Manual PiCar

Installation Manual

This installation manual will guide you step by step through the required software and the installation process.

Required Software

Before you can get started and connect to your PiCar you must install and configure your PC.

MATLAB/Simulink 2018a

To run the Simulink Example MATLAB 2018a or newer is required. It can be downloaded from the rzt homepage or your usual source.

You must make sure a C compiler is installed for your MATLAB installation. To check, enter

```
mex -setup
```

into your MATLAB command window. If it spits out an error, you probably don't have a compiler installed. Follow the MATLAB installation guide for the recommended compiler in the error message itself or search for "c compiler" in the MATLAB Add-On Explorer and choose *MinGW-w64 C/C++ Compiler*.

You can check if your installation was successful by entering the above command in the MATLAB command window again.

Next, you must install the MATLAB/Simulink support packages for Raspberry Pi. To do this, open your Add-On Explorer and search for "raspberry pi". The *MATLAB Support Package for Raspberry Pi Hardware* and the *Simulink Support Package for Raspberry Pi Hardware* are both required. Just click install in your Add-On Explorer.

PuTTY

Another required piece of software is PuTTY. PuTTY is a free and open-source SSH client which will be used to communicate and gain information about the Raspberry Pi in your PiCar.

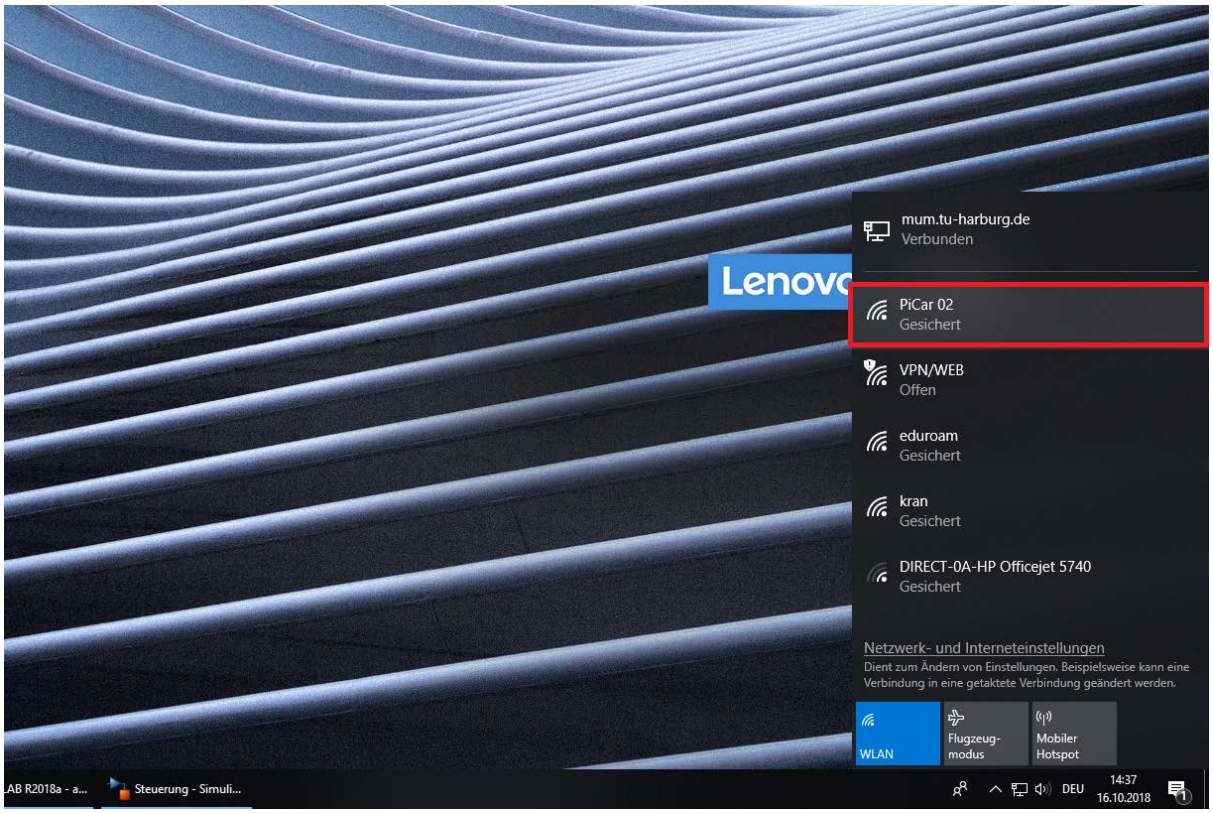
The software can be downloaded from this website:
https://www.chiark.greenend.org.uk/~sgtatham/putty/
but you can find it easily by typing "putty" in your preferred search engine. You should check for the latest version and your personal system requirements. Just download the installer and go through the installation process.

Getting Started

In this chapter you find a guide to connect to the PiCar and run the given Simulink example model.

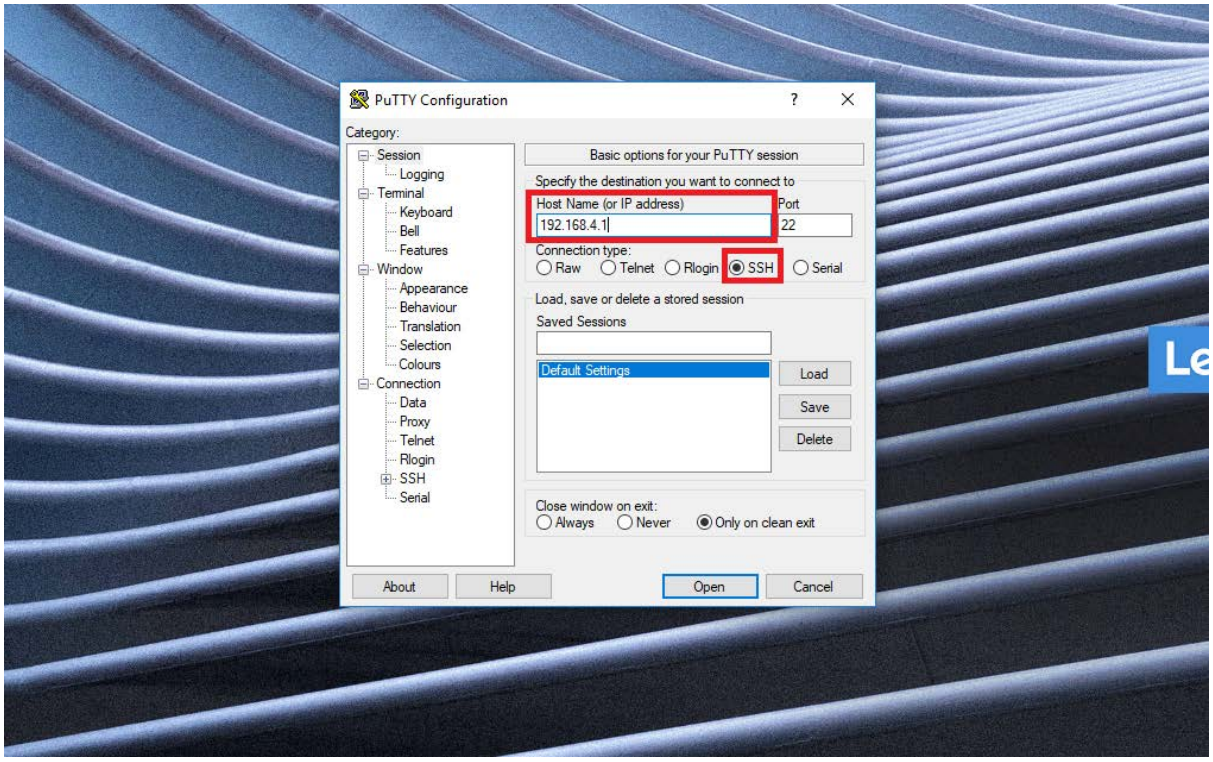
Connect To The PiCar

Now, that all the required software should be installed you can first try to connect to your PiCar. Turn on the PiCar by making sure the battery is connected and flipping the switch. Check your wireless networks. You should find a secured wireless network with the name “PiCar xx”. If it doesn’t show up immediately wait a few seconds and try again.



Connect to the network where “xx” stands for your car number. The WPA passphrase for all cars is “raspberry”. Now you can connect to your PiCar using PuTTY. Open the program, enter the IP address into the mask under the SSH tab like shown in the picture below and click “open”. The information is the same for all PiCars.

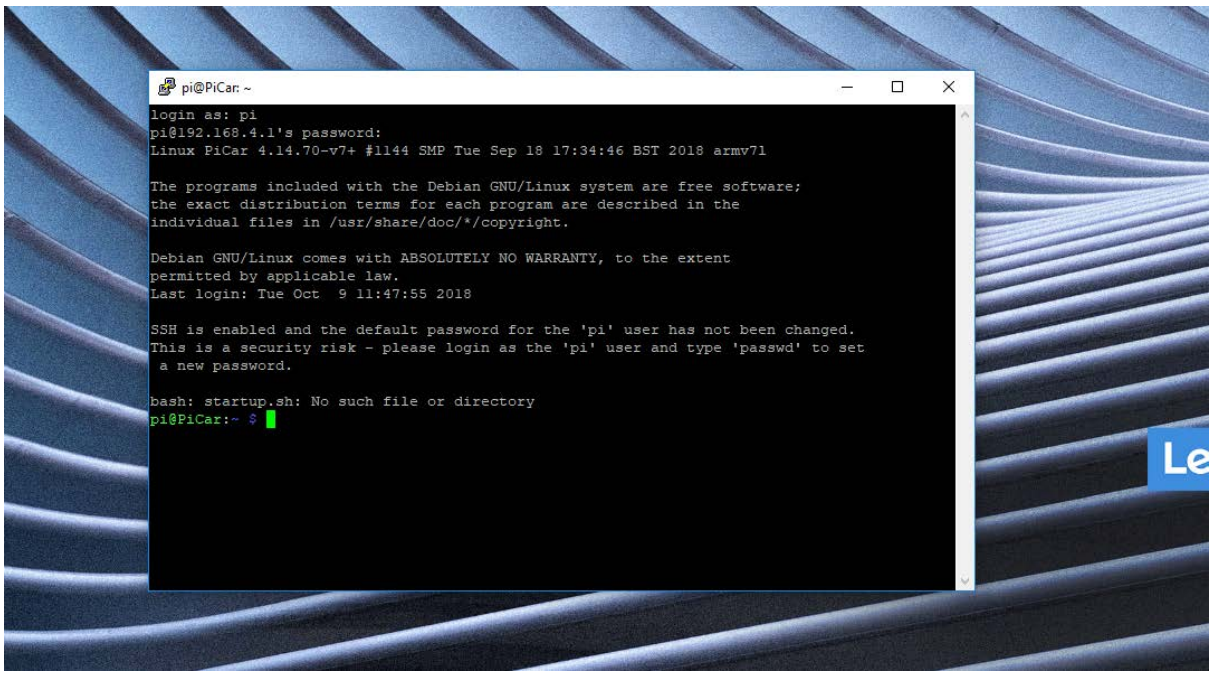
On your first connection you might get a warning message but you can just click “Yes”.



A black window should open asking you for username and password.

```
login as: pi
pi@192.168.4.1 password: raspberry
```

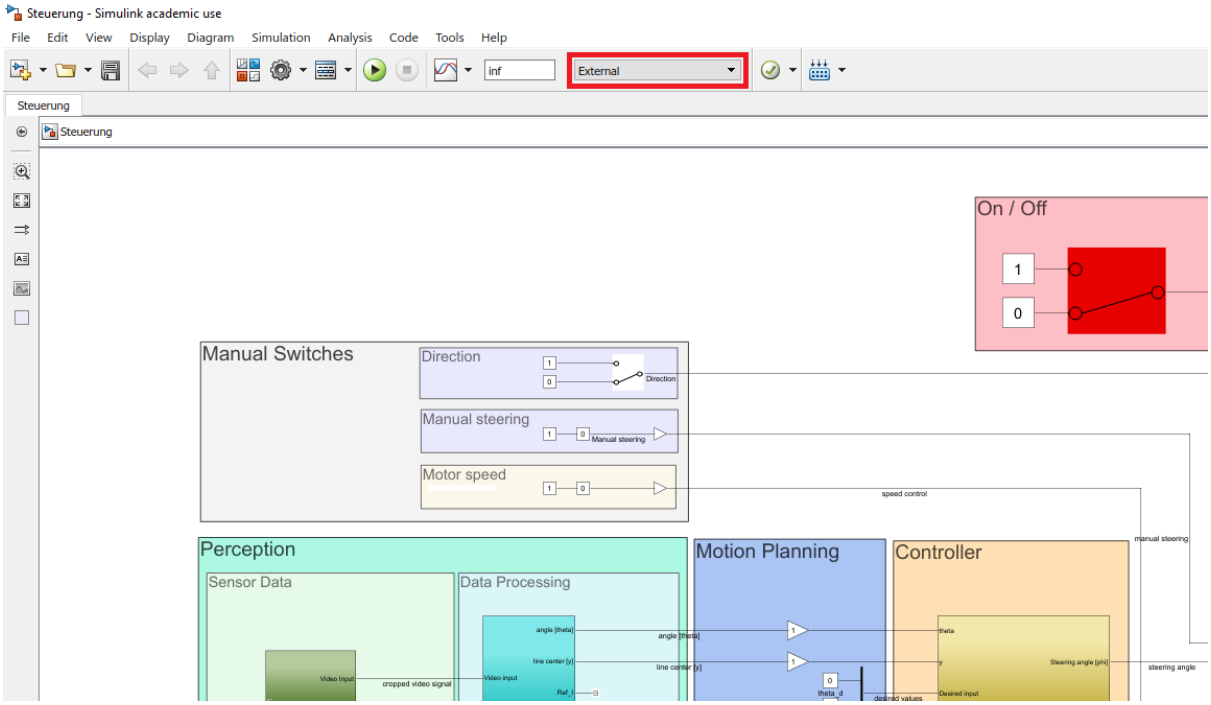
With the given user information you should be able to connect. The window should now look something like this.



You get a command line from which you can interface with the Raspberry Pi directly with Linux shell commands. A few commands you might find useful can be found in the Appendix.

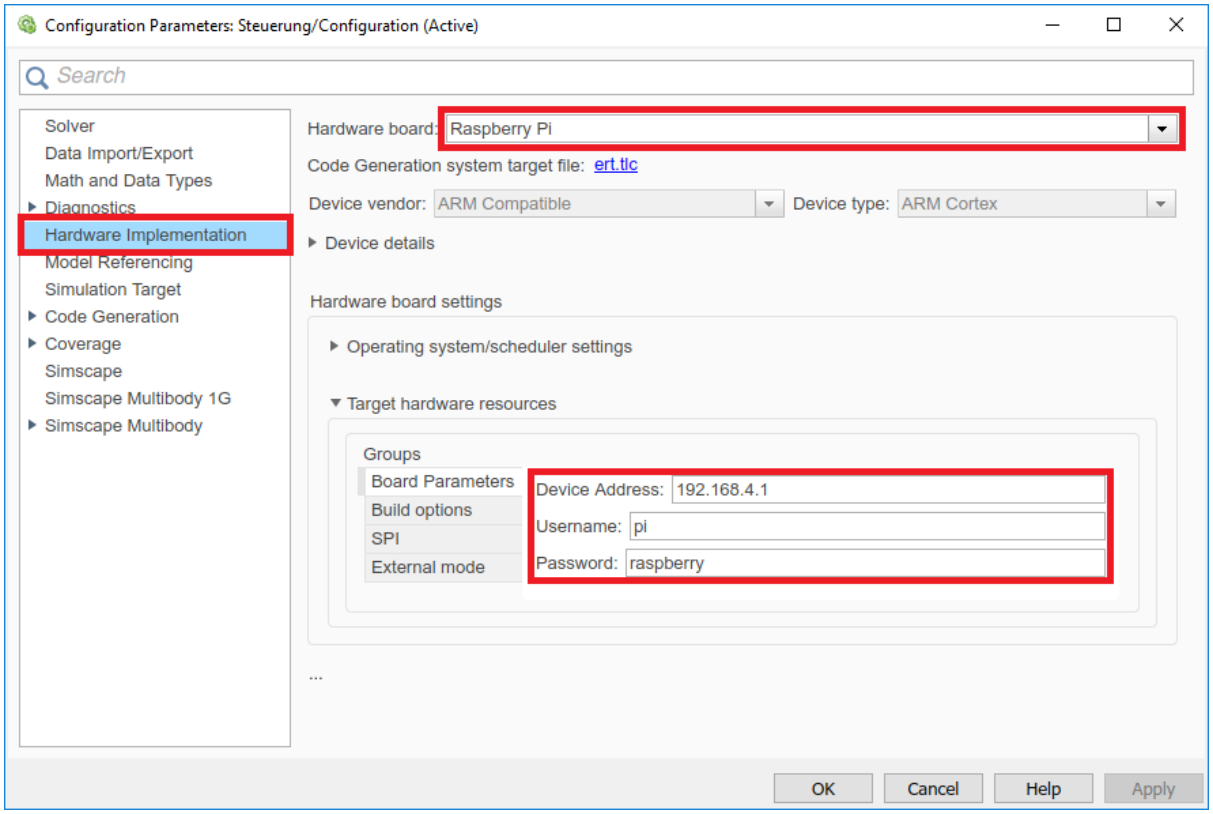
Run A Simple Simulink Model

Before you can try to run the given Simulink example model you have to make sure your Simulink Model Configuration Parameter are set correctly. Open Simulink and make sure it runs in *External Mode.*

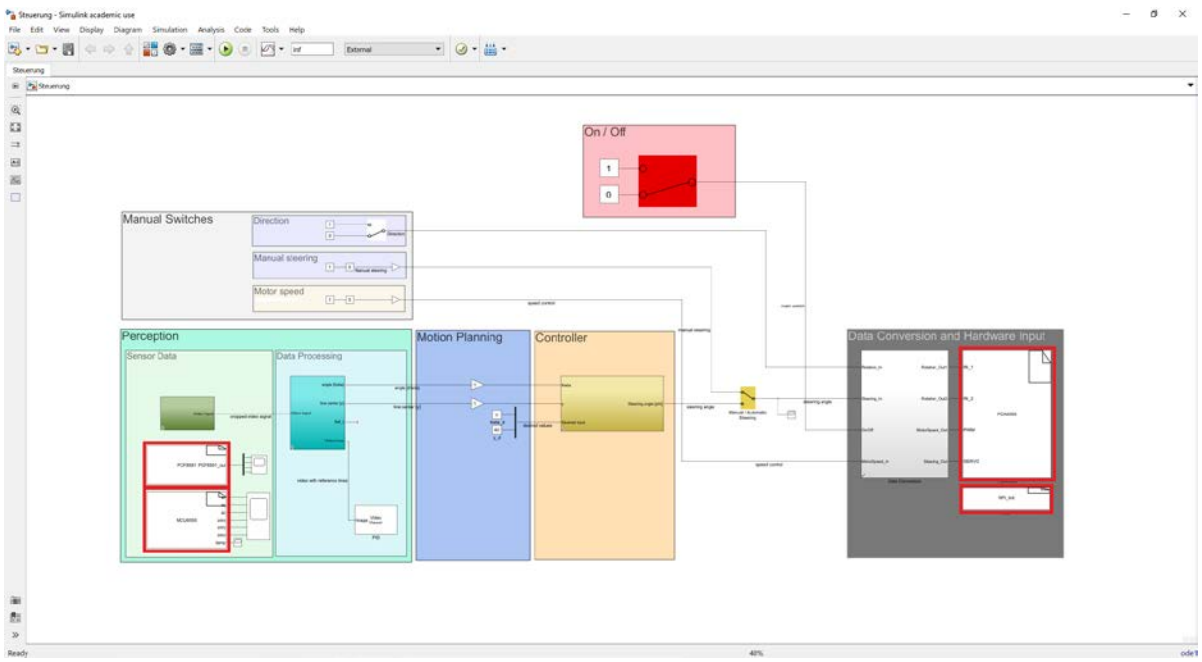


Now open your Model Configuration Parameters  .

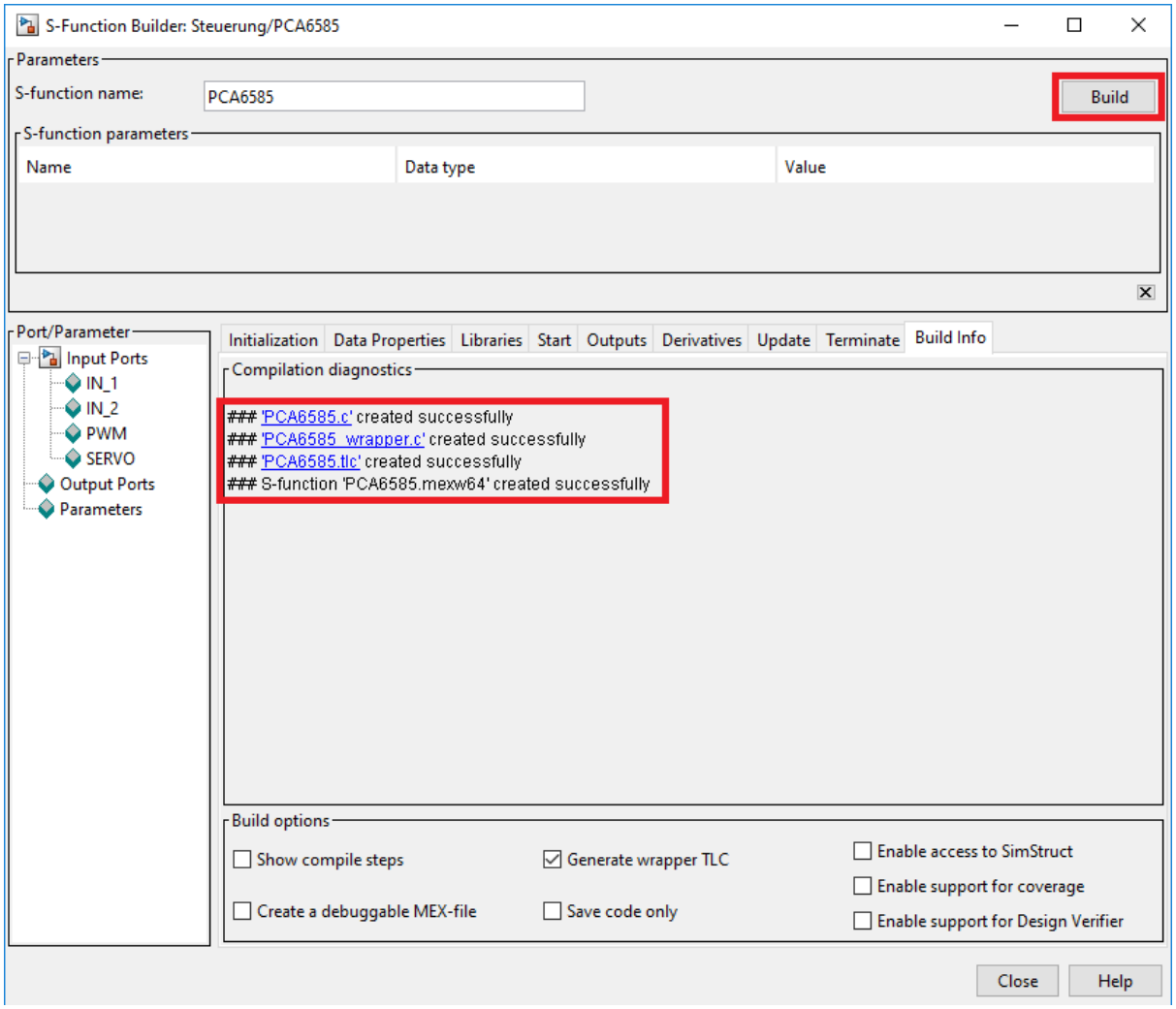
Under the tab *Hardware Implementation* fill in the mask with the given information like this:



The next step is to compile the C functions. This is done by opening an S-function Builder block in the Simulink model and click “Build” in the top corner. Repeat this step for all four blocks.



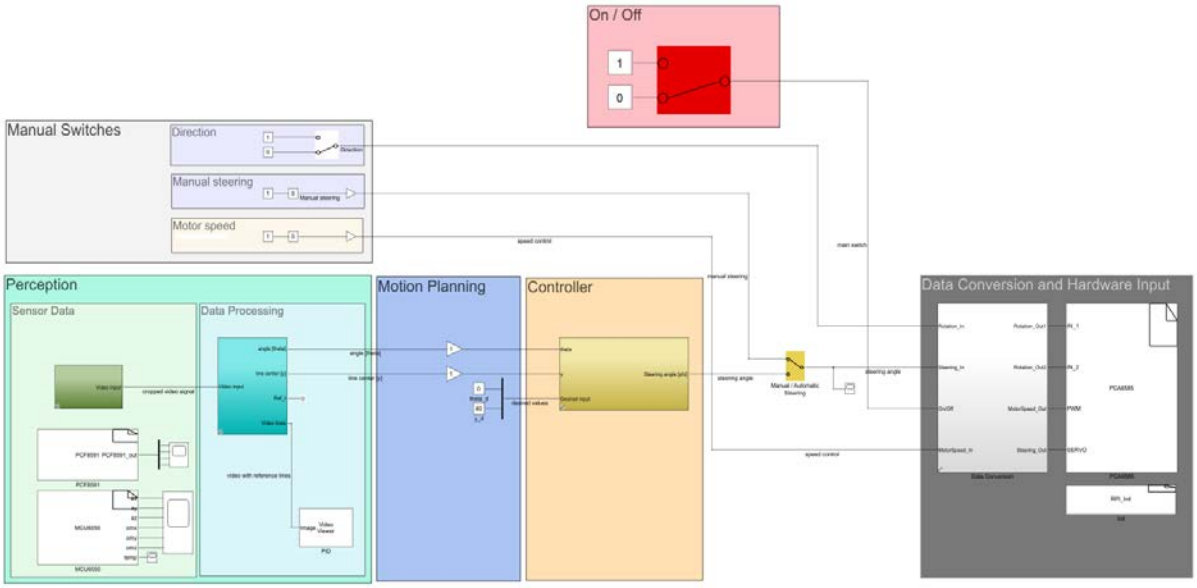
If all builds are finished successfully you can try running the model. This might take a few moments depending on your computer hardware.



A short overview over the components and the functionality is given in the next chapter.

Simulink Example

This is a simple example for a controller design. With this controller, the car is able to follow a high contrast line at relatively low speeds.



The model is separated into blocks that all do different task. Starting from the top is the red *On / Off* switch. For the motor to turn the switch has to be set to 1. It acts as some sort of emergency switch in case something goes wrong it can be used quickly to prevent damage to the PiCar.

The *Manual Switches* are used to control the car manually. Once the model is running you can change the driving direction, the steering angle and the speed.

The block *Sensor Data* contains all Simulink blocks taking inputs from the sensors. The green video block takes in input from the Raspberry Pi camera on the camera arm and crops it to reduce the amount of data. The *MCU6050* outputs data from the IMU.

In the *Data Processing* block the driving line is detected by extracting the camera image with simple image recognition algorithms.

Motion planning contains your desired input or your desired path to your *Controller* which in this case is just a simple P-/PD-Controller that will countersteer, if the car swerves of the driving line.

In the *Data Conversion and Hardware Input* block the calculated steering angle and driving speed are sent to the Hardware. Changes to this block should only be taken with great care or the car might not work at all.

Appendix

Linux Shell Commands

These commands can be used in PuTTY to gain information about the Raspberry Pi in your PiCar. You can just type them in your SSH command line. Some of them might be useful for your development process.

`ls`	show content of current directory
`cd`	change current directory
`cd ~`	change back to home directory
`cd ..`	move up by one in the folder hierarchy
`rm file`	remove “file”
`rm -r directory`	remove “directory” with all its contents
`sudo reboot`	reboot
`ifconfig`	show all active network adapters
`top`	show table of processes

Especially `top` might be useful to get an idea of the computational effort your model takes to run on the Raspberry Pi.

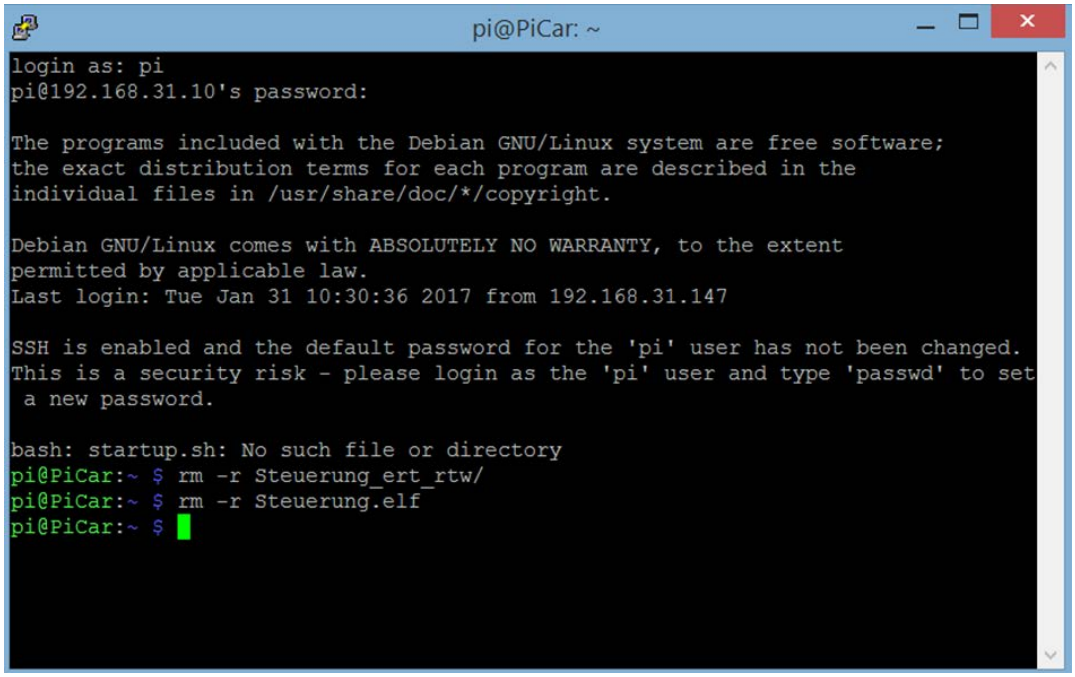
FAQ

Common Errors

For any unknown error message your first attempt on troubleshooting should be to delete files on the Raspberry Pi, which are related to the Simulink model that you are trying to run. The files should have the filename extension:

Steuerung.elt, Steuerung.log and the directory *Steuerung_ert_rtw*.

Delete these files with PuTTY. Afterwards, try to run the model again.


pi@PiCar: ~
login as: pi
pi@192.168.31.10's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 31 10:30:36 2017 from 192.168.31.147
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
bash: startup.sh: No such file or directory
pi@PiCar:~ $ rm -r Steuerung_ert_rtw/
pi@PiCar:~ $ rm -r Steuerung.elf
pi@PiCar:~ $