Project Description

Friday, January 18, 2019 1:46 PM

Carputer

The **Carputer** project came about from the need to have a simple backup car camera. Instead of buying an off-the-shelf solution I decided to build my own solution. Looking at different solutions; Intel NUC, OpenCV, JavaFX, Arduino, RaspberryPi the simplest solution appears to be to implement MotionEye on the Raspberry Pi for video streaming. The Raspberry Pi will also act as an access point which will allow and Android device to display the streaming video.

The decision was also made to utilize an Android tablet to display the streaming video from the Raspberry Pi's hosting MotionEye. The alternative was to use a similar size video monitor. The benefits of utilizing and Android device is the versatility. Another benefit of Android is to gain experience in creating applications for a mobile device. It has other applications and features that can used other than streaming video, such as music, maps, etc.. In addition, the Android device does not have to be left in the vehicle.

The rollout plan is to:

1. PRIMARY GOAL -> Display streaming video from Raspberry PI running MotionEye .
2. SECONDARY GOAL -> Display of OBD-II/CANBUS data. Raspberry PI will connect to OBD-II/CANBUS.
3. STRETCH GOAL -> Play music.
4. STRETCH GOAL -> Location services. Maps.
5. STRETCH GOAL -> Sync data. Upload data to cloud.

Architecture

Wednesday, April 12, 2017 5:04 PM

- **Computer**
  - RaspberryPi 3 Model B+
    - OS: **Raspbian**
    - **Hostname**: **Carputer**
    - Main computer
    - Acts as WiFI access point: **PINET**
    - Hosts **MotionEye**
      - Camera config for front camera
      - Camera config for rear camera
        - Camera type is network
        - Raspberry Pi Zero W is host for rear camera
    - Host for Front Camera
      - USB camera
    - Running **phpSysInfo**
      - Just because we can
    - **Mounting: TBD**
  - Raspberry Pi Zero W
    - OS: **Raspbian**
    - **Hostname**: **Carputer-Rear**
    - Hosts **MotionEye**
    - WiFI connection to **PINET**
    - Host for Rear Camera
      - USB camera

        or
      - Raspberry Pi Camera Pi v2 - 8 megapixel
    - **Mounting: TBD**

- **Display**
  - Android Tablet
    - Google Nexus 7 (post 2013)
    - Custom Android application
      - App name: **Carputer**
      - GitHub: https://github.com/billdoerr/Android-Carputer
    - WiFi connection to **PINET**
    - Able to utilize other Android features; Location, Compass, Music, Maps, etc
    - **Mounting: TBD**

- **Android Application**
  - App name: **Carputer**
  - GitHub: https://github.com/billdoerr/Android-Carputer
  - Features
    - MJPEG View
      - Front camera view
        - Ability to capture snapshot
      - Rear camera view
        - Ability to capture snapshot

        - Dual-view
            - Front/Rear camera view, display horizontally
    - Web View
        - For viewing **MotionEye**
        - For viewing **phpSysInfo**
        - Ability to take snapshot of screen
    - Image Archive
        - Can view snapshots taken
        - Able to delete snapshots
    - SSH
        - *Complete hack!*
        - Used to power off RaspberryPi's
        - Contains un-editable list of commonly used/needed commands
        - Able to manually execute a common via soft keyboard
    - Settings
        - Camera config
            - Ability to enable front or rear or both cameras
                - Front
                    - On or Off
                - Rear
                    - On or Off
                - Dual-view
                    - On or Off
            - Contains setting for camera streaming url's
                - http://IP:PORT
        - RaspberryPi config
            - Currently contains only configuration for **Carputer**
            - Config for multiple RaspberryPi's not implemented yet
            - Settings for **MotionEye**
            - Settings for **phpSysInfo**
            - Settings for SSH
                - IP
                - User
                - Pwd

- **Router**
    - RaspberryPi 3 Model B+
    - WiFi Access Point only, no internet
        - WiFi network id: **PINET**

- **Camera**
    - Front
        - RaspberryPi 3 Model B+
        - Logitech USB Camera (HD Pro Webcam C920)
    - Rear
        - Raspberry Pi Zero W
        - Logitech USB Camera (HD Pro Webcam C920)

          or
        - Raspberry Pi Camera Pi v2 - 8 megapixel
    - Left

        - Not deployed
    - Right
        - Not deployed

- **Storage**
    - OS
        - SanDisk 32GB Ultra microSDHC
    - MotionEye archive
        - SanDisk 64GB Cruzer Blade CZ50 USB 2.0 Flash Drive
        - One week retention of archived video

- **Power**
    - Cigarette lighter adapter, Usb Power
        - No power when automobile is not engaged
        - Two units
        - Power for main computer
        - Power for rear camera
    - Output, shared
        - 5V DC
        - 2.4 A
        - 1.0 A
    - Input
        - 12 - 24V DC
    - Juice4halt - J4H-5V-USB
        - Micro-UPS for Raspberry Pi using supercapacitors
        - Only used to supply power for graceful shutdown when no main power.

- **Cable**
    - USB Type A - USB Micro Type B
        - **Carputer** to cigarette lighter power adapter
    - USB Type A - USB Micro Type B
        - **Carputer-Rear** to cigarette lighter power adapter
    - Android Tablet
        - Only needed if table power is low
        - USB Type A - USB Micro Type B

- **Sensors - implement at a later date**
    - GPS
    - Accelerometer
    - Compass
    - Temperature
        - Inside
        - Outside
    - OBD-II
        - Bluetooth

- **Music - implement at a later date**

BOM

Wednesday, April 12, 2017 5:16 PM

Item	Qty	Unit Price	Total Price	Vendor	Link
Raspberry Pi 3 Model B+	1	39.95	39.95	SparkFun	https://www.sparkfun.com/products/14643
Raspberry Pi Zero W	1	10.00	10.00	SparkFun	https://www.sparkfun.com/products/14277
Raspberry Pi Camera Pi v2	1	29.95	29.95	SparkFun	https://www.sparkfun.com/products/14028
Pi Tin for the Raspberry Pi - Black	1	5.95	5.95	SparkFun	https://www.sparkfun.com/products/13102
Raspberry Pi Zero Case	1	5.95	5.95	SparkFun	https://www.sparkfun.com/products/14273
Real Time Clock Module - DS1307	1	15.95	15.95	SparkFun	https://www.sparkfun.com/products/12708
Juice4halt - J4H-5V-USB	1	70.00	70.00	Juice4halt (Europe - Slovakia)	https://juice4halt.com/
SanDisk 32GB Ultra microSDHC	2	7.69	15.38	Newegg	https://www.newegg.com/Product/Product.aspx?Item=9SIA12K65X1049&nm_mc=TEMC-RMA-Approvel&cm_mmc=TEMC-RMA-Approvel-_-Content-_-text-_-
SanDisk 64GB Cruzer Blade CZ50 USB 2.0 Flash Drive	1	9.35	9.35	Newegg	https://www.newegg.com/Product/Product.aspx?Item=9SIAET87A41486&nm_mc=TEMC-RMA-Approvel&cm_mmc=TEMC-RMA-Approvel-_-Content-_-text-_-
Cigarette Lighter Power Adapter	2	12.00	24.00	QFC	Yes, the grocery store.
Logitech USB Camera (HD Pro Webcam C920)	1	0.00	0.00	N/A	Discontinued. Already in possession.
Google Nexus 7 (post 2013)	1	0.00	0.00	ASUS	Discontinued. Already in possession. I love this device. Too bad they are discontinued. Would have like an Nvidia Shield also.
USB Type A - USB Micro Type B	3	0.00	0.00	N/A	Doesn't everyone have like a dozen of these.
Total:		**206.79**	**226.48**		

Deployment Instructions

Friday, January 18, 2019 1:43 PM

Android

Friday, January 18, 2019 1:43 PM

- **Android Application**
  - App name: **Carputer**
  - GitHub: https://github.com/billdoerr/Android-Carputer
  - Current Version: (NOT RELEASED)

  To deploy from Android Studio, click **Run -> Run 'app'** then select the your connected Android device.

Settings

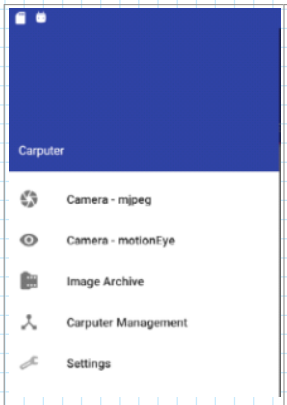
Sunday, January 20, 2019 7:10 AM

Carputer Settings

To configure the **Carputer** application requires the following settings. This is accessed via the app's main navigation menu or via the navigation drawer.

Note: The Raspberry Pi image for both the front and rear cameras have static ip's. The default values for Url's contained in the app settings have default values that reflect these static ip's. The Raspberry Pi image also has the motionEye application pre-configured and the app settings also reflect this with default values.

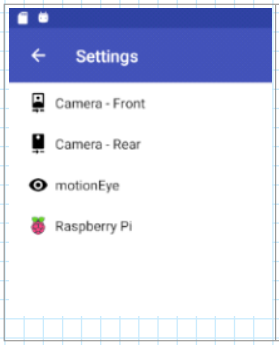
Navigation Drawer/Main Menu



The main menu and navigation drawer (opens from left edge) contains the following menu items.

Camera - mjpeg	Streaming video of front and rear cameras. Contains option menu, Snapshot, to capture image.
Camera - motionEye	Web view of motionEye video streaming and admin console.
Image Archive	List of snapshot images. Ability to delete images with a single tap and confirmation.
Carputer Management	View for SSH commands to the Raspberry Pi and web view of phpSysInfo running on the Pi.
Settings	Camera and Raspberry Pi node configuration settings.

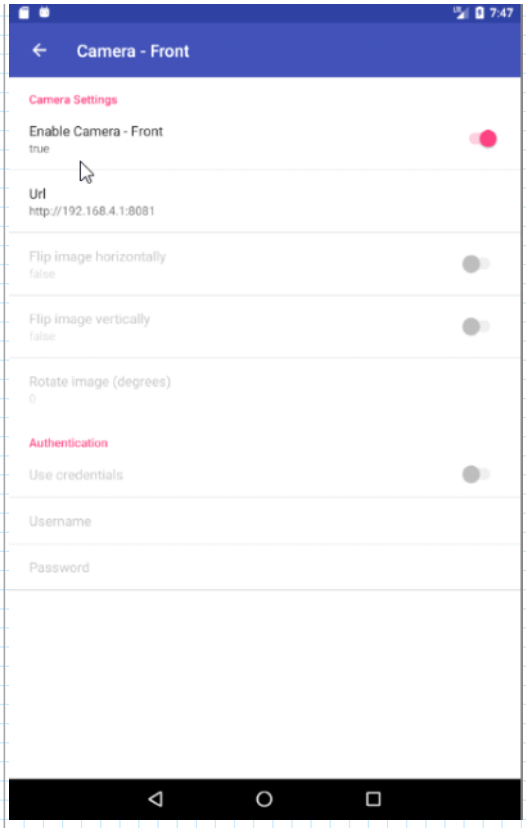
Settings



The Settings menu items which contain the following:

Camera - Front	Configuration settings for the front camera.
Camera - Rear	Configuration settings for the rear camera.
motionEye	Configuration settings for the motionEye application.
Raspberry Pi	Configuration settings for the main Raspberry Pi node.

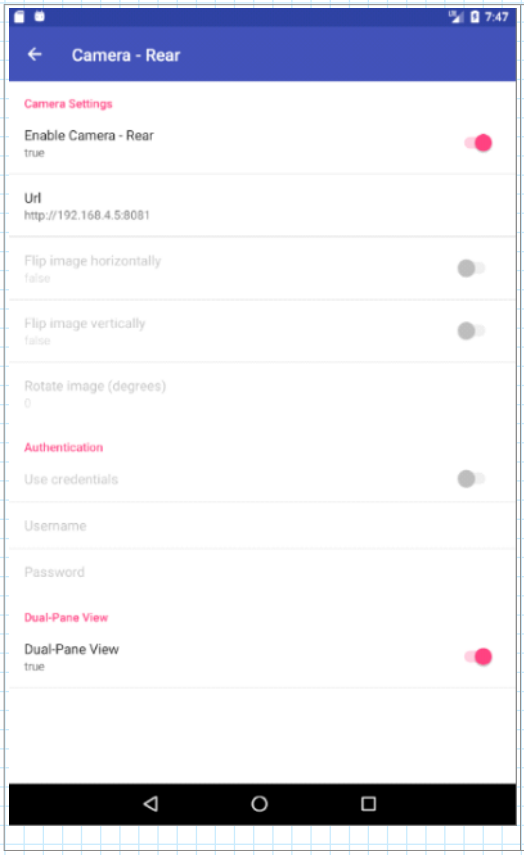
Camera Front



Settings configuration for the front camera, if enabled.

Enable Camera - Front	Slide if front camera is enabled.
Url	http://IP:PORT of motionEye streaming video url.
Flip image horizontally	Currently not supported.
Flip image vertically	Currently not supported.
Rotate image (degrees)	Currently not supported.
User credentials	Currently not supported.
Username	Currently not supported.
Password	Currently not supported.

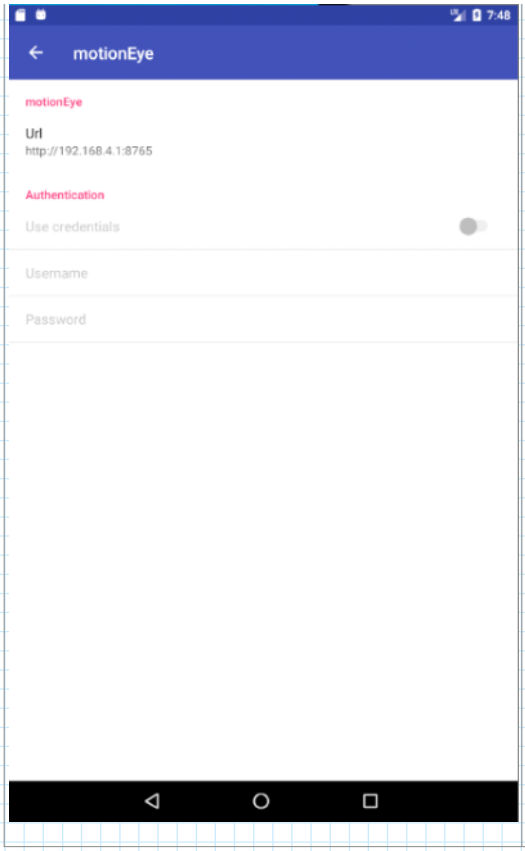
Camera Rear



Settings configuration for the rear camera, if enabled. This is also where you enable the dual-pane view.

Enable Camera - Rear	Slide if front camera is enabled.
Url	http://IP:PORT of motionEye streaming video url.
Flip image horizontally	Currently not supported.
Flip image vertically	Currently not supported.
Rotate image (degrees)	Currently not supported.
User credentials	Currently not supported.
Username	Currently not supported.
Password	Currently not supported.

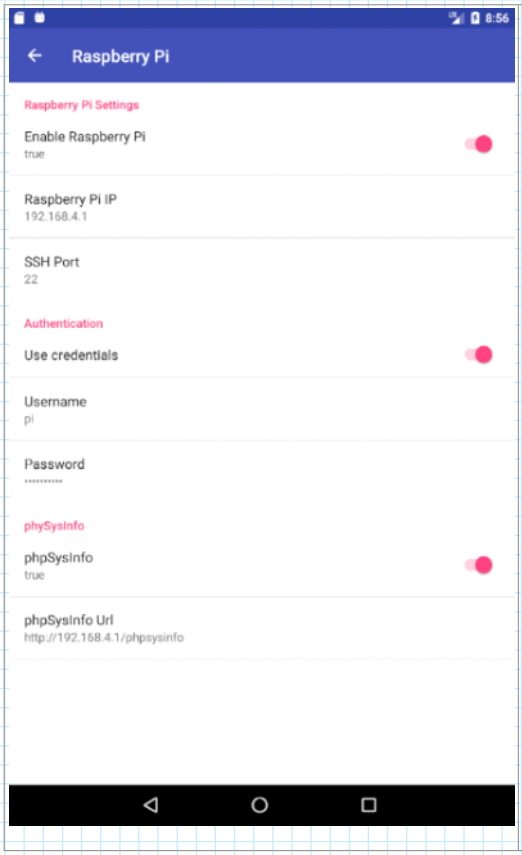
motionEye



Configuration settings for the motionEye application running on the Raspberry Pi. Currently only supports one motionEye application which is assumed the one running on the main Carputer Raspberry Pi. As a work-around, the values can be change to connect to other Raspberry Pi's that are connected to the PINET network.

Url	http://IP:PORT of motionEye admin console.
User credentials	Currently not supported.
Username	Currently not supported.
Password	Currently not supported.

Raspberry Pi



Raspberry Pi configuration settings. Currently only one node is supported and assumed these settings correspond to the main Carputer Raspberry Pi. As a work-around, the values can be change to connect to other Raspberry Pi's that are connected to the PINET network.

Enable Raspberry Pi	Slide to enable this configuration settings.
Raspberry Pi IP	IP of the main Raspberry Pi.
SSH Port	Port defined for SSH.
Use Credentials	Slide to enable the use of credentials for SSH.
Username	SSH user.
Password	SSH password.
phpSysInfo	Slide to enable the web view of the phpSysInfo monitoring web page.
phpSysInfo Url	URL of phpSysInfo monitoring web page.

RaspberryPi - Carputer

Friday, January 18, 2019 1:42 PM 5/9/2019 8:29 AM

- **Raspberry Pi Image Creation for Carputer**

Below outlines the steps needed to create the Raspberry Pi image for the main **Carputer** node.

1. **Download Image**
   a. From https://www.raspberrypi.org/downloads/raspbian/, download the '**Raspbian Stretch with desktop and recommended software**' image dated '**November 2018**'.

2. **Write to SD Card**
   a. Follow these instructions to install the operating system to an SD card: https://www.raspberrypi.org/documentation/installation/installing-images/README.md.

3. **Setup WiFi without keyboard or network access.**
   This step will allow you to connect your Raspberry Pi to your network then you can using a terminal emulator like 'PuTTy' to access the device.

   a. In the root of the SD card create a new file named **wpa_supplicant**.
   b. In the file add:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
ssid="YourNetworkSSID"
psk="Your Network's Passphrase"
key_mgmt=WPA-PSK
}
```

   c. **Note:** Ensure you change the values for **ssid** and **psk**.
   d. Save the file using the Linux LF. In Notepad++ this is configured by going to Edit -> EOL Conversion - Unix (LF).

4. **Setup SSH Access**
   This step will setup SSH access so that you can connect to the Raspberry Pi from other computers on the same network.

   a. All that is need is to create a new empty file named **ssh** in the root of the SD card.
   b. Save the file using the Linux LF. In Notepad++ this is configured by going to Edit -> EOL Conversion - Unix (LF).

5. **Access the Raspberry Pi**
   a. Using an application connect to the device. You may need to use a program like **Advance Ip Scanner** to determine the IP of the Raspberry device.
      i. The default Raspbian user/password is: **pi/raspberry**
   b. Once authenticated change the default password. Enter **passwd** command and enter the new password **scoobydoo**.
   c. I then run the command **sudo reboot**. Once back on-line validate the password change.

6. **Updates/Upgrades**
   This step is to update and upgrade the packages.

   a. Enter the command: **sudo apt-get update**
   b. Enter the command: **sudo apt-get upgrade**

7. **Enable VNC**
   This is an optional step, but useful nonetheless. You will need a desktop program like **VNC Viewer**. For detail instructions reference: https://www.raspberrypi.org/documentation/remote-access/vnc/.

   a. sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer
   b. sudo raspi-config. This will launch an ASCI UI.

   Now, enable VNC Server by doing the following:
   - Navigate to Interfacing Options.

   Scroll down and select **VNC -> Yes**

8. **raspi-config**
   This step is to change the hostname of the device.

   a. Using a program like **VNC Viewer** connect to the Raspberry Pi. The first time the device is accessed the **raspi-config** program should launch automatically. If not, click on the **Raspberry Icon -> Preferences -> Raspberry Pi Configuration**.
      i. Ignore changing password.
      ii. Change Hostname: **carputer**
      iii. Change resolution: 1280 x 1024
      iv. Change keyboard
      v. Change localization
   b. Reboot

9. **USB Flash Drive**

Click the link for the steps to configure a USB flash drive that will be used for motionEye video archiving.

10. **phySysInfo**
This step is completely optional. This will install the phpSysInfo monitoring tool on the Raspberry Pi. Reference: http://phpsysinfo.github.io/phpsysinfo/

a. sudo apt-get install apache2 php5 libapache2-mod-php5 (if not already done) I get a bunch of errors.
b. sudo apt-get install phpsysinfo
c. sudo apt-get install php-mbstring
d. sudo ln -s /usr/share/phpsysinfo /var/www/html
e. sudo reboot

Then enter into your browser: http://<your_ip>/phpsysinfo

11. **motionEye**
Click the link for the steps to install **motionEye**.

12. **Pause - Create Image**
At this point I would recommend creating an image of the SD card. The next step involves setting the Raspberry Pi as an access point. If this fails you can easily write the saved image to the SD card rather than to manually redo the installation steps up to this point.

13. **Configure the Raspberry Pi as an Access Point**
Follow this link for the steps to configure the Raspberry Pi as an access point.

Note: The /etc/hostapd/hostapd.conf will have the following values:

```
interface=wlan0
#driver=nl80211                 <- Comment out this line
ssid=PINET                      <- Set to PINET
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=scoobydoo        <- Set to scoobydoo
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

14. **Validate you are now able to connect to PINET**
a. From another device with WIFI capabilities verify that the PINET access point is available.
b. Make a connection to the PINET access point using the passphrase supplied in the hostapd.conf. Note: There will not be any internet connectivity.

15. **motionEye Camera Configuration**
Click this link for the steps to configure the cameras that will be used for video streaming.

16. **Real Time Clock (RTC)**
Reference this link for steps to implement RTC. Device used is the Adafruit DS1307.

DS1307 Wiring
- DS1307 Vin connects to RPi Pin 4
- DS1307 GND connects to RPi Pin 6
- DS1307 SDA connects to RPi Pin 3
- DS1307 SCL connects to RPi Pin 5

Configure the Raspberry Pi for I2C

Enable then I2C interfaces and automatic loading of the I2C kernel module.
- sudo raspi-config

If I2C interfaces was not enabled.
- sudo reboot
- sudo apt-get install python-smbus i2c-tools
- sudo i2cdetect -y 1
  You should see ID #68.

Set up the Raspberry Pi RTC Time

- sudo vi /boot/config.txt

Add the following to bottom of file.
- dtoverlay=i2c-rtc,ds1307
- sudo reboot
- sudo i2cdetect -y 1

You should see a wall of text appear, if UU appears instead of 68.

Remove fake hwclock

- sudo apt-get -y remove fake-hwclock
- sudo update-rc.d -f fake-hwclock remove

Now that we have disabled the fake-hwclock package we can proceed with getting the original hardware clock script that is included in Raspbian up and running again by commenting out a section of code.

Run the following command to begin editing the original RTC script.

- sudo nano /lib/udev/hwclock-set

Find and comment out the following three lines by placing # in front of it as we have done below.

```
if [ -e /run/systemd/system ] ; then
   exit 0
fi
```

Replace With

```
#if [ -e /run/systemd/system ] ; then
#    exit 0
#fi
```

- sudo date -s "5 Jun 2020 10:22:45"
- sudo hwclock -w
- sudo hwclock -r --debug

Disable systemd-timesyncd - systemd-timesyncd is a system service that may be used to synchronize the local system clock with a remote Network Time Protocol server.

- sudo systemctl stop systemd-timesyncd.service
- sudo systemctl disable systemd-timesyncd.service

- sudo reboot

Validate RTC has been correctly implemented

- date
- sudo i2cdetect -y 1
- sudo hwclock -r --debug

From <https://www.raspberrypi.org/forums/viewtopic.php?t=211308>

17. **Power Off Push Button Wiring**
Used in conjunction with the listen_for_shutdown.py script.

- BUTTON connects to board pin 37 (GPIO26)
- BUTTON connects to board pin 39 (GND)

18. **Power Off Push Button LED (Yellow) Wiring**
Used in conjunction with the listen_for_shutdown.py script.

- LED (Yellow) connects to board pin 33 (GPIO13)
- LED (Yellow) -> RESISTOR (330 ohm) connects to board pin 34 (GND)

19. **Heartbeat LED (Red) Wiring**
Used in conjunction with the heartbeat.py script.

- LED (Red) connects to Rpi pin 13 (GPIO27)
- LED (Red) -> RESISTOR (330 ohm) connects to Rpi pin 14 (GND)

20. **Install Python Script - listen_for_shutdown.py**

- sudo apt install python3-paramiko **< install required ssh package**
- cd /etc/init.d
- sudo vi listen_for_shutdown.sh
- sudo chmod +x listen_for_shutdown.sh
- sudo mkdir /var/log/carputer
- cd /usr/local/bin
- sudo vi listen_for_shutdown.py
- sudo chmod +x listen_for_shutdown.py
- sudo update-rc.d listen_for_shutdown.sh defaults
- sudo reboot
- cd /var/log/carputer/
- ls
- tail -f listen_for_shutdown.log

21. **Install Python Script - heartbeat.py**

- cd /etc/init.d
- sudo vi heartbeat.sh

- sudo chmod +x heartbeat.sh
- sudo mkdir /var/log/carputer
- cd /usr/local/bin
- sudo vi heartbeat.py
- sudo chmod +x heartbeat.py
- sudo update-rc.d heartbeat.sh defaults
- sudo reboot
- cd /var/log/carputer/
- ls
- tail -f heartbeat.log

22. Setting up the Raspberry Pi for Juice4halt

- Introduced in v1.6 with the addition of the Juice4halt UPS Pi HAT.

- Install Python Script - juice4halt.py
  - cd /usr/local/bin
  - sudo vi juice4halt.py

- Create Shell Script - shutdown_script.sh
  - mkdir -p /home/pi/juice4halt/bin/
  - cd /home/pi/juice4halt/bin
  - vi shutdown_script.sh
  - chmod 755 shutdown_script.sh

- Configure Call to Shutdown Script - /etc/rc.local file
  - sudo vi /etc/rc.local
  - Add the following line before the exit 0 in the last line
    # Call the juice4halt shutdown script. This will shutdown slave nodes before master node.
    /home/pi/juice4halt/bin/shutdown_script.sh &

23. Image Archive

This step allows Apache to display directory index. Also creates a symbolic link to the motionEye video archives.

a. cd /var/www/html
b. sudo mv index.html index.bak
c. ln -s /mnt/motioneye image_archive

24. Install firmware version script

- cd /bin
- sudo vi carputer
  - #! /bin/sh
  - cat /etc/carputer/version

- sudo chmod +x carputer

25. Update firmware version

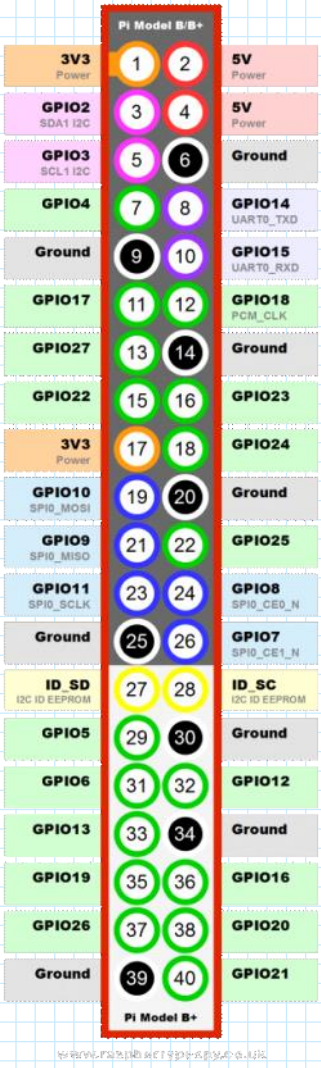
- cd /etc
- sudo mkdir carputer
- cd carputer
- sudo vi version
  Carputer master node
  v1.6.1
  Released 24Jun2022

26. Create Image

Create an image of the SD card. This allows you to have a backup of the image used for the Carputer. This will allow you to have a development device that is exactly the same as deployed to your vehicle.

Put version number used in Step #25 in filename: **2018-11-13-raspbian-stretch-motioneye-pinet_v1.6.1.img**.

Note: Even number pins are the outside of the board
Odd number pins are toward the inside of the board.



RaspberryPi - Carputer-Camera-Rear

Friday, January 18, 2019 1:42 PM

- **Raspberry Pi Image Creation for the Carputer-CameraRear node**

Below outlines the steps needed to create the Raspberry Pi image for the main **Carputer-CameraRear** node.

1. **Download Image**
   a. From https://www.raspberrypi.org/downloads/raspbian/, download the '**Raspbian Stretch with desktop and recommended software** ' image dated '**November 2018**'.

2. **Write to SD Card**
   a. Follow these instructions to install the operating system to an SD card: https://www.raspberrypi.org/documentation/installation/installing-images/README.md.

3. **Setup WiFi without keyboard or network access.**
   This step will allow you to connect your Raspberry Pi to your network then you can using a terminal emulator like 'PuTTy' to access the device.

   a. In the root of the SD card create a new file named **wpa_supplicant**.
   b. In the file add:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
ssid="PINET"
psk="scoobydoo"
key_mgmt=WPA-PSK
}
```

   c. **Note:** The values for **ssid** and **psk** should match the values configured in the hostapd.conf on the main **Carputer** node.
   d. Save the file using the Linux LF. In Notepad++ this is configured by going to Edit -> EOL Conversion - Unix (LF).

4. **Setup SSH Access**
   This step will setup SSH access so that you can connect to the Raspberry Pi from other computers on the same network.

   a. All that is need is to create a new empty file named **ssh** in the root of the SD card.
   b. Save the file using the Linux LF. In Notepad++ this is configured by going to Edit -> EOL Conversion - Unix (LF).

5. **Access the Raspberry Pi**
   a. Using an application connect to the device. You may need to use a program like **Advance Ip Scanner** to determine the IP of the Raspberry device.
      i. The default Raspbian user/password is: **pi/raspberry**
   b. Once authenticated change the default password. Enter **passwd** command and enter the new password **scoobydoo**.
   c. I then run the command **sudo reboot**. Once back on-line validate the password change.

6. **Updates/Upgrades**
   This step is to update and upgrade the packages.

   a. Enter the command: **sudo apt-get update**
   b. Enter the command: **sudo apt-get upgrade**

7. **Enable VNC**
   This is an optional step, but useful nonetheless. You will need a desktop program like **VNC Viewer**. For detail instructions reference: https://www.raspberrypi.org/documentation/remote-access/vnc/.

   a. sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer
   b. sudo raspi-config. This will launch an ASCI UI.

   Now, enable VNC Server by doing the following:
   - Navigate to Interfacing Options.

   Scroll down and select **VNC -> Yes**

8. **raspi-config**
   This step is to change the hostname of the device.

   a. Using a program like **VNC Viewer** connect to the Raspberry Pi. The first time the device is accessed the **raspi-config** program should launch automatically. If not, click on the **Raspberry Icon -> Preferences -> Raspberry Pi Configuration**.
      i. Ignore changing password.
      ii. Change Hostname: **carputer-camera-rear**
      iii. Change resolution: 1280 x 1024
      iv. Change keyboard
      v. Change localization
   b. Reboot

9. **motionEye**
   Click the link for the steps to install **motionEye**.

10. **Configure Static IP**

In this step you will configure the node to have a static Ip. This will give the node a static Ip of **192.168.4.5.**

a. sudo nano /etc/dhcpcd.conf

```
interface eth0
static ip_address=192.168.4.6/24
static routers=192.168.4.1
static domain_name_servers=192.168.4.1

interface wlan0
static ip_address=192.168.4.5/24
static routers=192.168.4.1
static domain_name_servers=192.168.4.1
```

b. sudo reboot

11. **Validate you are now able to connect to PINET**

a. The access point will have a static IP of: **192.168.4.5**
b. From another device with WIFI capabilities verify that the PINET access point is available.
c. Make a connection to the PINET access point using the passphrase supplied in the hostapd.conf. Note: There will not be any internet connectivity.
d. List dhcp leases from the main **Carputer** (192.168.4.1) node.
   - sudo cat /var/lib/misc/dnsmasq.leases

12. **motionEye Camera Configuration**

Click this link for the steps to configure the cameras that will be used for video streaming.

13. **Install SSHPass**

- sudo apt-get install sshpass

14. **Disable systemd-timesyncd**

systemd-timesyncd is a system service that may be used to synchronize the local system clock with a remote Network Time Protocol serv er.

- sudo systemctl stop systemd-timesyncd.service
- sudo systemctl disable systemd-timesyncd.service
- sudo systemctl status systemd-timesyncd.service

15. **Install script for updating date/time**

This step installs a script that will update the date/time with the master node after boot.

- crontab -e

```
# Sync date/time with master
@reboot /bin/timesync.sh
```

Verify cron job.

- crontab -l

Install timesync.sh script

- cd /bin
- sudo vi /bin/timesync.sh

  (Insert contents of timesync.sh)

Make executable

- sudo chmod 777 timesync.sh

16. **Install firmware version script**

- cd /bin
- sudo vi carputer
  - #! /bin/sh
  - cat /etc/carputer/version
- sudo chmod +x carputer

17. **Update firmware version**

- cd /etc
- sudo mkdir carputer
- cd carputer
- sudo vi version

```
Carputer-Camera-Rear
v1.2.1
Released 9Jan2022
```

18. **Create Image**

Create an image of the SD card. This allows you to have a backup of the image used for the Carputer. This will allow you to have a development device that is exactly the same as deployed to your vehicle.

Put version number used in Step #16 in filename: **2018-11-13-raspbian-stretch-motioneye-pinet-static-camera-rear_v1.2.1.img**.

motionEye Configuration - Carputer

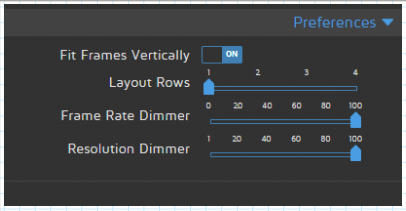
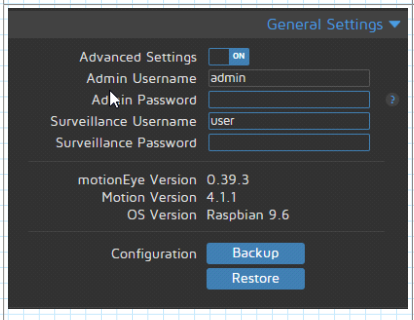
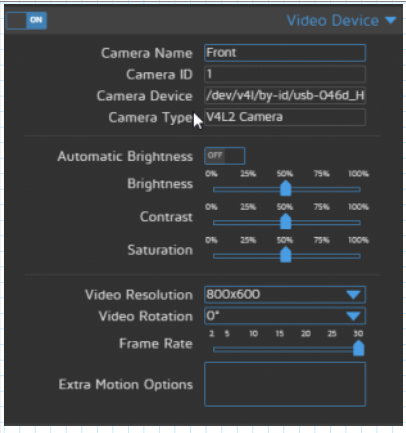
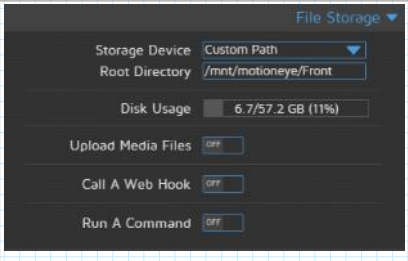
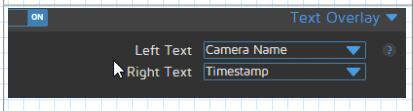
Tuesday, January 22, 2019 10:27 AM

This section describes the camera configuration within the **motionEye** admin console. http://IP:8765

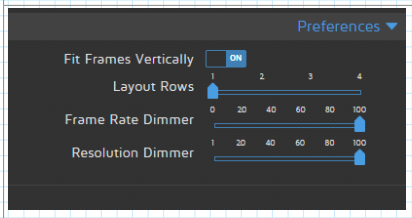
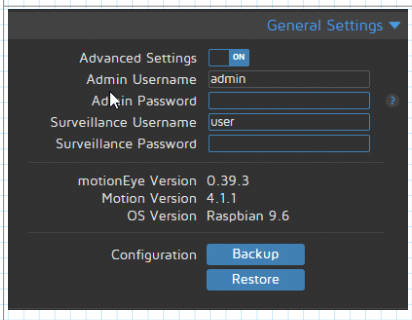
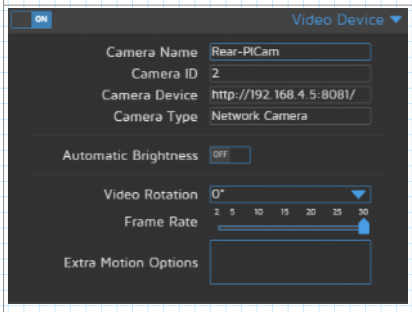
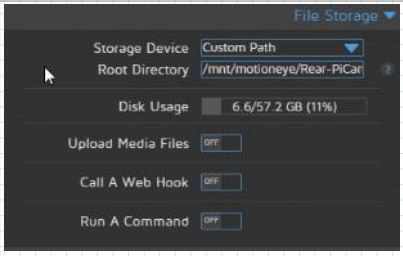
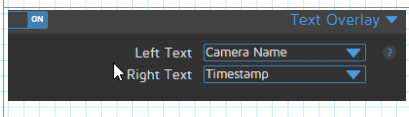
User/Pwd: admin/<blank>

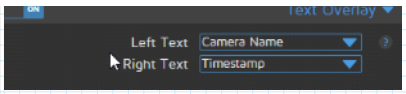
Note: First configure the **carputer-camera-rear** node then configure the **carputer** node. The **carputer** node configure will require the streaming URL of the **carputer-camera-rear** node.

- **motionEye Front Camera Configuration**

Preference Settings	Value
	**Add Camera** Camera Type: Local V4L2 Camera Camera: HD Pro Webcam C920
	**Preferences** Fit Frames Vertically: On Layout Rows: 1
	**General Settings** • Use default values
	**Video Device** Camera Name: Front Video Resolution: 800x600 Frame Rate: 30
	**File Storage** Store Device: Custom Path Root Directory: /mnt/motioneye/Front
	**Text Overlay** • Use default values

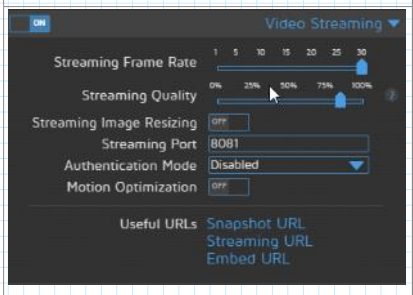
- **motionEye Rear Camera Configuration**

Preference Settings	Value
	**Add Camera** Camera Type: Network Camera Layout Rows: http://192.168.4.5:8081 **Note:** This is this streaming URL when we configured the **carputer-camera-rear** node. Camera: MJPEG Network Camera
	**Preferences** Fit Frames Vertically: On Layout Rows: 1
	**General Settings** • Use default values
	**Video Device** Camera Name: Rear-PiCam Frame Rate: 30
	**File Storage** Store Device: Custom Path Root Directory: /mnt/motioneye/Rear-PiCam
	**Text Overlay** • Use default values
	**Video Streaming**



Text Overlay

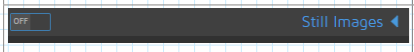
- Use default values



Video Streaming

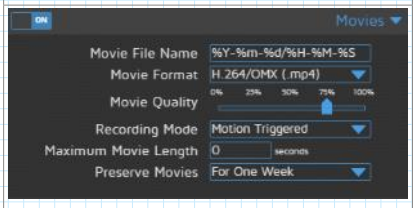
Streaming Frame Rate	30
Steaming Quality	100%
Streaming Port	8081

Note: You will use this **Streaming URL** when configuring the cameras within the Android Carputer app settings.



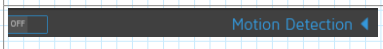
Still Images

- Not enabled



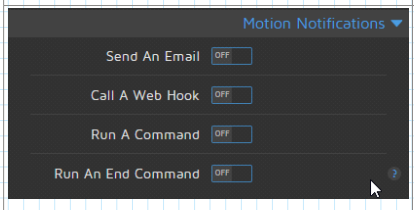
Movies

Movie File Name	Default Value
Movie Format	H.264/OMX (.mp4)
Movie Quality	75%
Recording Mode	Continuous
Max. Movie Length	600
Preserve Movies	3 days



Motion Detection

- Not enabled

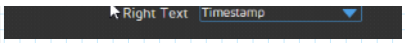


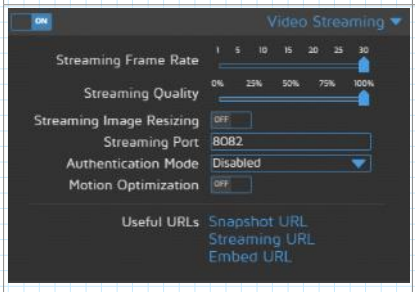
Motion Notifications

- Not configured

Working Schedule

- Not enabled

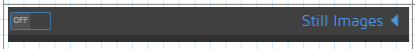




Video Streaming

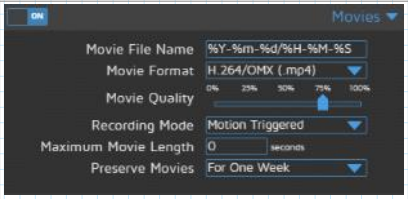
Streaming Frame Rate	30
Steaming Quality	100%
Streaming Port	8082

Note: You will use this **Streaming URL** when configuring the cameras within the Android Carputer app settings.



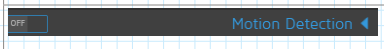
Still Images

- Not enabled



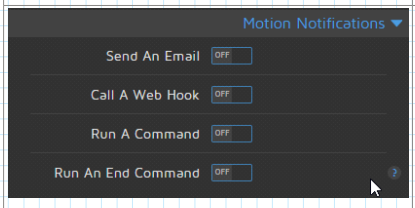
Movies

Movie File Name	Default Value
Movie Format	H.264/OMX (.mp4)
Movie Quality	75%
Recording Mode	Continuous
Max. Movie Length	600
Preserve Movies	3 days



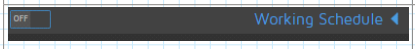
Motion Detection

- Not enabled



Motion Notifications

- Not configured



Working Schedule

- Not enabled

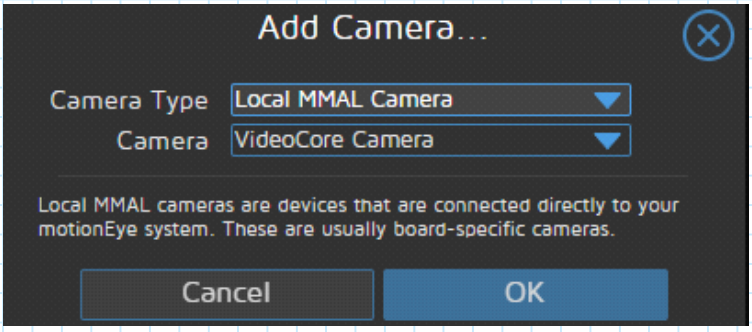
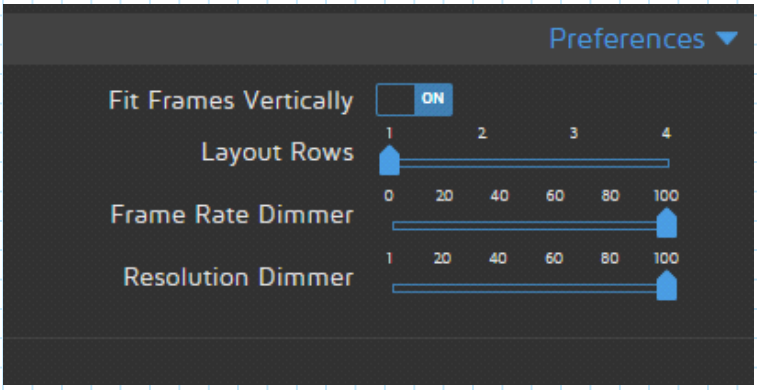
motionEye Configuration - Carputer-Camera-Rear

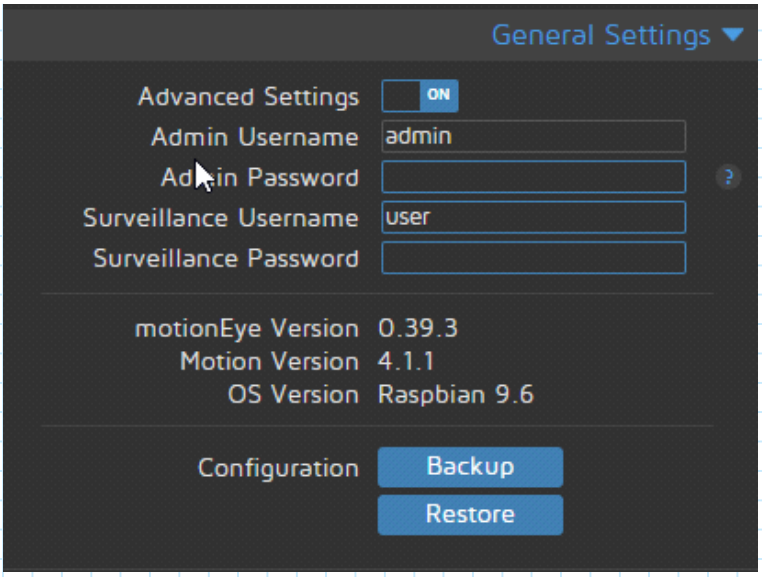
Tuesday, January 22, 2019 10:27 AM

This section describes the camera configuration within the **motionEye** admin console. http://IP:8765

Note: First configure the **carputer-camera-rear** node then configure the **carputer** node. The **carputer** node configure will require the streaming URL of the **carputer-camera-rear** node.

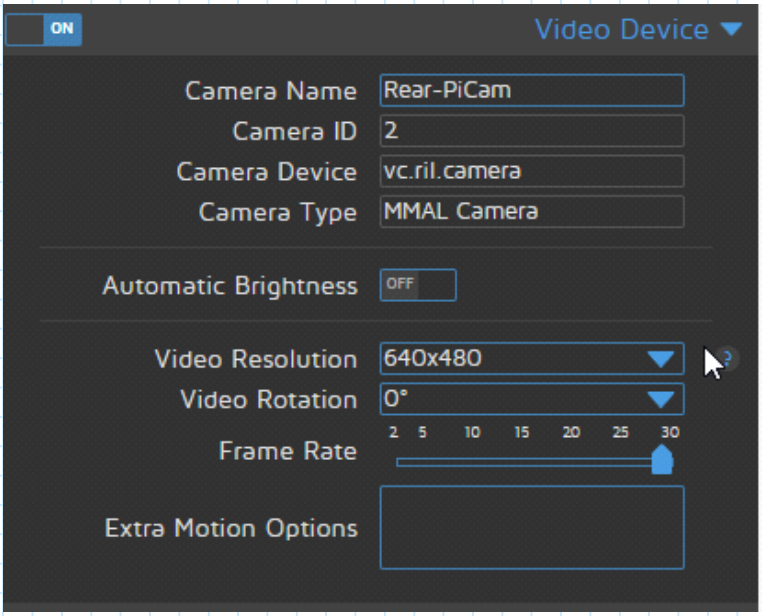
- **motionEye Camera Configuration - Carputer-Camera-Rear**

Preference Settings	Value
	**Add Camera** **Camera Type** Local MMAL Camera **Layout Rows** VideoCore Camera
	**Preferences** **Fit Frames Vertically** On **Layout Rows** 1



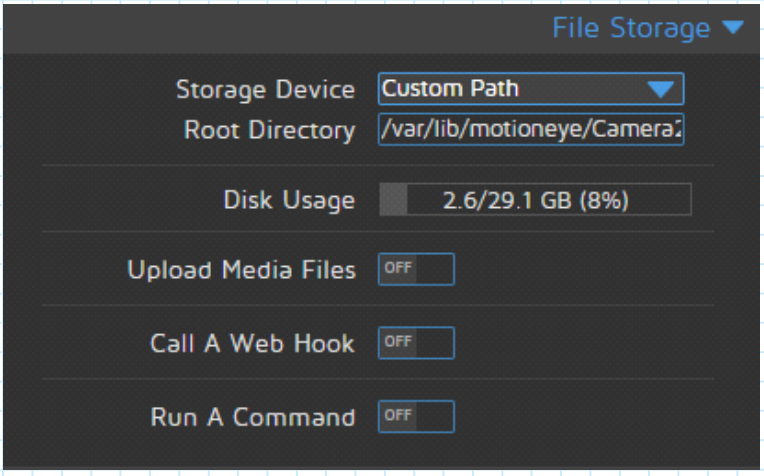
General Settings

- Use default values



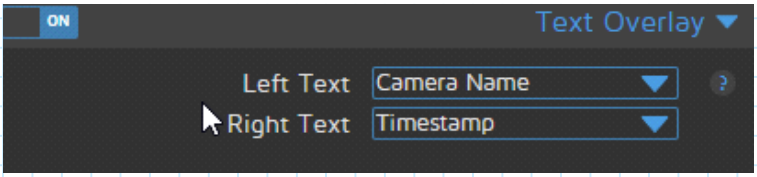
Video Device

Camera Name	Rear-PiCam
Video Resolution	640x480
Frame Rate	30



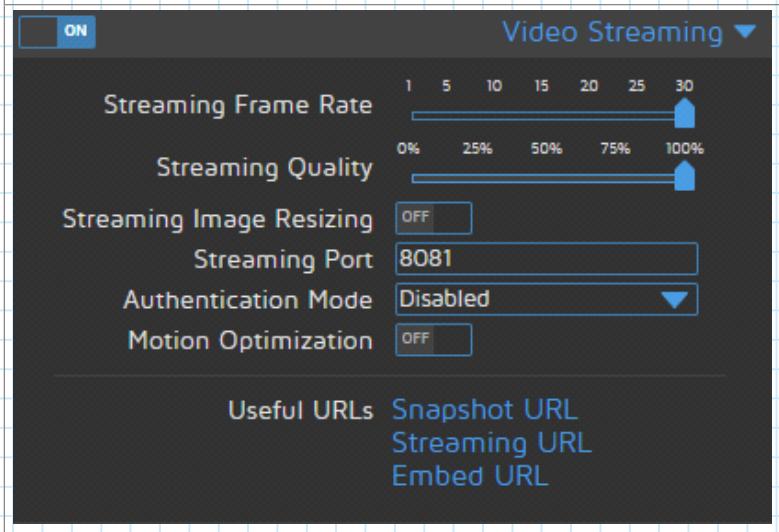
File Storage

- We are not performing any file storage on the **carputer-camera-rear** node.



Text Overlay

- Use default values



Video Streaming

Streaming Frame Rate	30
Steaming Quality	100%
Streaming Port	8081

Note: You will use this **Streaming URL** when configuring **motionEye** on the **carputer** node.

Still Images

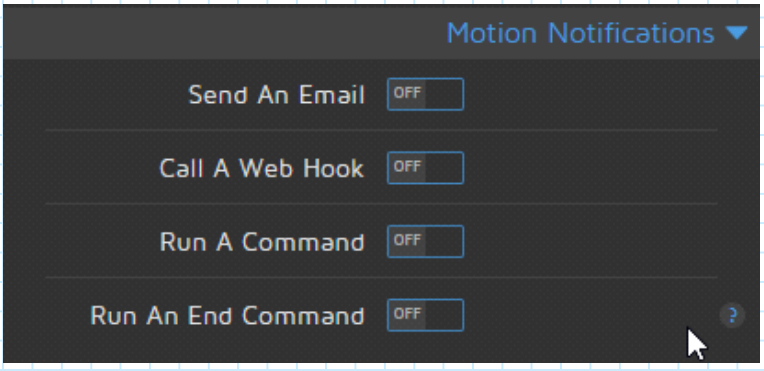
- Not enabled

Movies

- Not enabled

Motion Detection

- Use Not enabled



Motion Notifications

- Not configured

Working Schedule

- Not enabled

motioneye

Thursday, January 10, 2019 10:03 AM

Upgrade:
pip install --upgrade motioneye

From <https://github.com/ccrisan/motioneye/wiki/Installation>

Install On Raspbian

Calin Crisan edited this page on Apr 7 · 29 revisions

Before Proceeding

- Read the general Installation page first.
- These instructions apply only to an up-to-date Raspbian Stretch.
- All commands require *root*; use sudo before each command or become root using sudo -i.
- If you want to use the CSI camera module for the Raspberry PI, make sure you have enabled it in raspi-config.

Instructions

1. **Install ffmpeg and v4l-utils**:
   sudo apt-get install ffmpeg v4l-utils

   Note: v4l-utils appears to be preinstalled on Raspbian systems

2. **Install libmariadbclient18 and libpq5 required by motion**:
   sudo apt-get install libmariadbclient18 libpq5

3. **Install motion**:
   sudo wget https://github.com/Motion-Project/motion/releases/download/release-4.1.1/pi_stretch_motion_4.1.1-1_armhf.deb
   sudo dpkg -i pi_stretch_motion_4.1.1-1_armhf.deb

   Note: Raspbian Stretch comes with motion version 4.0; it is however recommended that you install version 4.1

4. **Install the dependencies from the repositories**:
   sudo apt-get install python-pip python-dev libssl-dev libcurl4-openssl-dev libjpeg-dev libz-dev

5. **Install motioneye, which will automatically pull Python dependencies (tornado, jinja2, pillow and pycurl)**:
   sudo pip install motioneye

6. **Prepare the configuration directory**:
   sudo mkdir -p /etc/motioneye
   sudo cp /usr/local/share/motioneye/extra/motioneye.conf.sample /etc/motioneye/motioneye.conf

7. **Prepare the media directory**:
   sudo mkdir -p /var/lib/motioneye

8. **Add an init script, configure it to run at startup and start the motionEye server**:
   sudo cp /usr/local/share/motioneye/extra/motioneye.systemd-unit-local /etc/systemd/system/motioneye.service
   sudo systemctl daemon-reload
   sudo systemctl enable motioneye
   sudo systemctl start motioneye

9. **To upgrade to the newest version of motionEye, just issue**:
   sudo pip install motioneye --upgrade
   sudo systemctl restart motioneye

Default:
user: admin
password: (blank)

http://192.168.1.104:8765/
http://192.168.4.1:8765/

From <https://github.com/ccrisan/motioneye/wiki/Install-On-Raspbian>

USB Storage

Saturday, October 20, 2018 5:25 PM

https://blog.alexellis.io/attach-usb-storage/

Step 1. Identify the drive
The easiest way to identify your drive is to only plug one in at a time. The Linux command lsblk will list all bulk storage devices.

```
lsblk
```

Step 2. Create the partitions
We will now use the fdisk tool to wipe out the existing partitions and create new ones. This is not always essential but ensures we have a known state on the disk.

```
Enter 'd' to delete partition if one exists.

sudo fdisk /dev/sda
```

Step 3. Format the new partition
You can check that the new partition was created by typing in:

```
sudo fdisk -l /dev/sda1
```

We'll now format the partition we just created and at the same time attach a unique label.

```
sudo mkfs.ext4 -L MOTIONEYE /dev/sda1     <- for ext4
sudo mkfs.vfat -F32 /dev/sda1 <- FAT32
sudo mkfs.ntfs -f  -L MOTIONEYE /dev/sda1     <- for ntfs       fyi:  without the '-f' formatting is a very slow process, 64GB ~4.0 hrs. Use Windows instead for format NTFS.
```

Step 4. Pick a mount-point
For a Linux operating system we need to pick a directory to mount our storage under. This could be almost any folder including /home/pi for instance. We'll keep things simple and pick /mnt/motioneye.

```
sudo mkdir /mnt/motioneye
```

Let's test the mount point temporarily and then make things permanent.

```
sudo mount -L MOTIONEYE /mnt/motioneye
ls /mnt/motioneye/
   lost+found
```

Step 5. Make it permanent
If you intend on using the drive permanently then follow this step. The next step involves editing the fstab file which is used to mount disks at system boot time:

```
sudo nano /etc/fstab
```

Since we used an ext4 file-system and that is also used for the SD card's root filesystem, we can use the same settings. You should see two lines like the following:

```
proc            /proc           proc    defaults          0       0
/dev/mmcblk0p1  /boot           vfat    defaults          0       2
/dev/mmcblk0p2  /               ext4    defaults,noatime  0       1     <- for ext4
/dev/mmcblk0p2  /               ntfs    defaults,noatime  0       1     <- for ntfs
```

Add a line underneath, save the file, then reboot:

```
LABEL=MOTIONEYE /mnt/motioneye ext4    defaultsj,nofail,noatime  0   1    <- for ext4
LABEL=MOTIONEYE /mnt/motioneye ntfs    defaultsj,nofail,noatime  0   1    <- for ntfs
LABEL=MOTIONEYE /mnt/motioneye vfat defaultsj,nofail,noatime  0   1    <- for fat32
```

Now that your drive is attached let's look at how to keep the drive working reliably and a few uses for that extra storage.

Note: 'nofail' above prevents device from not booting if usb drive not present.

Best practices

- Always shutdown with shutdown -h 0 or halt -h - never pull the power cable.
- If you are using the drive only temporarily then type in sudo umount /mnt/motioneye before pulling out the USB cable - or shutdown the system first.
- If you have a power-cut or accidental power-out then you can repair the filesystem like this:

```
$ sudo systemctl stop motioneye
$ sudo umount /mnt/motioneye
$ sudo fsck /dev/sda
      fsck from util-linux 2.25.2
      e2fsck 1.42.12 (29-Aug-2014)
      MOTIONEYE: clean, 11/19169280 files, 1251934/76669184 blocks
$ sudo mount /mnt/motioneye
$ sudo systemctl start motioneye
```

Change partition from ext4 to ntsf

1. sudo systemctl stop motioneye
2. sudo umount /mnt/motioneye
3. sudo umount /dev/sda1
4. sudo apt install ntfs-3g
5. Perform steps on left specifying **ntfs**. For steps #2 and #3, Windows Disk Management can be used.
   - Delete volume.
   - Create volume.
     - Volume Name: MOTIONEYE
     - Quick Format
6. sudo reboot
7. Validate
   - Verify http://192.168.4.1:8765/ is display correct storage size under File Storage.
   - Verify Windows is able to read USB stick.

From <https://blog.alexellis.io/attach-usb-storage/>

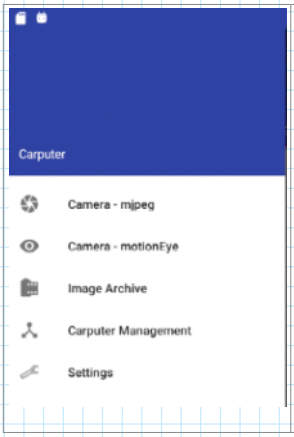
User Interface

Sunday, January 20, 2019 7:53 AM

Carputer Settings

The **Carputer** navigation is via the app's main navigation menu or via the navigation drawer.

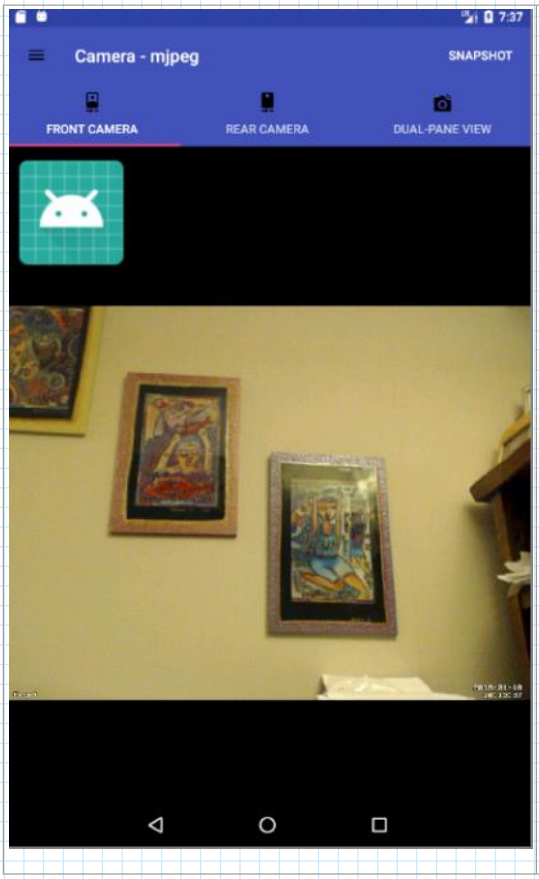
Navigation Drawer/Main Menu



The main menu and navigation drawer (opens from left edge) contains the following menu items.

Camera - mjpeg	Streaming video of front and rear cameras. Contains option menu, Snapshot, to capture image.
Camera - motionEye	Web view of **motionEye** video streaming and admin console.
Image Archive	List of snapshot images. Ability to delete images with a single tap and confirmation.
Carputer Management	View for SSH commands to the Raspberry Pi and web view of **phpSysInfo** running on the Pi.
Settings	Camera and Raspberry Pi node configuration settings.

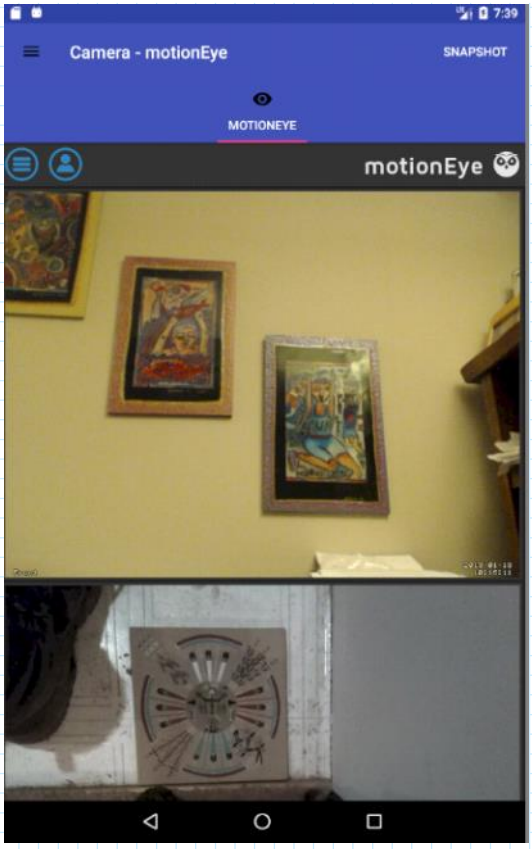
Camera - mjpeg



This view contains a tab layout that displays the mjpeg streaming url from motionEye. The view's and url's are configured via the settings menu item.

Front Camera	Displays front camera streaming video, if enabled. Has ability to capture **Snapshot** of current frame and will display this in thumbnail view.
Rear Camera	Displays rear camera streaming video, if enabled. Has ability to capture **Snapshot** of current frame and will display this in thumbnail view.
Dual-Pane View	Displays both front and rear camera's streaming video in a vertical layout, if enabled. Does not support the **Snapshot** feature.
Snapshot	Options menu that when clicked will capture the frame currently being displayed. The dual-pane view does not currently support this **Snapshot** feature. This is due to poor performance with the mjpeg widget used to display the streaming video. **Note**: With the Raspberry Pi image, motionEye is configured to capture still images and movie with the **Capture Mode** set to '**Motion Triggered**'. The **Preserve Pictures** setting is configured '**For One Week**'.

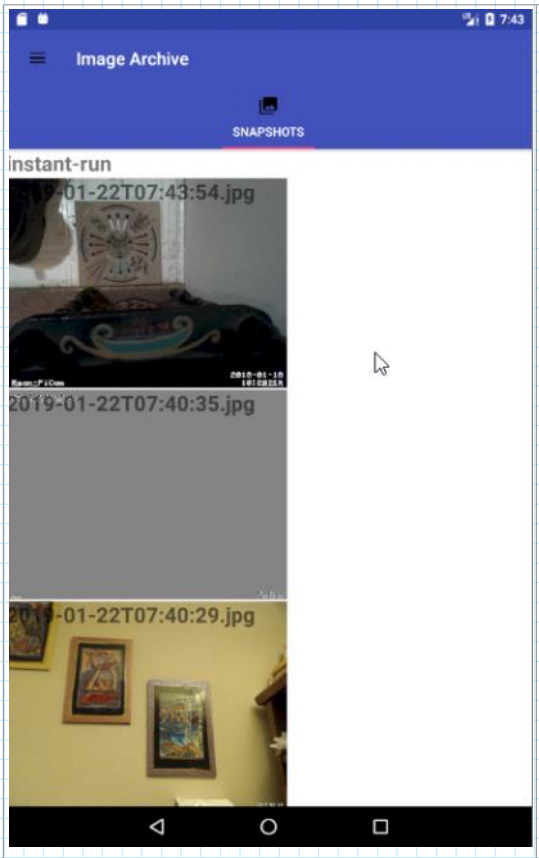
Camera - motionEye



This view displays the motionEye admin console web page along with the streaming video of the configured cameras.

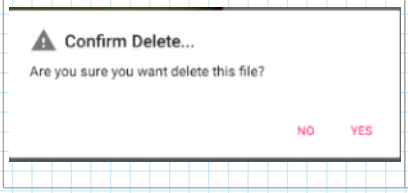
motionEye	Default view that displays the configured camera's in a vertical layout. The URL is configured via **Settings -> motionEye**.
Snapshot	Options menu that when clicked will capture the frame currently being displayed. The dual-pane view does not currently support this **Snapshot** feature. This is due to poor performance with the mjpeg widget used to display the streaming video. **Note**: With the Raspberry Pi image, motionEye is configured to capture still images and movie with the **Capture Mode** set to '**Motion Triggered**'. The **Preserve Pictures** setting is configured '**For One Week**'.
	Click on this icon to authenticate in the motionEye admin console.
	motionEye admin console.

Image Archive

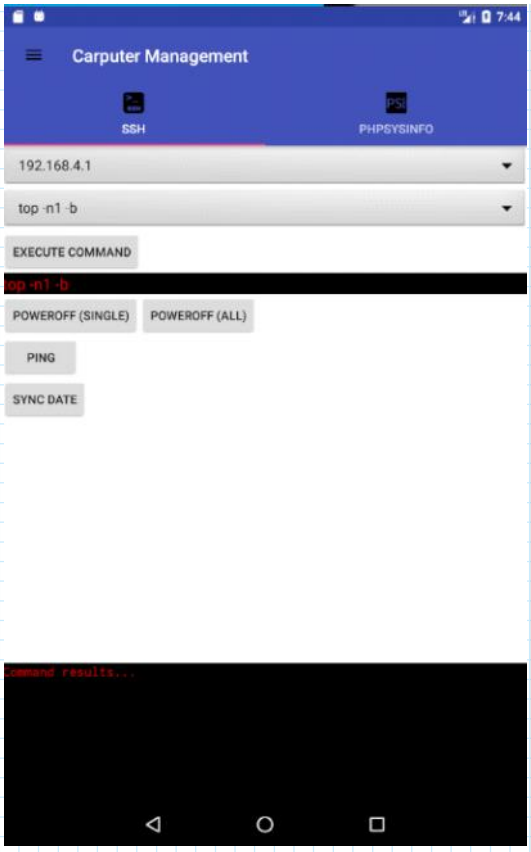


View that contains a vertical list of snapshots. Image contains timestamp of when the frame was captured. To delete image just tap and image and confirm delete.

Snapshots	View that contains a vertical list of snapshots. Image contains timestamp of when frame was captured.
Confirm Delete	To delete image just tap and image and confirm delete.



Carputer Management

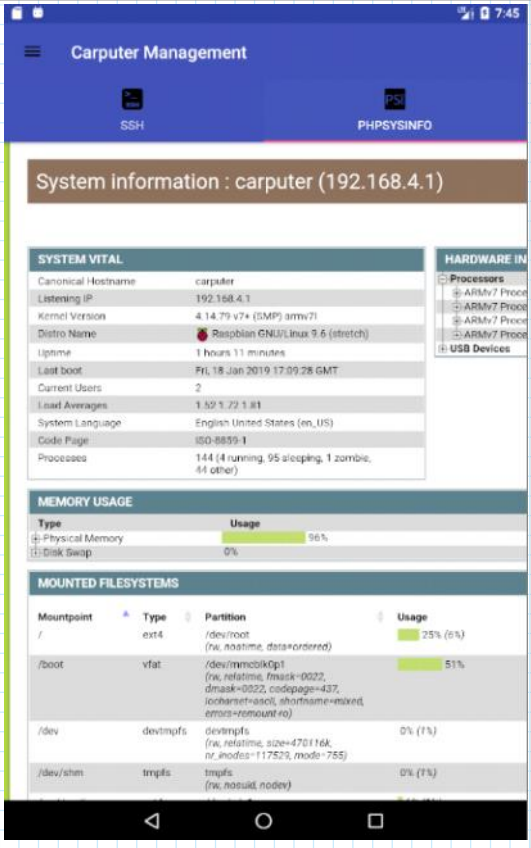


Web view of the phpSysInfo systems monitor. Not sure if it adds much value, but hey it looks cool.

SSH	This view is a complete hack and plans are in place for improvement.

First drop down	List of Raspberry Pi Ip's.
Second drop down	List of hard coded Linux command that might be useful.
Execute Command	When clicked will run the command that is display in the command text field.
Command Text	Editable text view that displays the details of the command selected from the drop down. User has the ability to enter commands not contain in the list. Command is then sent when the **Execute Command** button is clicked.
Poweroff (Single)	Will send a "**sudo shutdown -h 0**" to the Raspberry Pi ip selected from the drop down list.
Poweroff (All)	Will send "**sudo shutdown -h 0**" to **all** the Raspberry Pi devices.
Ping	Sends a '**ping**' command to ip of the selected Raspberry Pi.
Sync Date	Syncs the of the Android device with the selected Raspberry Pi. This needs to be performed since the Raspberry Pi does not have a **Real Time Clock**. Will return the value of the Raspberry Pi's configured date/time as confirmation.
Reply	Displays the output of the selected command.

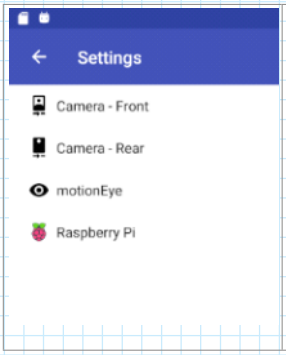
phpSysInfo	Web view of the phpSysInfo systems monitor. Enabled via **Settings -> Raspberry Pi - phpSysInfo**.



Web view that displays the phpSysInfo monitoring web page. The data is automatically refreshed.

phpSysInfo	Web view that displays the phpSysInfo monitoring web page. The data is automatically refreshed.

Settings



Please refer to Deployment Instructions -> Android -> Settings for details on settings configuration.

Camera - Front	Configuration settings for the front camera.
Camera - Rear	Configuration settings for the rear camera.
motionEye	Configuration settings for the motionEye application.
Raspberry Pi	Configuration settings for the main Raspberry Pi node.

Version History

Thursday, May 9, 2019 2:20 PM

Component	Version	Release Date
This document		
	1.6.1	24Jun2022
	1.6	18Apr2021
	1.5	28Nov2019
	1.4	4Jul2019
	1.3	17May2019
	1.2	25Apr2019
	1.1	25Apr2019
	1.0	28Jan2019
Android		
	1.5.3	9Nov2019
	1.5.2	25Apr2019
	1.5.1	20Apr2019
	1.5	8Apr2019
	1.4.1	4Apr2019
	1.4	31Mar2019
	1.3.1	15Mar2019
	1.3	14Mar2019
	1.2	12Mar2019
	1.1	5Mar2019
	1.0	28Jan2019
Hardware		
	1.2	18Apr2021
	1.1	25Apr2019
	1.0	28Jan2019
Firmware – Master		

	1.6.1	24Jun2022
	1.6	18Apr2021
	1.5	28Nov2019
	1.4	4Jul2019
	1.3	17May2019
	1.2	24Apr2019
	1.1	24Apr2019
	1.0	28Feb2019
Firmware – Slave		
	1.2.1	9Jan2022
	1.2	28Nov2019
	1.1	24Apr2019
	1.0	28Feb2019
Firmware – Python and Shell Scripts		
juice4halt.py	1.0	18Apr2021
shutdown_script.sh	1.2.1	18Apr2021
heartbeat.py	1.0	28Jan2019
listen_for_shutdown.py	1.1	4Jul2019
	1.0	29Jan2019
heartbeat.sh	1.0	28Jan2019
listen_for_shutdown.sh	1.0	28Jan2019

timesync.sh	1.1.1	9Jan2022

Misc

Saturday, January 19, 2019 1:47 PM

Links

Saturday, June 23, 2018 3:02 PM

https://github.com/f1xpl/openauto

https://www.iotgadgets.com/2018/02/android-auto-comes-raspberry-pi-open-auto-project/

https://github.com/opencardev/crankshaft

https://blog.tremend.com/android-auto-car-infotainment-systems/

https://opensource.com/article/18/3/openauto-emulator-Raspberry-Pi

http://hackaday.com/2017/02/20/homemade-subaru-brz-head-unit-is-hidden-masterpiece/

https://www.autopi.io/

https://lembergsolutions.com/blog/how-guide-obdii-reader-app-development

OBD-II/CANBUS
Pyton parser: https://github.com/brendan-w/python-OBD
Python timer: https://stackoverflow.com/questions/474528/what-is-the-best-way-to-repeatedly-execute-a-function-every-x-seconds-in-python

http://android-er.blogspot.com/2015/05/play-stream-video-from-raspberry-pi-on.html

http://helloraspberrypi.blogspot.com/2015/05/stream-raspberry-pi-camera-module-video.html

Diagram

Thursday, May 9, 2019 7:37 PM

I like this case.