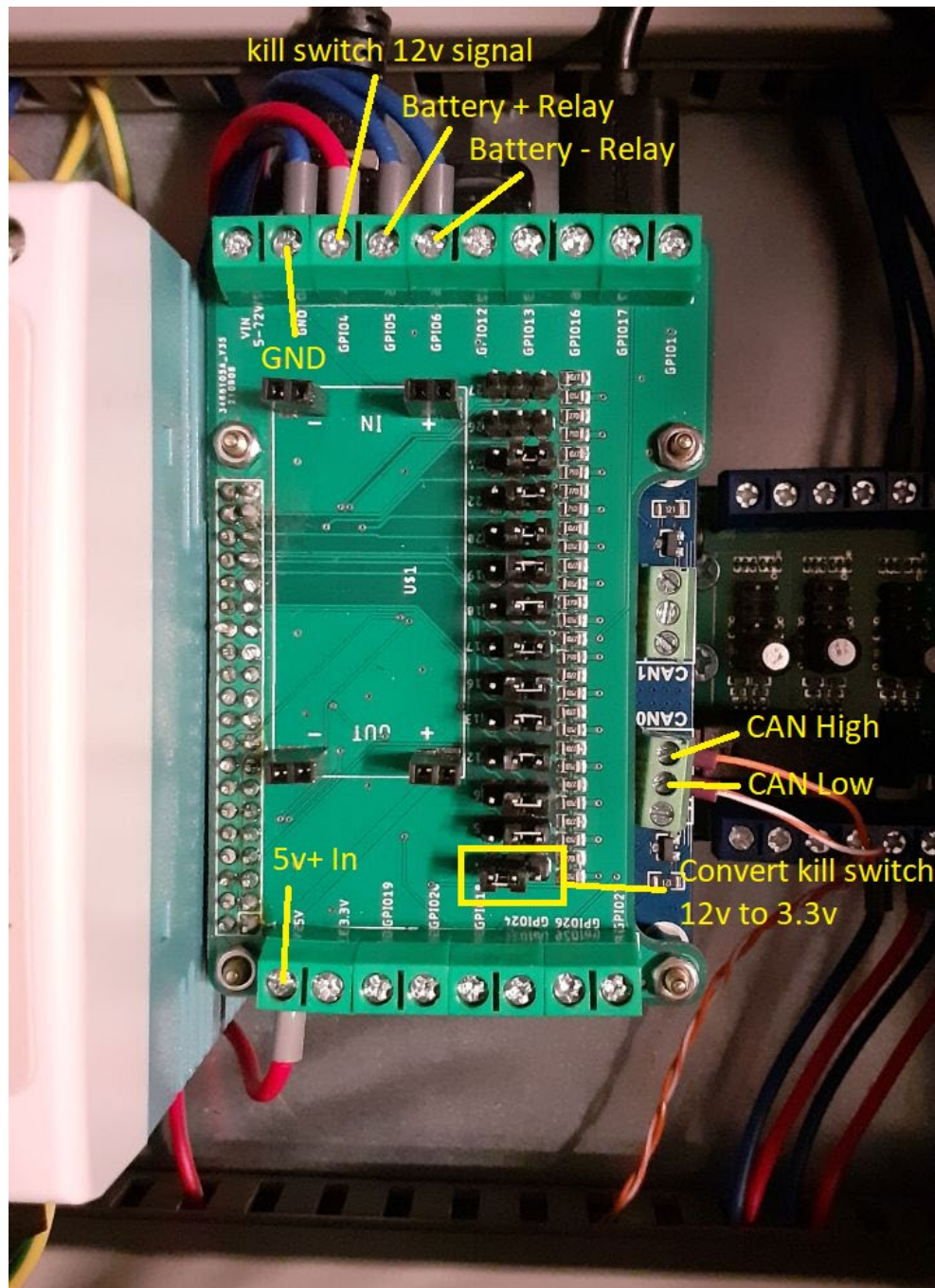


1. Prerequisites and Wiring

- Make sure your Raspberry Pi is correctly wired and the jumpers on the GPIO shield are correctly set. Wrong jumper positions can destroy your Raspberry Pi, just like wrong wiring. Read https://github.com/SunshadeCorp/GPIO-Adapter/raw/main/documentation/manual_en.pdf for more information on how to use the gpio shield.
- Look up how to configure and wire the Economizer LT Board: https://github.com/jontubs/EconomizerLT/blob/main/EconomizerLT_Documentation_EN_v003.pdf



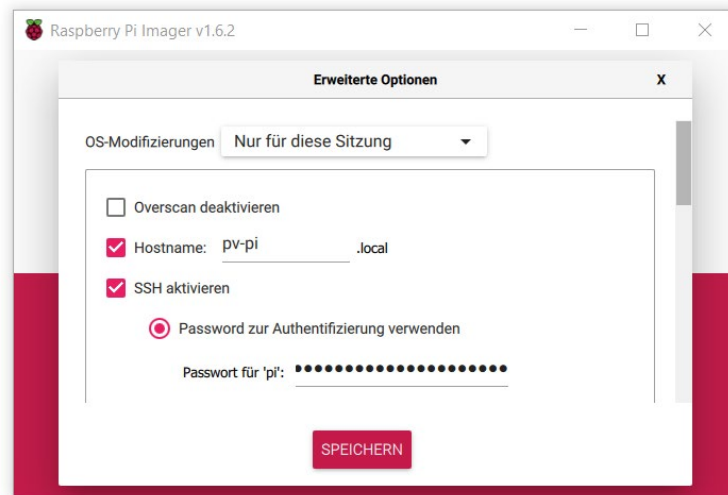
- Explanation
 - GND: Common ground of 12v and 5v power supplies
 - 12v+: This is the 12v signal from the emergency button. It is not used for power supply, but to detect whether the emergency button has been pressed.
 - Relay Control: These signals go to the Economizer board to control the relays.
 - 5v+: This is used as power supply for the Raspberry Pi.
 - Make sure all the jumper positions are correct. Notice that the last the jumper is positioned to the left.

2. Set Up a Base Raspberry Pi Image

- Download and install the Raspberry Pi imager from <https://www.raspberrypi.com/software/>
 - Note: Raspberry Pi Imager v1.7.1 was used in this manual.
- Insert the Micro SD Card into the computer.
 - Note: a SanDisk Extreme 64 GB Micro SDXC card was used in this manual.
- Start the Raspberry Pi Imager to setup the Raspberry Pi Image.



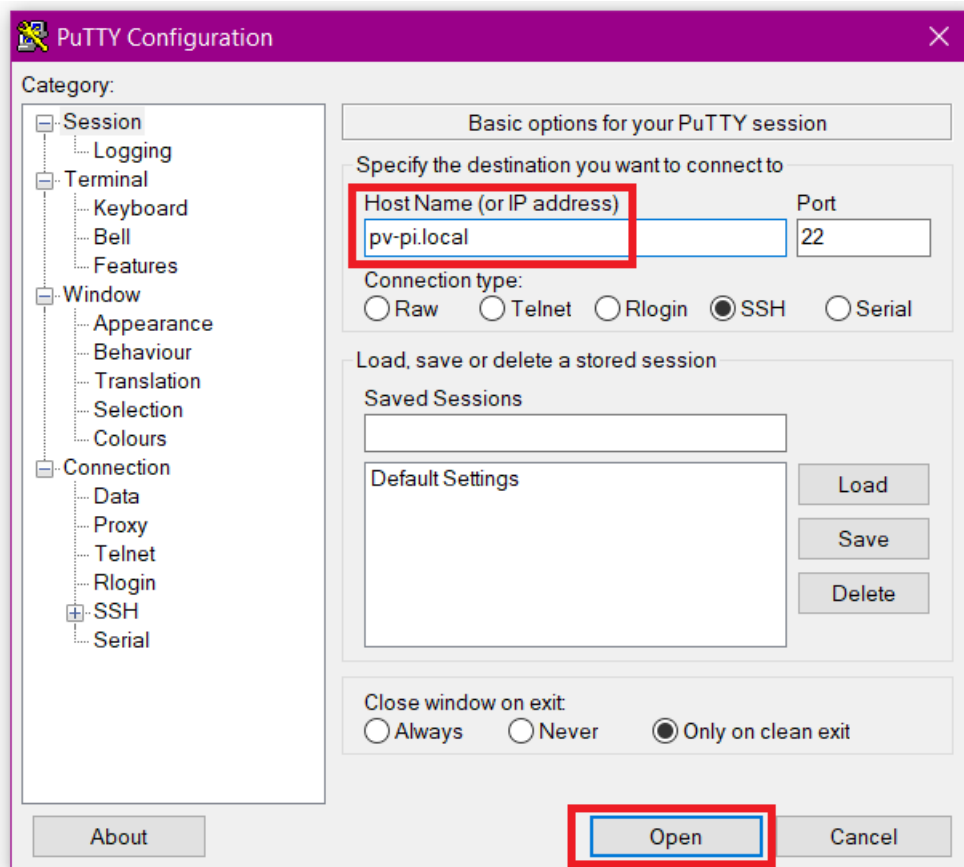
- Choose OS: Raspberry Pi OS (32-bit).
 - Note: Raspberry Pi OS Version 2021-01-28 was the default image at time of writing.
- Select SD card: Choose your SD card volume.
- Click on the gear symbol to change some image settings.
- A popup comes up, allowing you to change some options for your raspberry pi.



- Make the following changes:
 - Choose a hostname for your pi. In this example, the hostname “pv-pi” is used.
 - Enable SSH.
 - Choose your authentication method. In this case, password authentication was used. Do not use the standard credentials (user: pi; password: raspberry) because that will trigger a warning on every startup.
 - If you want, you can set up the Wi-Fi credentials for a development connection. However, a wired Ethernet connection is recommended and Wi-Fi will not work well inside the installation case.
 - Choose your appropriate country, language and keyboard settings.
 - Click ‘Skip Setup Assistant’
- Select “Save”.
- Back in the main screen, click on “write” now and confirm.
- The Raspberry Pi image is now written to the micro SD card. Windows might offer you to reformat the SD card after the imager is finished. Do not do that.
- After the image is written and verified, you can remove the SD card.

3. SSH into your Raspberry Pi

- Insert the micro SD card into your Raspberry Pi and plug in your power supply.
- We now want to get access to a command line on your pi. In order to do this, we are going to use an SSH connection. Use your favourite tool to establish an SSH connection to the Pi. In this case, we are using Putty on Windows. (<https://www.putty.org/>)



- Enter the host name that you configured in Raspberry Pi Imager and add the suffix “.local”. Click on open.
- Accept the SSH fingerprint.
- You will now be asked to input your user credentials for the Raspberry Pi.
- Enter the credentials that you configured in Raspberry Pi Imager. After that, you should be able to use the command line on your Raspberry Pi.

4. Create an ext4 Partition on your USB Storage Device

- Insert your USB storage device. We are going to do a clean reformat and install an ext4 partition so any existing data is going to be lost. If you already have an ext4 partition at /dev/sda1, you can skip this step.
- Switch to root user.

```
$ sudo su root
```

- The control unit stores measurement data on a separate storage device, such as an SSD. To be able to use your storage device it needs to be mounted. First, we need to create a partition on your device.
- Format your USB device and create an ext4 partition using fdisk. **TODO: Explain this in detail**

```
$ fdisk /dev/sda
new partition table:
g
new partition:
n
print table:
p
write:
w
$ umount /dev/sda1
$ mkfs.ext4 /dev/sda1
```

5. Install EasyBMS-master

- Make sure you are acting as root user.

```
$ sudo su root
```

- Clone the control-pi-docker repository to the /docker directory.

```
$ git clone https://github.com/SunshadeCorp/control-pi-docker.git /docker
$ cd /docker
```

- Check out the next-gen branch to be able to use the install script.

```
$ git checkout next-gen
```

- Execute the install script. The script installs easy-bms on your Raspberry Pi and all of its requirements. The script will ask you to choose the password for the MQTT connections in your system, as well as passwords for the MariaDB users 'homeassisstant' and 'root'. Remember these, because the BMS slaves will have to use the same credentials.

```
$ ./install.sh
```

- If you want to use a specific branch in any of the sub repositories, then now go check these branches out inside the build directory.

6. Configure the EasyBMS Slave Mapping

- Edit the slave mapping for the BMS master according to your configuration.

```
$ nano /docker/build/easybms-master/slave_mapping.yaml
```

- Let us take a look at the example slave mapping file:

```
slaves:
  aabbccddeeff:
    number: 1
    total_voltage_mesurer: true
  bbccddeeffaa:
    number: 2
  ccddeeffaabb:
    number: 3
  ddeeffaabbcc:
    number: 4
    total_current_mesurer: true
```

- Edit this example mapping according to your own installation. Under the top level ("slaves:"), the MAC-addresses of the BMS slaves are listed. Find out which MAC-addresses your BMS slaves have and create an entry for each of them.
- Then, there are three additional properties that need to be configured. The number of each slave exists to be able to identify a particular BMS slave in your battery cabinet. Mark each of your batteries and its BMS with a number starting from 1 and then enter this number in this file with the corresponding MAC-address.
- In your battery cabinet, there should be one BMS slave that takes an additional role as a system voltage measurement unit. For this BMS slave, add the line "total_voltage_mesurer: true".
- In the same way, there should be a current measurement unit. Add "total_current_mesurer: true" for this BMS slave.
- You are done when you have configured an entry for each of your BMS slaves. Save the file and continue to the next step.

7. Configure the MQTT Connection to the SH10RT Inverter

- Edit the configuration of Modbus4Mqtt:

```
$ nano /docker/modbus4mqtt/sungrow_sh10rt.yaml
```

- In the first line of the file, the IP of the sungrow inverter in the network is listed. Find out what the actual IP of your Sungrow inverter is and insert the IP address here.

```

GNU nano 5.4 /docker/modbus4mqtt/sungrow_sh10rt.yaml
ip: 192.168.178.66
port: 502
update_rate: 2
address_offset: 0
variant: sungrow
scan_batching: 100
registers:
  # # - #pub_topic: "date_time/minutes"
  # #   address: 5003
  # # - #pub_topic: "date_time/seconds"
  # #   address: 5004
  - pub_topic: "start_stop"

```

- Save the file

8. Startup & Update

- You can now start the EasyBMS-master and its services using the startup script. The first startup will take much longer because the containers are being downloaded and built. This might take a while. Sometimes, one of the servers the docker images are downloaded from might not be available. In this case, just try again.

```
$ ./start.sh
```

- The default password for the homeassistant 'admin' user is 'password'. Please change the default password after your first login.
- If you want to perform an update of Easy-BMS, you can use the update.sh script to do that.

```
$ ./update.sh
```

9. Optional: Set Up Remote Access via VPN

- First, you need to have a server with OpenVPN installed. For this guide, we assume that you already have an OpenVPN server set up. If not, you can check out https://github.com/SunshadeCorp/documentation/raw/475901a863550dba19a38d80ef497a333fcb0cdf/OpenVPN%20Server%20Setup/openvpn-server-setup_en.pdf on how to set up a simple OpenVPN server.
- The OpenVPN client is automatically installed on your Raspberry Pi in the install script.
- The maintainer of your OpenVPN server should provide you a client configuration file. This file contains the keys required to log in to the VPN server
- Client configuration files for OpenVPN come as ovpn-files or conf-files. We are going to need a conf file, so rename your file to .conf if required.

```
$ mv vpnclient.ovpn vpnclient.conf
```

- Copy your configuration file to /etc/openvpn

```
$ cp vpnclient.conf /etc/openvpn/
```

- Add the executable permission to update-resolv-conf

```
$ sudo chmod +x /etc/openvpn/update-resolv-conf
```

- Initiate the VPN connection. On reboot, OpenVPN is going to reconnect automatically.

```
$ ...
```

10. Optional: Set Up a Telegram Bot

11. Useful Links

- How to generate a public private key pair
<https://www.ssh.com/academy/ssh/keygen>
- How to mount a usb stick or an SSD on Raspberry Pi
<https://jankarres.de/2013/01/raspberry-pi-usb-stick-und-usb-festplatte-einbinden/>
- Information on fstab and mount options
<https://wiki.debian.org/fstab>
- How to install docker on Raspberry Pi
<https://dev.to/elalemnyo/how-to-install-docker-and-docker-compose-on-raspberry-pi-1mo>
- How to set up a service to execute a shell script using system
<https://tecadmin.net/run-shell-script-as-systemd-service/>