V PWR	17	18	GPIO 24	
SPI0 MOSI	GPIO 10	19	20	GND	
SPI0 MISO	GPIO 9	21	22	GPIO 25	
SPI0 SCLK	GPIO 11	23	24	GPIO 8	SPI0 CS0
	GND	25	26	GPIO 7	SPI0 CS1
	Reserved	27	28	Reserved	
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
SPI1 MISO	GPIO 19	35	36	GPIO 16	SPI1 CS0
	GPIO 26	37	38	GPIO 20	SPI1 MOSI
	GND	39	40	GPIO 21	SPI1 SCLK

In the MitionCamera.pdf is a scetch of the build of the Raspberry-Pi with the Motion-Sensor and the Camera. Do not use other Pins on your own and build this scetch like its shown in the scetch. If you have done the installation correctly and power on the Raspberry-Pi without an image, the red LED should be one, if motion gets tracked. Note that the picture pins must be rotated 180 degrees so that the order of the pins matches the positioning of the Raspberry Pi with the sketch. The two pictures can be viewed enlarged at the following addresses:

- https://www.facing-south.com/img/motionCamera.jpg
- https://roboticsbackend.com/wp-content/uploads/2019/05/raspberry-pi-3-pinout.jpg

Software

Operating-System

The operating system that was installed on the Pi is the Raspberry Pi OS (Raspberry Pi OS (32-bit) Lite). The image can be found here: https://www.raspberrypi.org/downloads/raspberry-pi-os/

Auto-Login

1. Open the Raspberry Pi configuration tool with the following command: `sudo raspi-config`
2. Choose point 3: `Boot Options`
3. Now choose point B2:
   `Console Autologin Text console, automatically logged in as 'pi' user`
4. After a restart, the user pi should now be logged in automatically.

Camera-Interface

5. Open the Raspberry Pi configuration tool with the following command: `sudo raspi-config`
6. Choose point 5: `Interfacing Options`
7. Now choose point P1: `Camera`
8. Choose "Yes" to enable the Camera.

W-LAN (WPA2-Enterprise)

If you are not using WPA2-Enterprise skip this step and go to W-LAN (WPA2).

1. Open the Raspberry Pi configuration tool with the following command: `sudo raspi-config`
2. Choose point 4: `Internationalisation Options`
3. Now choose point I4: `Change Wi-fi Country` and set your country-code
4. Ensure that W-LAN Networks are available: `sudo iwlist wlan0 scan | egrep "(ESSID)"`
5. If the expected WLAN network is within range, the configuration can be made in wpa_supplicant.conf to establish the connection:
   `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`
6. The following must now be added here:

```
network={
 ssid="AU_WiFi"
 # For hidden SSIDs
 scan_ssid=1
 mode=0
 key_mgmt=WPA-EAP
 pairwise=CCMP TKIP
 identity="XXXXXXXX"
 password="XXXXXXXX"
 phase1="peaplabel=0"
 phase2="auth=MSCHAPV2"
}
```

7. Open File: `sudo nano /lib/dhcpcd/dhcpcd-hooks/10-wpa_supplicant`

8. Search for: `nl80211,wext`
9. And change it for: `wext,nl80211`
10. Reload network: `sudo service networking restart`
11. Reboot: `sudo Reboot`

W-LAN (WPA2)

If you are not using WPA2-Enterprise skip this step and go to W-LAN (WPA2).

1. Open the Raspberry Pi configuration tool with the following command: `sudo raspi-config`
2. Choose point 4: `Internationalisation Options`
3. Now choose point I4: `Change Wi-fi Country` and set your country-code
4. Ensure that W-LAN Networks are available: `sudo iwlist wlan0 scan | egrep "(ESSID)"`
5. If the expected WLAN network is within range, the configuration can be made in wpa_supplicant.conf to establish the connection:
   `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`
6. The following must now be added here:

```
network={

  ssid="AU_WiFi"

  psk="testingPassword"
}
```

7. Reload Network: `sudo service networking restart`
8. Reboot: `sudo Reboot`

Libraries/Updates

Um die in unserem Python-Script verwendeten Bibliotheken zu verwenden, müssen einige zusätzliche Pakete, sowie Updated installiert werden, dafür müssen die folgenden Befehle ausgeführt werden:

- `sudo apt update`
- `sudo apt full-upgrade`
- `sudo apt-get install python3-picamera`
- `sudo apt install python3-pip`
- `sudo pip3 install pyTelegramBotAPI`
- `sudo apt-get install rpi.gpio`
- `sudo apt-get install python3-dev`
- `sudo apt-get install python3-setuptools`
- `sudo apt-get install libtiff-dev`
- `sudo pip3 install Pillow`
- `sudo apt-get install libopenjp2-7`
- `sudo apt-get install libtiff5`
- `sudo pip3 install systemd`
- `sudo apt-get install git`

Directions

The following folders must be created:

- /home/pi/images
- /home/pi/github

Set permissions for both folders (you need to be inside the "github"-folder to enter these commands):

- `sudo chmod 777 github`
- `sudo chmod 777 images`

Download

The entire repository of the MotionCamera project can be loaded directly from Github into the "github" folder. When you are inside the „github“-folder entert he following command:

- `sudo git clone https://github.com/Facing-South/MotionCamera.git`

When this step has been completed, the folder structure should look like this:

- /home/pi/github/MotionCamera/MotionCameraClient

Starting Software

If you are now in the file path specified under Download after downloading the project folder, you can start the bot manually with the following input:

- python3 MotionCameraClient.py

As soon as the software is running, the command / start can be entered in the associated telegram group. Now all captured images are loaded directly into the telegram group.
As soon as the Raspberry-Pi is connected to the power, it also tries to start the monitoring software automatically. As soon as the Raspberry-Pi is connected to the power, it also tries to start the monitoring software automatically. You can tell whether the monitoring software could start itself after a waiting time of approx. 1-2 minutes by the flashing yellow LED, which will flash three times.

Implementing Auto-Start

Create Service File

Open a sample unit file using the command as shown below:

```
sudo nano /lib/systemd/system/sample.service
```

Add in the following text:

```
[Unit]

Description = Motion-Camera

After=multi-user.target


[Service]

Type=idle

ExecStart=/usr/bin/python3
/home/pi/github/MotionCamera/MotionCameraClient/MotionCameraClient.py

WorkingDirectory=/home/pi/github/MotionCamera/MotionCameraClient

#StandardOutput=inherit
```

```
#StandardError=inherit

Restart=always

User=pi


[Install]

WantedBy=multi-user.target
```

Safe those settings with ctrl + x and hit y.

The permission on the unit file needs to be set to 644 :

- `sudo chmod 644 /lib/systemd/system/sample.service`

Configure systemd

Now the unit file has been defined we can tell systemd to start it during the boot sequence:

- `sudo systemctl daemon-reload`
- `sudo systemctl enable sample.service`

Reboot the Pi and your custom service should run:

- `sudo reboot`

Usage

Power on

To start the surveillance camera, it is sufficient to connect it to the power provided that an image has been installed. After about 2 minutes, the yellow LED flashes 10 times. It will flash as long as the connectino to the bot could not be established. If the flashing stops, it is the sign that the camera can now receive commands from Telegram and thus start monitoring.

Error

If the system is unable to connect to the internet or there are other problems, the yellow LED also flashes 10 times over and over again.

Commands

The cameras are controlled via the Telegram app. The following commands can be sent to the bot, provided that you are in the same group as the bot:

Command	**Function**
/start	Starts the survaillance
/stop	Ends the survaillance

As soon as the motion detector detects movement, the red LED lights up. As soon as a photo has been taken, the yellow LED starts to flash. The motion sensor needs about 6 seconds after each movement to be able to react to the next movement. During this time the yellow LED flashes slowly. As soon as the motion sensor is ready to record the next motion, the yellow LED flashes three times in quick succession.

Telegram

Create Bot

To create a Telegram bot I recommend following these instructions: https://core.telegram.org/bots. If you follow the individual steps of these instructions, you will receive a unique token that belongs to the bot created and must be kept secret. Otherwise it could happen that the bot can be abused. This received token must be used in line 15 of the MotionCameraClient.py script (replace the "XXXXX" with your token):

- 15: bot = telebot.TeleBot("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")

After that you can Invite the bot into your telegram group.

Get Group-ID

So that the bot places the recorded surveillance images in the group in which you have invited the bot, the ID of this group has to be determined. To get this ID, I recommend following those instructions: https://stackoverflow.com/questions/32423837/telegram-bot-how-to-get-a-group-chat-id.
Once the ID has been determined, it must also be entered in the MotionCameraClient.py script. This happens in line 28 and looks like this:

- 28: bot.send_photo(chat_id=XXXXXX, photo=open("/home/pi/images/image.jpg", "rb"))

At this point, too, the "XXXXX" must be replaced by the determined group ID.

Reproduce

In order to reproduce this project, only the hardware has to be purchased and set up correctly. Then a copy of the motion camera image can be copied to a data carrier and inserted into the Raspberry Pi. If necessary, you can adjust the bot's token and the group ID in the following file:

- /home/pi/github/MotionCamera/MotionCameraClient

In this folder a text file with the name token.txt can be found. You can open this with the following command:

- sudo nano /home/pi/github/MotionCamera/MotionCameraClient/token.txt

Line 1 contains the token and line two the group ID. After these entries have been changed, the key combination ctrl + x must be pressed and then confirmed with y.