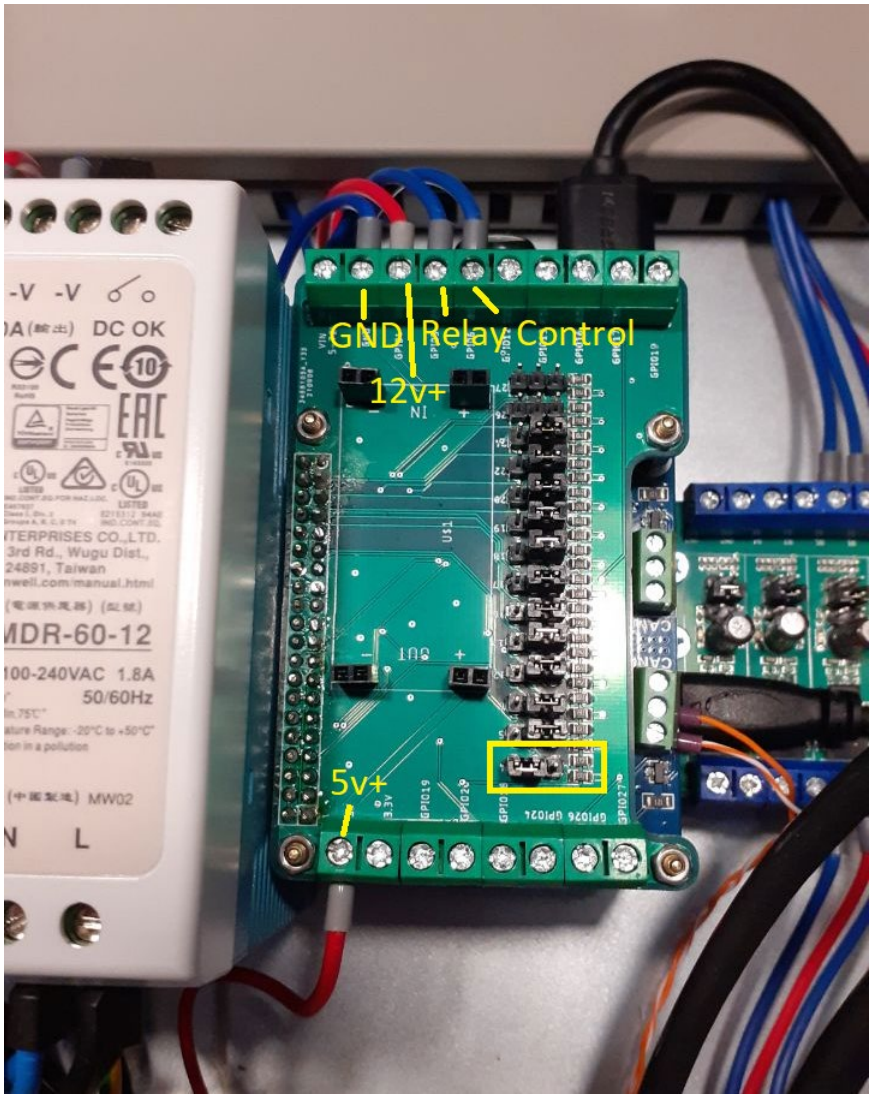
1. Prerequisites and Wiring

- 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.



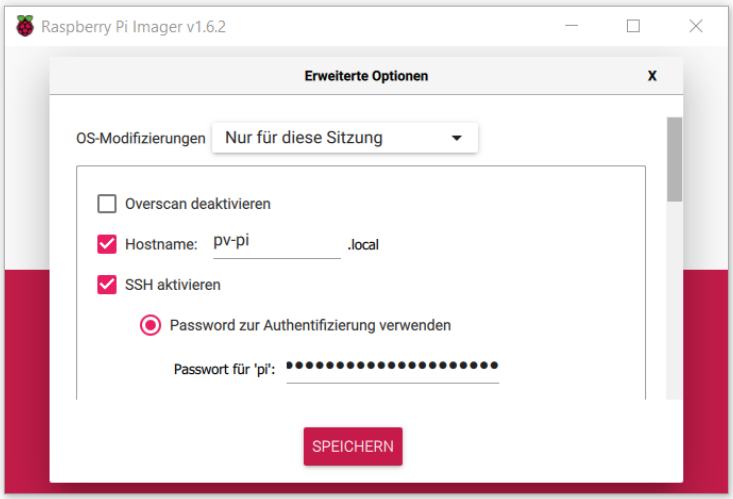
- Explanation
  - GND: Common ground of 12v and 5v power supplies
  - 12v+: This is the 12v signal from the emergency button. It is not used for power supply, but to detect whether the emergency button has been pressed.
  - Relay Control: These signals control go to the Economizer board to control the 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.
- Press Ctrl+Shift+X
- 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 manual, the hostname “pv-pi” is used.
    - Activate 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.

- 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. First Boot and Force HDMI

- Insert the SD card into the Raspberry Pi and power up the Pi.
- Open your SSH tool on PC and connect to your raspberry pi by using the hostname you previously defined. Enter your raspberry pi user credentials.
- Open up the boot config.txt

```
$ sudo nano /boot/config.txt
```

- Uncomment the following line and then save the file:

```
hdmi_force_hotplug=1
```

- Reboot the Raspberry Pi and connect the display.

```
$ sudo reboot
```

- You should now get a picture on your display. Enabling force HDMI averts a possible startup race between the raspberry pi and the display. Normally, the display would need to be powered on before the raspberry pi.

4. Mount your 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.
- 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
```

- Now we create the directory to mount the device to. Use the /mnt/ssd directory for this:

```
$ mkdir /mnt/ssd
```

- Use blkid to find the device id and path you want to mount.

```
$ blkid -o list -w /dev/null
```

```
root@pv-pi:/home/pi# blkid -o list -w /dev/null
device                  fs_type     label       mount point                 UUID
------------------------------------------------------------------------------------------------------------------------
/dev/mmcblk0p1          vfat        boot        /boot                       E183-6233
/dev/mmcblk0p2          ext4        rootfs      /                           1232a209-2596-48f0-a078-731d10b918ad
/dev/sda1               exfat                   /media/pi/96F2-953A         96F2-953A
root@pv-pi:/home/pi#
```

- In this case, the device I want to mount is /dev/sda1. Now we mount the device to /mnt/ssd. Create an entry in fstab to automatically mount the device on boot.

```
$ nano /etc/fstab
```

- Add the following line with the uuid from above: (adjust UUID according to your setup)

```
UUID=96F2-953A /mnt/ssd ext4 defaults,nofail 0 0
```

- Now mount the device using the mount command

```
$ mount –a
```

5. Set Up SSH keys and Git

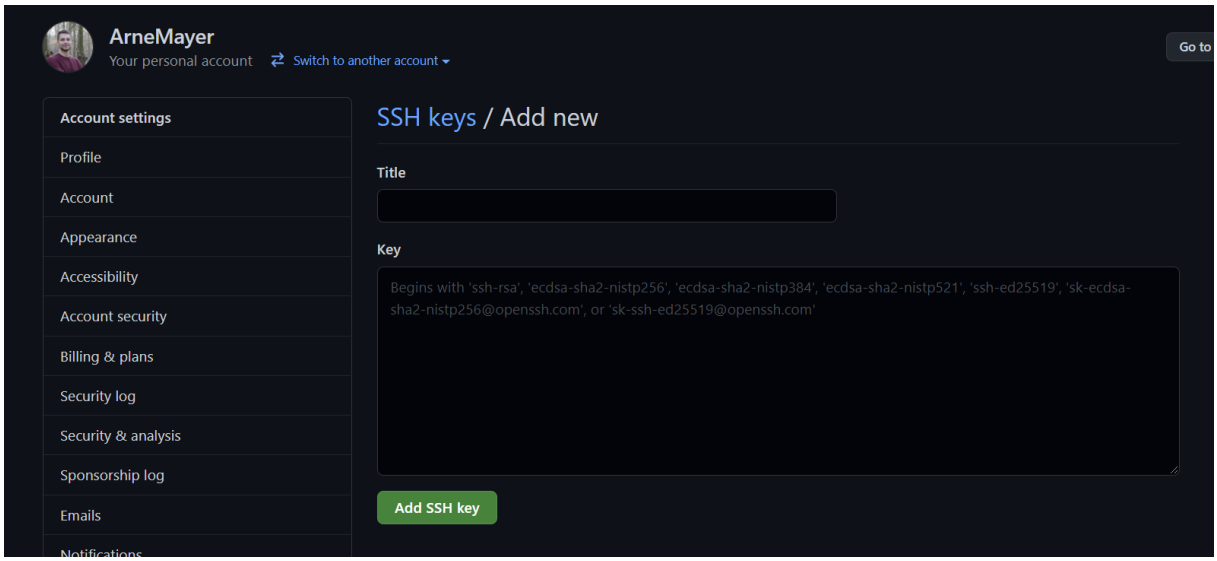
- Generate SSH keys for the root user. Leave the default location for the key and use a passphrase if you want. In this manual, a passphrase is not used.

```
$ 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.



6. 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
```

- If there is a specific branch you want to use, then get the branch using git checkout. In this case, the branch I want to use is called ‘install-script’.

```
$ git checkout install-script
```

- Execute the install script. The install script downloads and installs docker and docker-compose. Also it clones the rest of the SunshadeCorp repositories into its appropriate directories. The script will ask you to choose the log username and password for the mqtt connections in your system. Remember these because the BMS slaves 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.

7. 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
```

8. Startup

- You can now start the EasyBMS-master and its services with docker-compose. The first startup will take much longer because the containers are being downloaded and built. This might take a while. Remove the –d flag if you want to see the docker-compose log as the services are starting up.

```
$ cd /docker && docker-compose up -d
```

9. Optional: Set Up Remote Access via VPN

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/elalemanyo/how-to-install-docker-and-docker-compose-on-raspberry-pi-1mo

12. Notes

- In order to be able to pull branches, you may need to specify the default pull behavior. You can do this by typing:

```
$ git config pull.rebase false
```