RaspVMC Installation

Table of Contents

1. Introduction:

This procedure explains how to install a software controller for Controlled Mechanical Ventilation (CMV) from Zendher on a Raspberry PI. Tested on “Luxe” Version of 450 and 550 models. This CMV is also known as WRT 950/960 or Storkair 450/550

Current procedure has been made with the Raspbian NOOBS image for Raspberry available at https://www.raspberrypi.org/downloads/

This procedure will place the Raspberry as the CMV controller. 2 devices will be able to control the speed of the CMV. (The Raspberry itself and the control unit from the CMVC known as Comfosense).

All this procedure is based on the work of JCoenen, a Belgian motivated guy, many thanks to him!

His github page for this project: https://github.com/jcoenencom/raspVMC

2. Syntax

All command of choices you have to do are written in this document with the “Courrier New” font and are in bold.

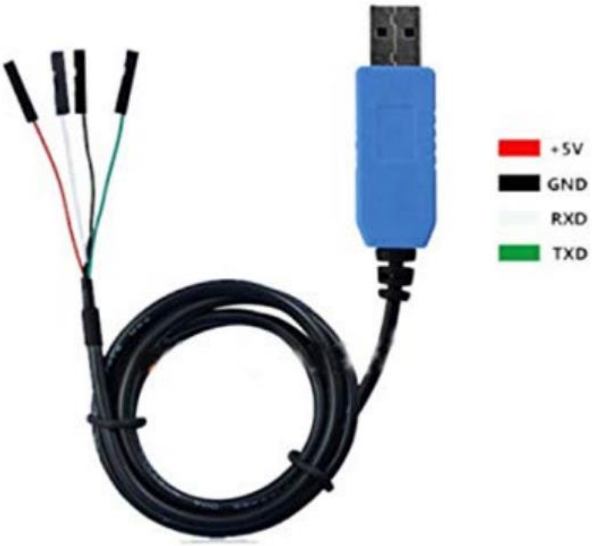
Example: ***`sudo reboot`***

3. Glossary

CMV	Controlled Mechanical Ventilation system
Raspberry	A micro computer board with all integrated (Graphic, RAM, CPU, LAN, etc.)
GPIO	(general-purpose input/output) This name of the PIN connector of the Raspberry
Raspbian	A dedicated linux version for the Raspberry board
RS232	Serial connection used to communicate information between device and computer
DB9	A standard electronic connector

4. Hardware needed

- 1 x Raspberry PI (each version)
- 1 x SD or Micro SD card depending your Raspberry version
- 1 x RS232 to TTL Converter with Male DB9 (3V to 5.5V)
- 1 x USB TTL cable
- Wire, solding material etc. to connect devices together

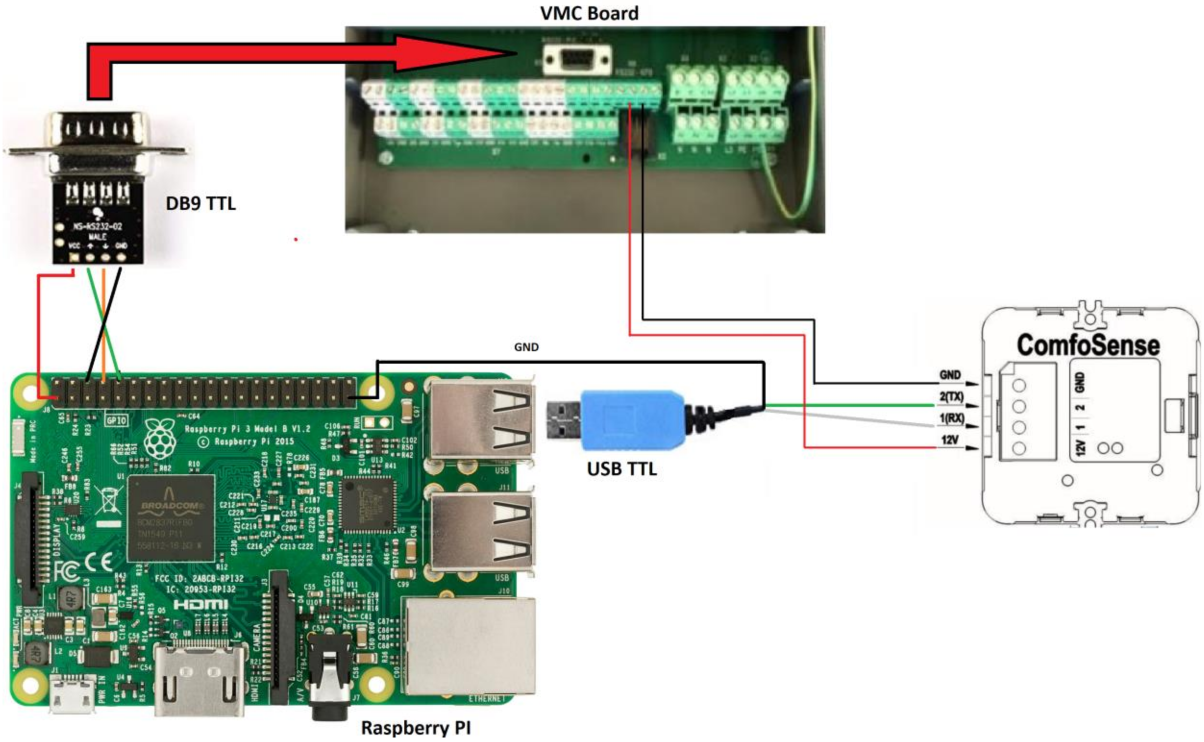
Example of a Raspberry PI 3	
Example of a RS232 to TTL Converter with Male DB9 (3V to 5.5V)	
Example of a USB TTL cable	

5. Connection schema

a. Global Schema

Here is an example to show you how you can connect all the devices together.

Note: Adapt the cabling with your configuration (Raspberry GPIO differ from version)



6. Installation

1. Install Raspbian Lite on a SD Card

   Refer to the Raspberry website

2. Enable SSH

   a. Enter **`sudo raspi-config`** in a terminal window
   b. Select **`Interfacing Options`**
   c. Navigate to and select **`SSH`**
   d. Choose **`Yes`**
   e. Select **`Ok`**
   f. Choose **`Finish`**

3. Update Raspbian

   a. **`sudo apt-get update`**
   b. **`sudo apt-get dist-upgrade`**

4. Disable AMA0 service

   a. Run the following commands to stop than disable the service

   ```
   sudo systemctl stop serial-getty@ttyAMA0.service
   sudo systemctl disable serial-getty@ttyAMA0.service
   ```

   b. Edit the CMDline.txt file to avoid the service start on boot

   ```
   sudo nano /boot/cmdline.txt
   ```

   - The file content looks like this
     dwc_otg.lpm_enable=0 ***console=serial0,115200*** *console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes root wait*
   - Remove this part: ***console=serial0,115200***

   c. Check the service has been disabled

   ```
   sudo systemctl status serial-getty@ttyAMA0.service
   ```

5. Install Apache2

   a. Install Apache 2

   ```
   sudo apt-get install apache2

   sudo a2enmod cgi
   ```

6. Install FHEM

    a. Login as root

    ```
    sudo su
    ```

    b. Add FHEM key

    ```
    wget -qO - http://debian.fhem.de/archive.key | apt-key add -
    ```

    c. Add FHEM source

    ```
    echo "deb http://debian.fhem.de/nightly/ /" >>
    /etc/apt/sources.list
    ```

    d. Update

    ```
    apt-get update
    ```

    e. Install FHEM

    ```
    apt-get install fhem
    ```

    f. Quit root mode

    ```
    exit
    ```

7. Install RaspVMC

   a. Run the following commands to download and install RaspVMC

```
cd $HOME
wget https://github.com/jcoenencom/raspVMC/archive/master.zip
unzip master.zip
cd raspVMC-master
./install.bash
```

The script will ask you if you want to install, answer Y or N (Case sensitive)
FHEM is not installed do you want it to be installed (Y/N) ? **N**

   b. Creation of the VMC service

```
sudo nano /etc/systemd/system/VMCserver.service
```

copy/paste the content below in this new file

```
[Unit]
Description=VMC python server
After=syslog.target network.target

[Service]
Type=simple
User=root
WorkingDirectory=/home/pi/raspVMC-master
ExecStart=/home/pi/raspVMC-master/server.py
StandardOutput=syslog
StandardError=syslog

Restart=always

[Install]
WantedBy=multi-user.target
```

Save the file with "`CTRL + O`" than quit the editor with "`CTRL + X`"

   c. Activation of the VMC Server service

```
sudo systemctl enable VMCserver
```

   d. Copy web file needed

```
cd /home/pi/raspVMC-master
cp *html /var/www/html
cp -r json* /var/www/html/
cp *.cgi /usr/lib/cgi-bin
cp VMC.py /usr/lib/python2.7
```

   e. Copy file for graphics in FHEM

```
cd /home/pi/raspVMC-master
sudo cp *.gplot /opt/fhem/www/gplot/
```

copy the FHEM config from raspVMC to FHEM

```
cd /home/pi/raspVMC-master
cat fhem.cfg
```

copy the content of the fhem.cfg file

```
sudo nano /opt/fhem/fhem.cfg
```

paste the copied content at the end of the file
Save the file with “**`CTRL + O`**” than quit the editor with “**`CTRL + X`**”

8. VMC.ini file

   Here is a sample of a VMC.ini file located in the folder “/home/pi/raspVMC-master”

```
[VMC]
device = /dev/ttyAMA0

[server]
bind =
port = 10001

[control]
port = 10002

[client]
server = 127.0.0.1

[socat]
pty = /tmp/ttyVMC

[debug]
log = /var/log/VMClog.log
level = 3
```

9. Disable FHEM looking for new USB devices

   FHEM could be really slow to start (15 minutes) because it tries to find everything that could be connected to the USB. In order to avoid this behavior, you have to comment a config line.

Open the fhem.cfg file

```
sudo nano /opt/fhem/fhem.cfg
```

Comment the following line by adding a # in front of the line (in red below)

```
# Disable this to avoid looking for new USB devices on startup
#define initialUsbCheck notify global:INITIALIZED usb create
```

Save the file and exit text editor

10. Automate second Socat for Comfosense

    In order to start this second Socat process, you need to edit the Crontab of the Root user

    Edit the crontab

    ```
    sudo crontab -e
    ```

    Add this line to the Crontab

    ```
    @reboot sleep 60 && socat /dev/ttyUSB0,raw,echo=0,b9600 tcp4-connect:127.0.0.1:10001 &
    ```

11. Reboot and check

    Reboot the system with command:

    ```
    sudo reboot
    ```

    Then check everything is working and you have your 2 Socat process with command:

    ```
    ps -ef | grep socat
    ```