

---

# **Dyno System Configuration Documentation**

***Release 0.0***

**Samuel Lindgren**

**Nov 15, 2019**



---

## Contents

---

<b>1</b>	<b>Stm32</b>	<b>3</b>
1.1	Udev rules . . . . .	3
1.2	Firmware Build Dependencies . . . . .	3
1.3	Visual Studio Code . . . . .	4
<b>2</b>	<b>Docker</b>	<b>5</b>
<b>3</b>	<b>CAN</b>	<b>7</b>
3.1	USB-CAN . . . . .	7
<b>4</b>	<b>C#</b>	<b>9</b>
4.1	.NET . . . . .	9
4.2	Monodevelop . . . . .	9
<b>5</b>	<b>Balena</b>	<b>11</b>
5.1	NVM . . . . .	11
5.2	balena CLI . . . . .	11
<b>6</b>	<b>Indices and tables</b>	<b>13</b>



Contents:



### 1.1 Udev rules

Copy *49-stlinkv2.rules* and *99-dfu.rules* from this repo to */etc/udev/rules.d*.

Reload udev rules by running:

```
sudo udevadm control --reload-rules && udevadm trigger
```

### 1.2 Firmware Build Dependencies

Install by running:

```
sudo add-apt-repository ppa:team-gcc-arm-embedded/ppa
sudo apt-get update
sudo apt-get install gcc-arm-embedded
sudo apt-get install openocd
sudo add-apt-repository ppa:jonathonf/tup && sudo apt-get update && sudo apt-get _
↪install tup
```

Install python 3.7:

```
sudo apt install python3.7-dev
```

Set python3.7 to default python version in bash:

```
echo "alias python=python3.7" >> ~/.bash_aliases
```

Verify successful installation and versions:

```
arm-none-eabi-gcc --version
arm-none-eabi-gdb --version
openocd --version      # should be 0.10.0 or later
tup --version           # should be 0.7.5 or later
python --version        # should be 3.7 or lat
```

## 1.3 Visual Studio Code

Download and install [Visual Studio Code](#).

Extensions:

- C/C++
- Cortex-Debug
- PlatformIO IDE



## CHAPTER 2

---

### Docker

---

Install docker by running:

```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu_
↳bionic stable"
sudo apt update
sudo apt install docker-ce
```

Check that installation is successful:

```
sudo systemctl status docker
```

Run docker without sudo:

```
sudo usermod -aG docker ${USER}
```

Restart the computer for the groups to update.

Install docker-compose by running:

```
sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-
↳compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```



### 3.1 USB-CAN

We at Dyno use two slightly different clones of the [Lawicel CANUSB](#) USB can interface.

- 70200 from Auvideo with [muCAN](#) firmware
- [CANable](#) with [cantact](#) firmware

Full instructions for setup and troubleshooting with socketcan on Linux can be found [here](#).

Here are the necessary steps in short form:

- Copy `90-slcan.rules` from [this repo](#) to `/etc/udev/rules.d/`.
- Copy the contents of the `slcan` folder in this repo to `/usr/local/bin` and make sure the scripts are executable.
- Run `sudo apt-get install can-utils` at in a terminal.
- Add `can`, `can_raw` and `slcan` to `/etc/modules` as a list (each goes in a separate line).
- Restart the computer.

---

**Note:** With this default setup, the speed of the CAN bus is set to 1000 Kbit/s

We also have a PCAN-USB interface.

---

To test that everything is working, connect an interface run `candump can0` in one terminal and `cansend can0 123#DEADBEEF`.

You should see the message you sent in the terminal where you ran `candump`.



# CHAPTER 4

C#

## 4.1 .NET

Install [.NET Core](#) for Linux.

```
wget -q https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
sudo add-apt-repository universe
sudo apt-get install apt-transport-https
sudo apt-get update
sudo apt-get install dotnet-sdk-2.2
```

Check that installation worked:

```
dotnet --version
```

Should version should be 2.2.x

## 4.2 Monodevelop

```
sudo apt install apt-transport-https dirmngr
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys_
↪3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF
echo "deb https://download.mono-project.com/repo/ubuntu vs-bionic main" | sudo tee /
↪etc/apt/sources.list.d/mono-official-vs.list
sudo apt update
sudo apt-get install monodevelop
```

Usage: Navigate to project folder in terminal and run `monodevelop <some_project>.csproj``



### 5.1 NVM

Balena CLI requires npm to install.

```
curl -sL https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh -o-  
↪install_nvm.sh  
bash install_nvm.sh  
source ~/.profile  
nvm install 10.16.0  
nvm alias default 10.16.0  
nvm use default
```

To get nvm working with sudo, use the following commands:

```
n=$(which node); \  
n=${n%/bin/node}; \  
chmod -R 755 $n/bin/*; \  
sudo cp -r $n/{bin,lib,share} /usr/local
```

### 5.2 balena CLI

Installing

```
npm install balena-cli -g --production --unsafe-perm  
sudo npm install balena-cli -g --production --unsafe-perm
```

Logging in

```
balena login
```

For local mode development tips, use [this link](#).





## CHAPTER 6

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`