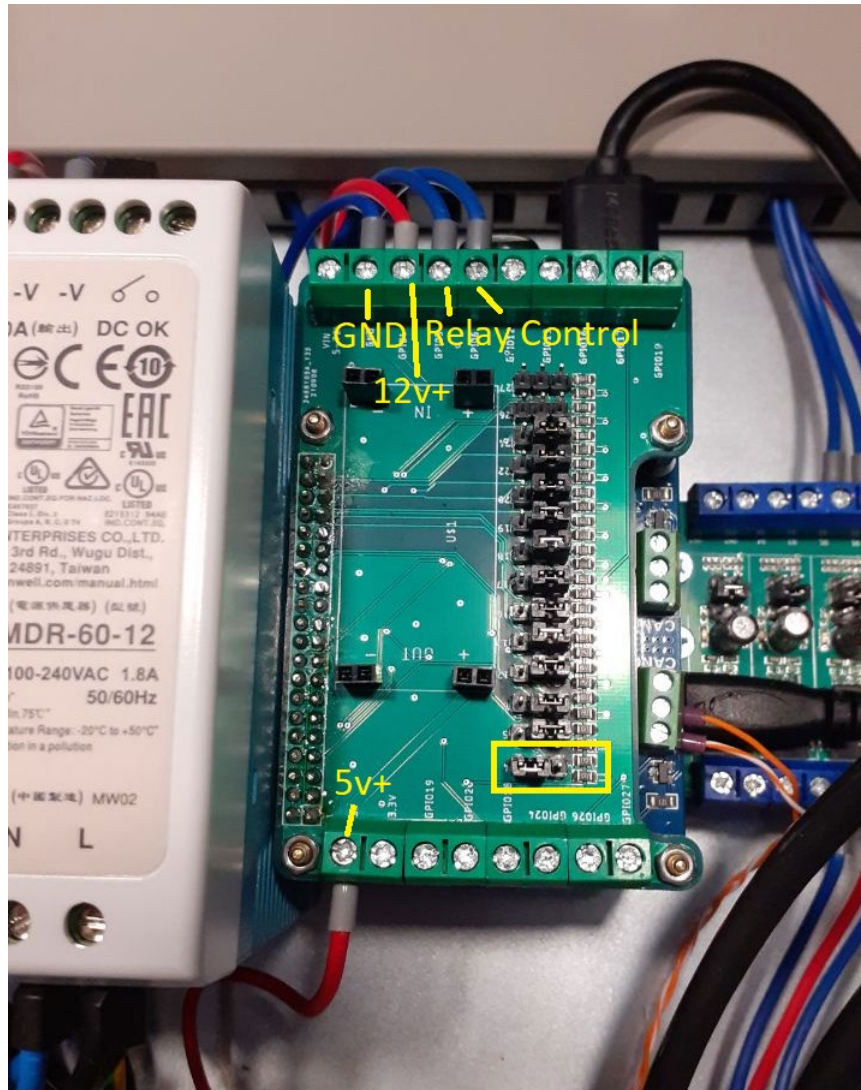


1. Prerequisites and Wiring

- 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.



- 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 control 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.

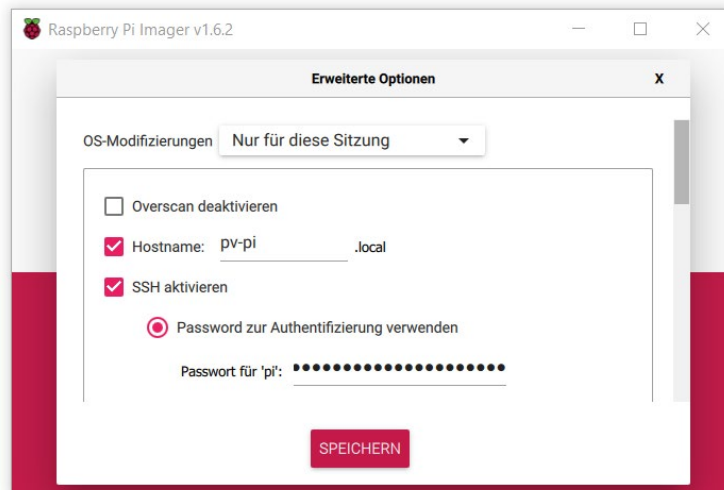
2. Set Up a Base Raspberry Pi Image

- 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.
- Skip Setup Dialog: Check.
- 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. Set Up SSH Keys and Git

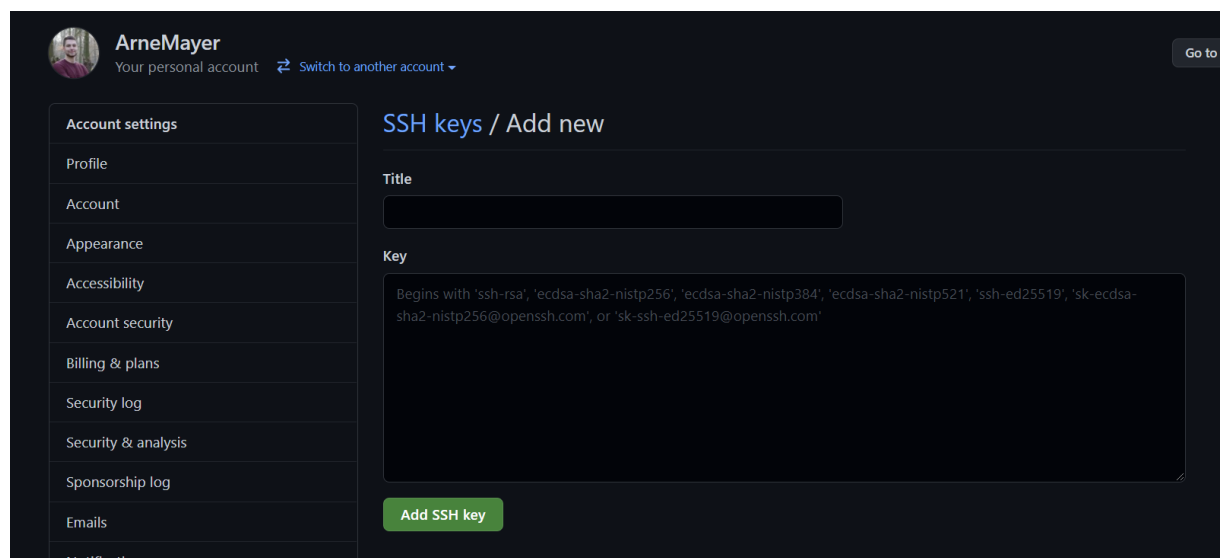
- Generate SSH keys for the root user. Leave the default location for the key and use a passphrase if you want. In this manual, a passphrase is not used.

```
$ sudo su root
$ ssh-keygen
```

- Output the public SSH key onto the screen. The public key is not a secret. It will be used to grant your raspberry pi rights to clone the SunshadeCorp git repositories.

```
$ cat ~/.ssh/id_rsa.pub
```

- In your web browser, visit the ssh key configuration page of your github account.
<https://github.com/settings/keys>
- Click “New SSH key”. Choose a name for this key. Paste the output of the cat command into the key field and save. Your raspberry pi is now able to access the SunshadeCorp repositories.



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 git@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 install script 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 'homeassistant' 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 EasyBMS-master

TODO: Explain this in detail

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

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

7. 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.

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

- If you want to perform an update of easy-bms, you can use the update.sh script to do that.

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

8. Optional: Set Up Remote Access via VPN

9. Optional: Set Up a Telegram Bot

10. 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/elalemany/how-to-install-docker-and-docker-compose-on-raspberry-pi-1mo>