Autonomous Driving Car Application Note

How to build

About this document

Scope and purpose

This Application Note is intendend to enable users to build an autonomous driving car combined with the Infineon XMC1100 Boot Kit and the DC Motor Control Shield with BTN8982TA.

Intended audience

This document is intended for anyone who wants to build an autonomous driving car (1:10) based on the donkeycar project.

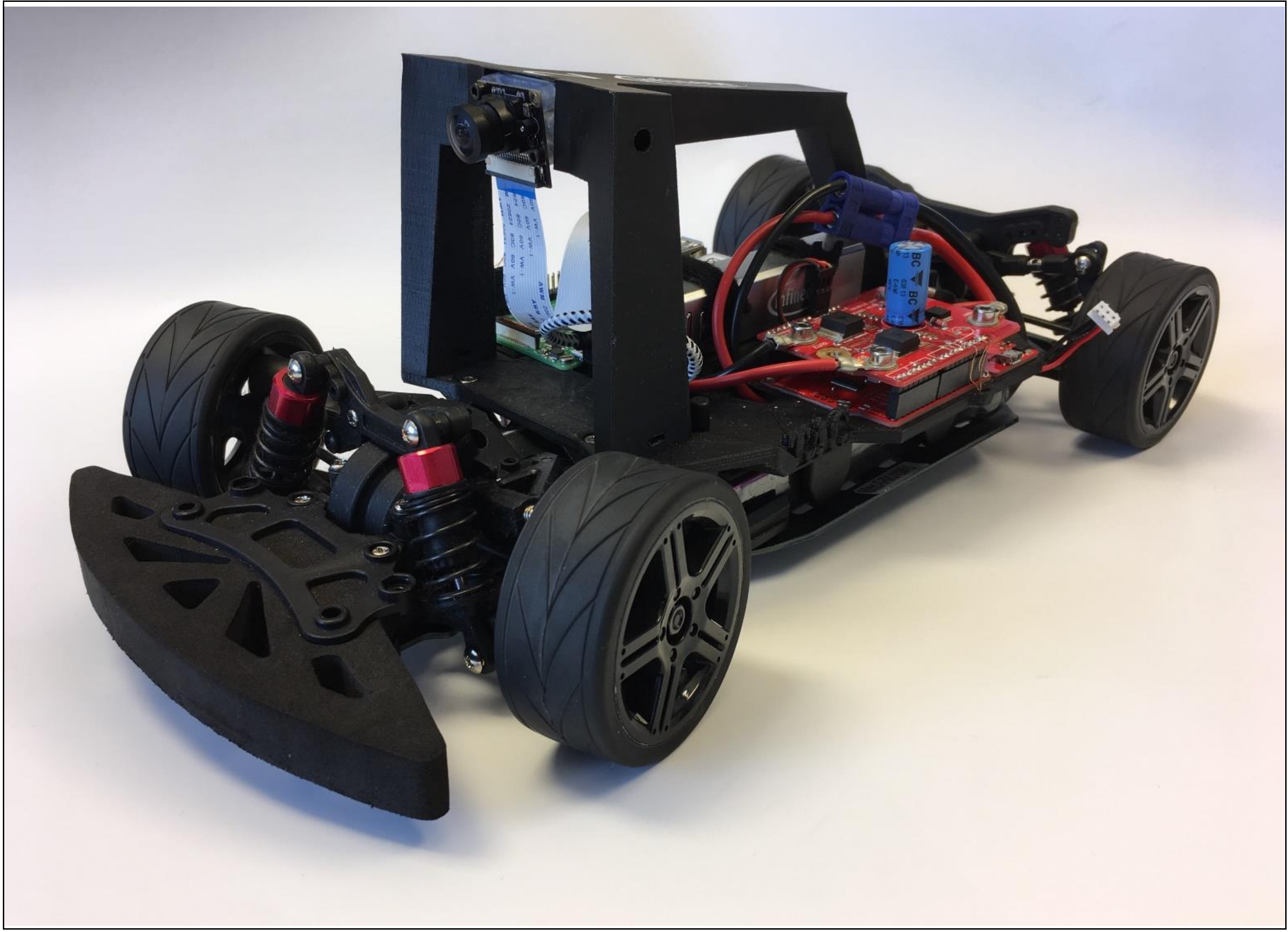


Figure 1 IFX DonkeyCar

Table of contents

1 Baseplate

The basis of the electronic unit of the car is the baseplate. This part is the interface between the electronic components and the cars chassis.

1.1 I2C bus connection

Start with the three wires of the I2C bus. These connect the Raspberry Pi with the XMC1100 Boot Kit.
Thread the cales through the cable inlet on the bottom side of the baseplate. You can leave them open for now on both ends. Later you can solder them to the boards or add connectors to them.

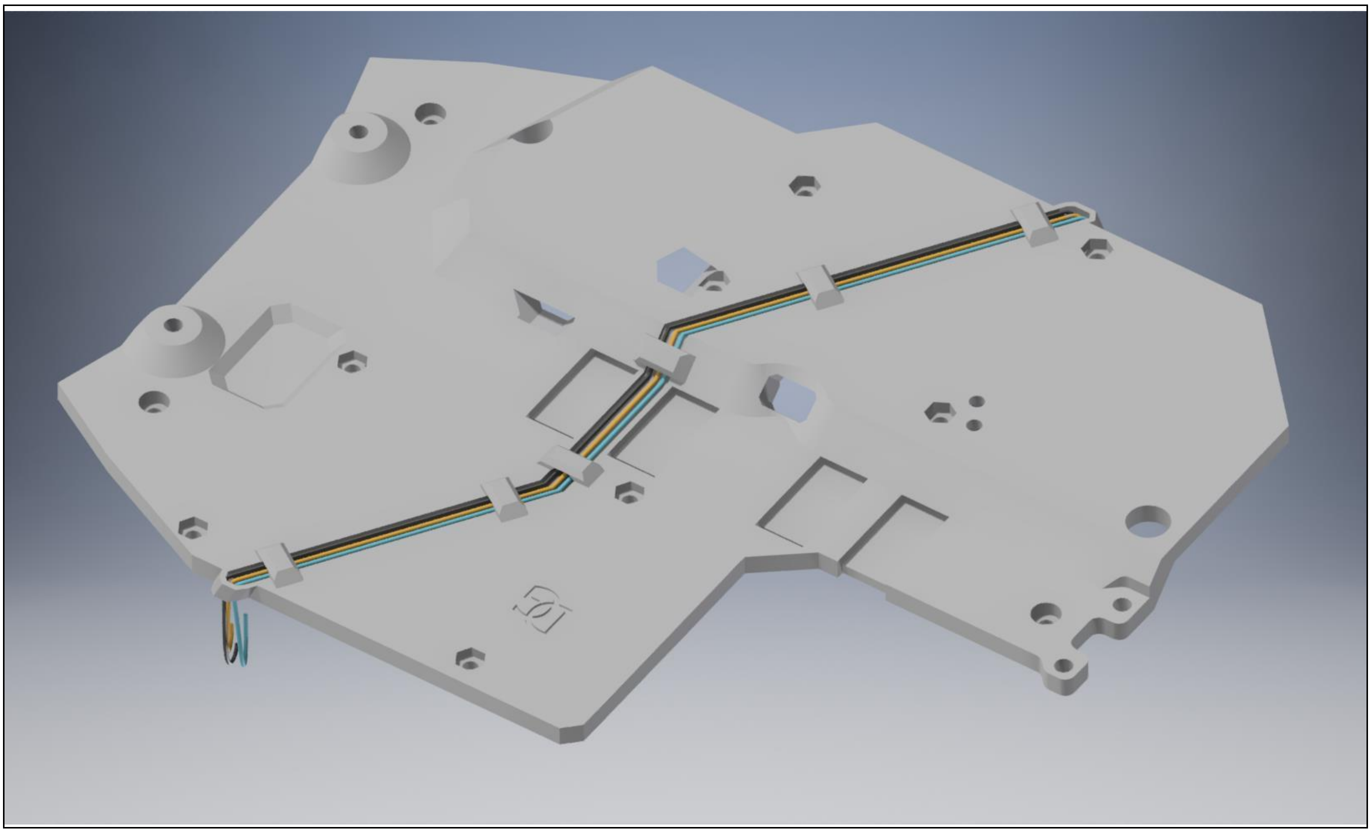


Figure 2 I2C cables on baseplate

1.2 Power supply

The XMC microcontroller and the motorshield are powered by a 2S lipo battery which is located underneath the XMC Boot Kit. The power can be disconnected by deplugging a connector.

The voltage is distributed by two M2,5x10 screws from the bottom of the baseplate via the XMC PCB to the motorshield.

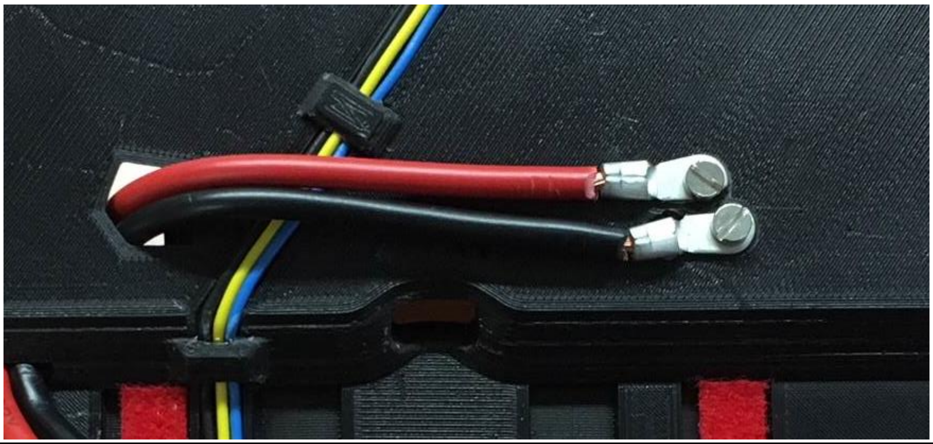


Figure 3 Power supply bottom side of the baseplate

Start with screwing the cables from the bottom of the baseplate. Glue them as well to protect them from getting loose due to the movement of the cable.
After this you can screw the XMC1100 Boot Kit on top of the baseplate. The screws will stand out the power supply connectors of the Boot Kit.

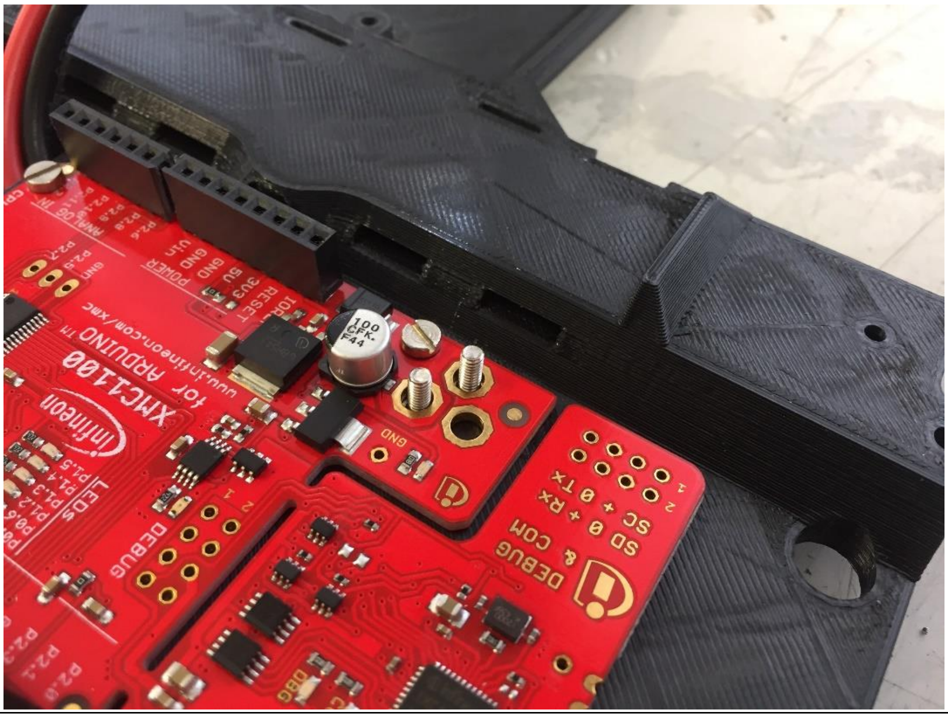


Figure 4 Power supply bottom side of the baseplate

Now you can screw the next cables onto the PCB which connects the XMC1100 Boot Kit power supply with the DC motorshield.

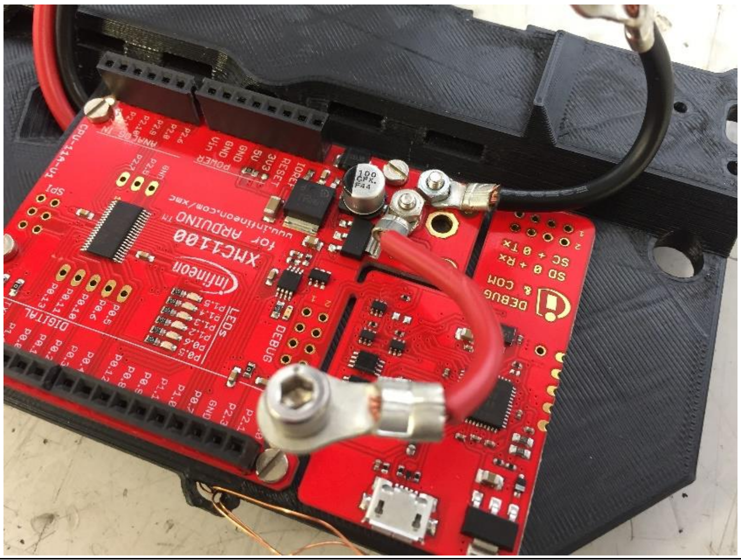


Figure 5 Power supply bottom side of the baseplate

Please make shure that there is an air gap between the two nuts. You can also save this connection with some glue.

1.3 DC Motor Control Shield with BTN8982TA

Before you can stack the motorshield on top of the controller you have to do a small modification to it. You need to add a pin header to one side of the shield which will work as a servo connector later.
Use the GND and 5V output of the XMC Boot Kit as well as Pin 8 as signal output.
Replace the existing pins on GND and +5V with long ones so you get a connection from the Boot Kit to the top of the motor shield. Pin 8 has to be connected to the third pin. This one must be disconnected to the 3V output of the Boot Kit. You can add pins for the I2C connection on the shield in the same way.

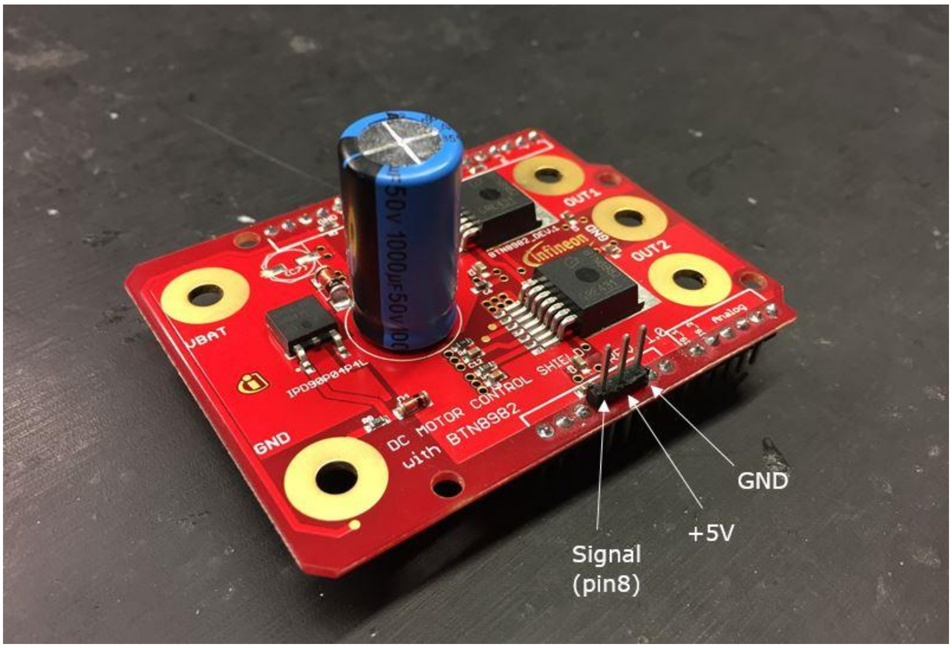


Figure 6 Servo connector on DC Motor Control Shield

Now you can stack the motor shield onto the controllerboard and connect the two motor cables to OUT1 and OUT2 for a bidirectional drive.

1.4 Raspberry Pi

Get the donkeycar image onto your Raspberry, plug in the SD card and screw the Pi onto the baseplate.

To power the Raspberry you will need a 5V power source. Therefore you can use a phone charger battery which will be placed in the middle of the baseplate. To attach the 5V battery to the baseplate you can thread cable ties (we used Velcro ones) or rubber bands through the guidings. Connect the Raspberry to the battery later with a small micro usb cable.

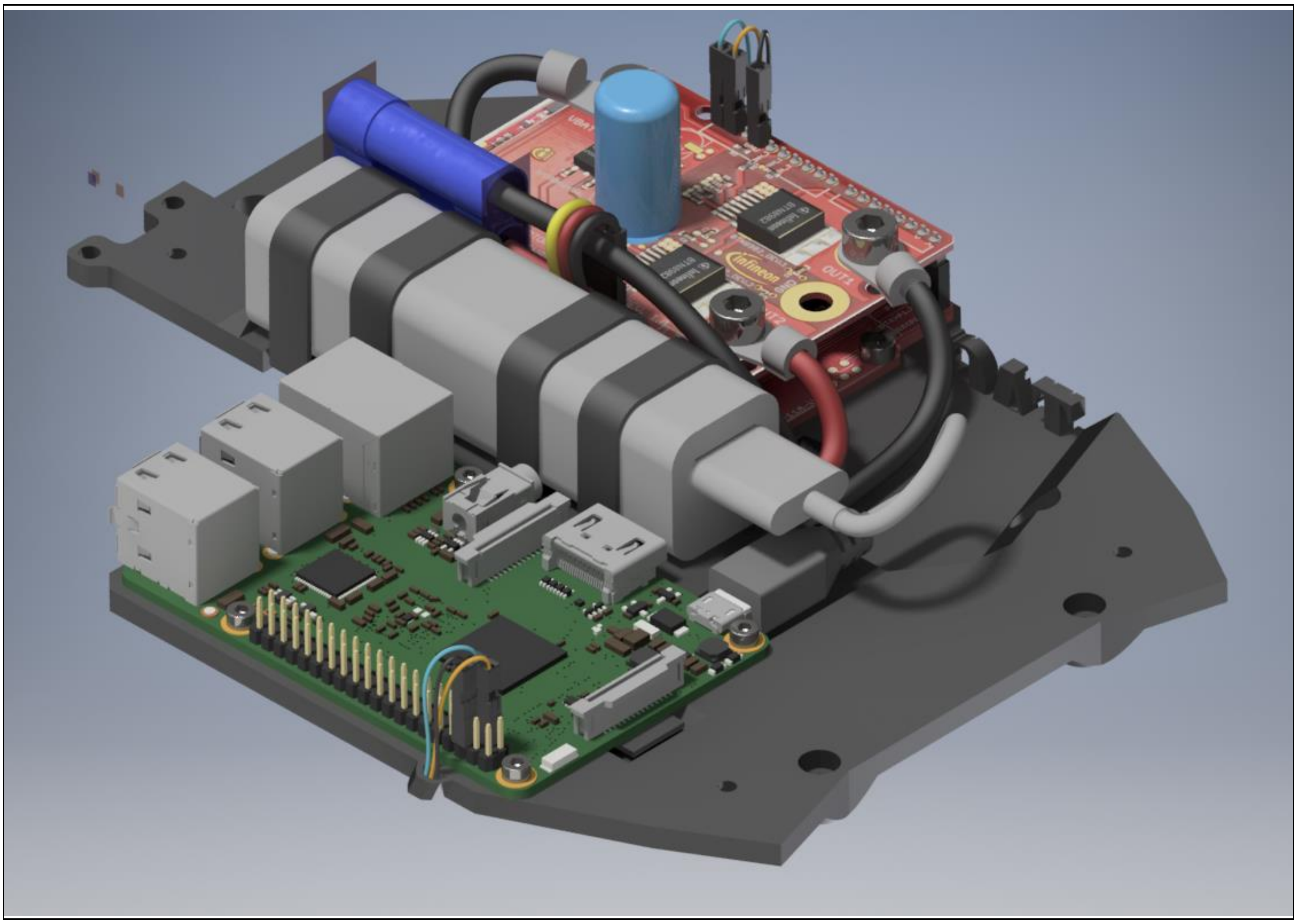


Figure 7 Electronics assembled on baseplate

2 Camera mounting

The camera mounting is a SLA printed part that enables a tiltatable mounting of the camera.

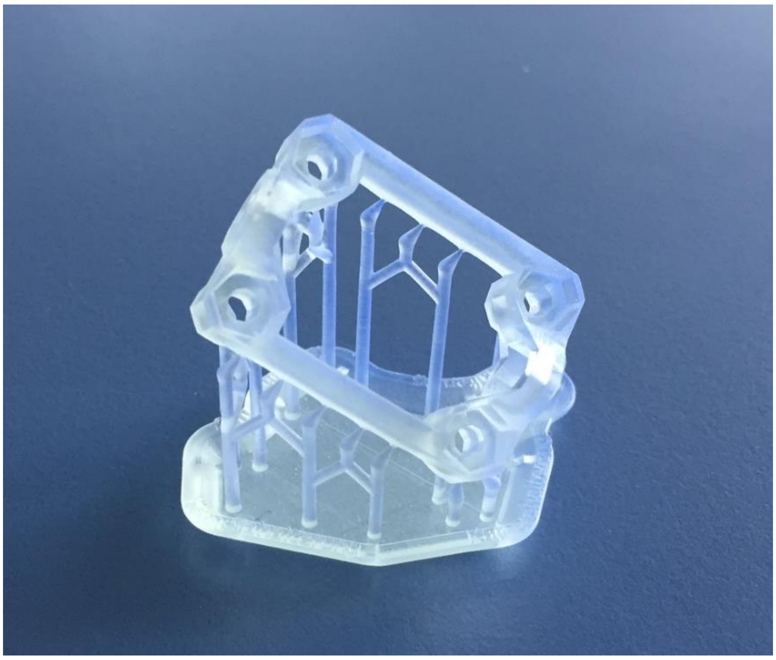


Figure 8 SLA printed camera mounting

It offers several inlays for M2,5 nuts (for the roll cage mount) and M2 nuts (for the camera mount) to fix the camera to the car.

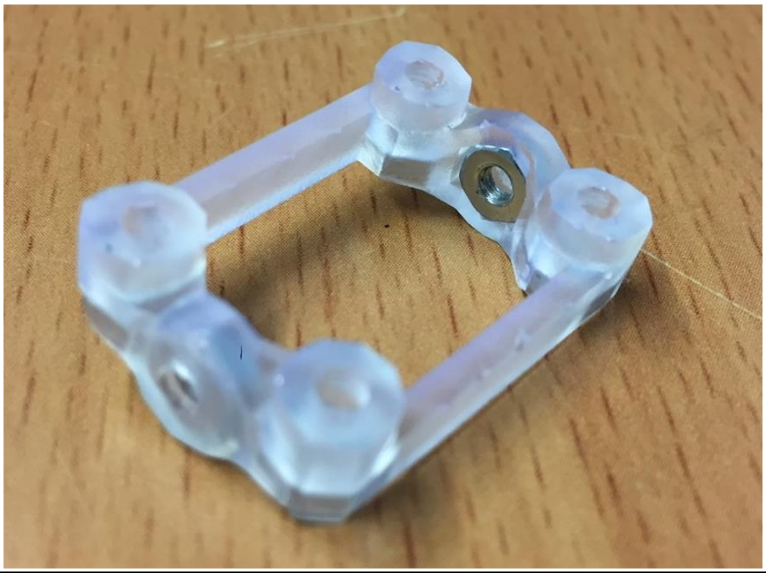


Figure 9 Camera mounting assembled with nuts

Screw the camera onto the mounting and proceed with the roll cage.

3 Roll cage

The roll cage is there to grab the car later on. Therefore it is connected via 3 screws to the baseplate.

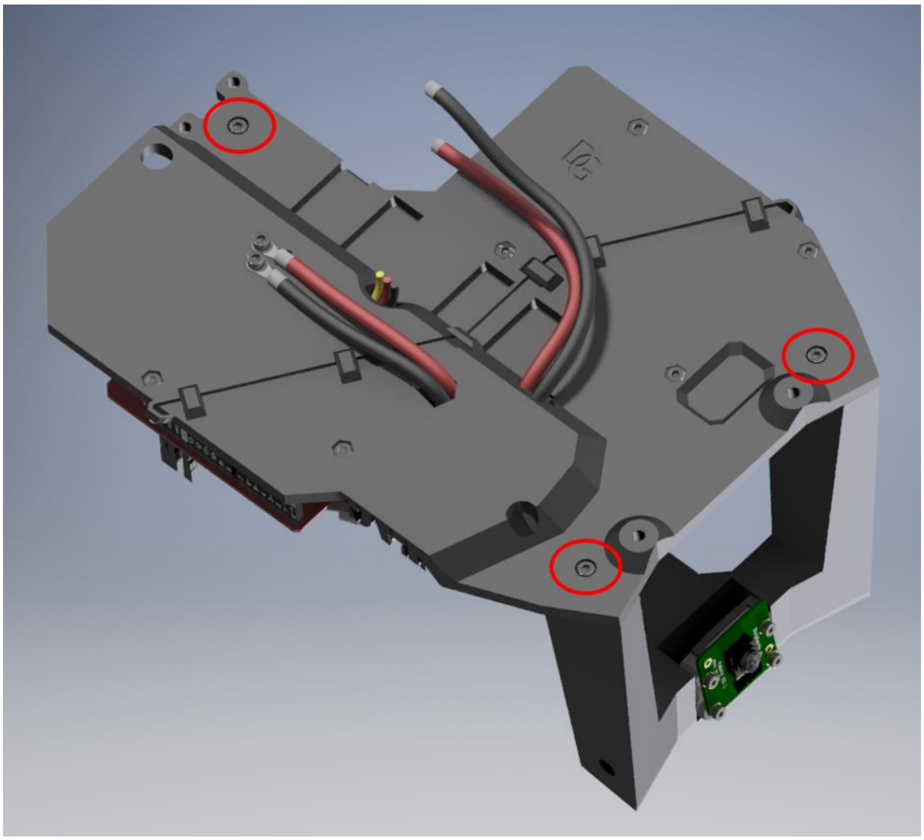


Figure 10 Anchoring points roll cage and baseplate

The second function of the roll cage is to mount the camera onto the car.
The tiltable camera mounting can be screwed onto the cage with two screws. Loose the screws to adjust the camera angle.

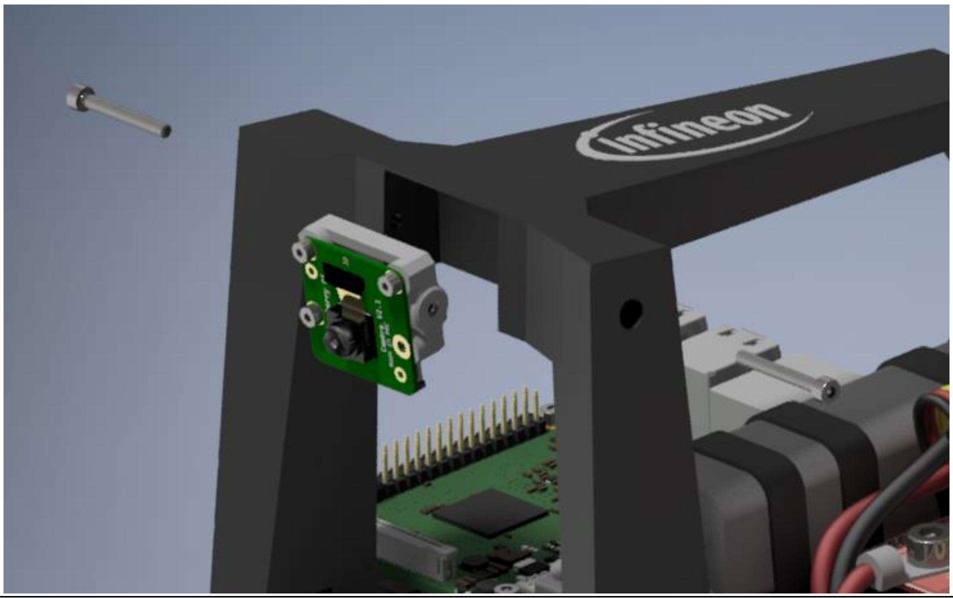


Figure 11 Anchoring points camera mounting and roll cage

4 Chassis

The baseplate can now be screwed onto the Reely TC-04 Chassis with four screws.

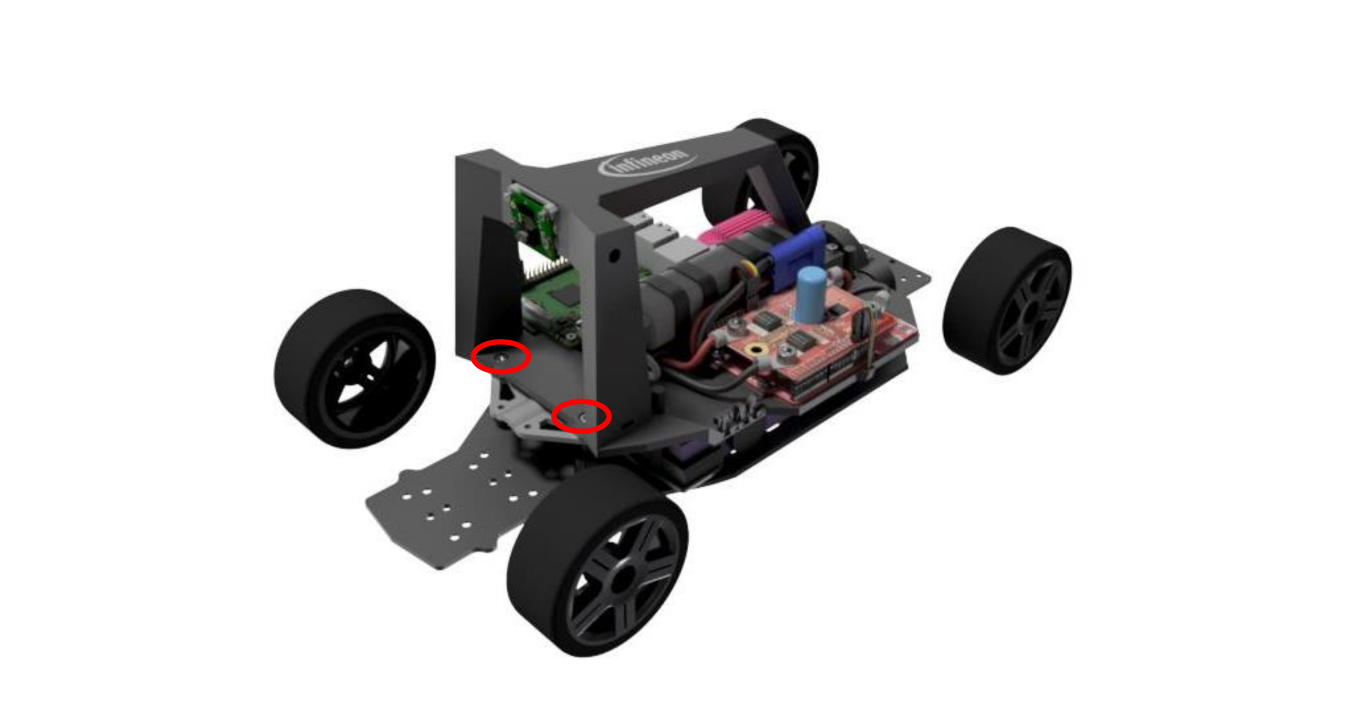


Figure 12 Baseplate mounted on chassis

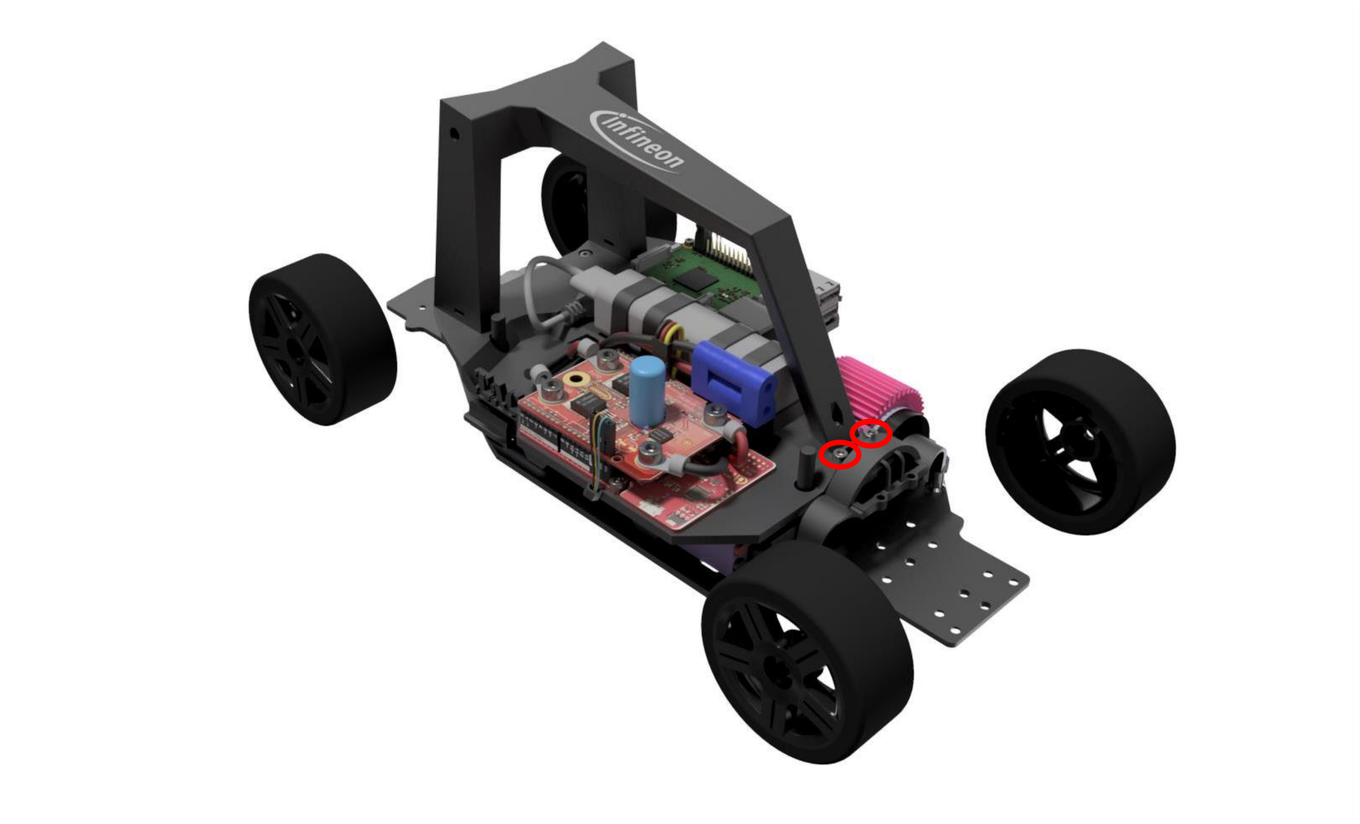


Figure 13 Baseplate mounted on chassis backside

5 Software architecture

The software of the autonomous driving car can be separated in three parts.
Two software parts run on the microcontrollers build into the car. These communicate via I2C Bus.

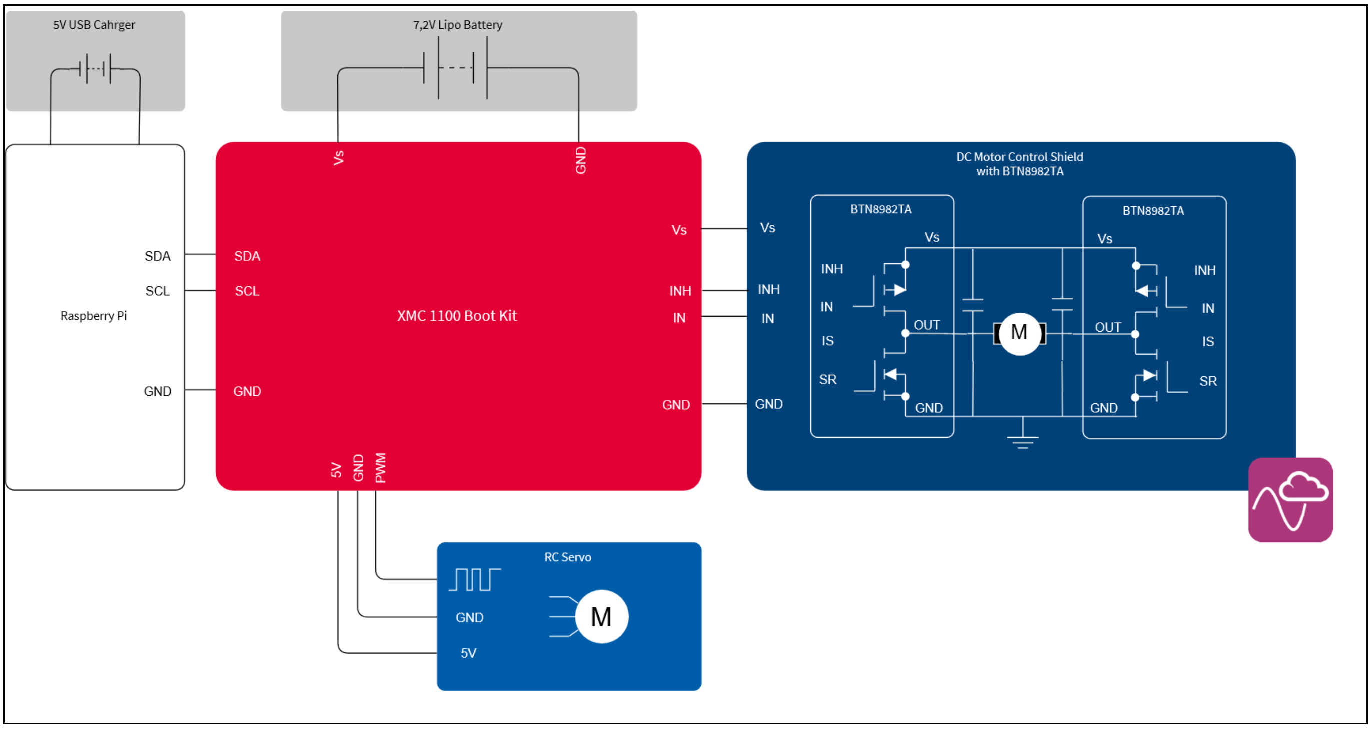


Figure 14 Microcontroller architecture of the autonomous driving car

The third software runs on a PC or server.

5.1 Software to connect the car to your control device and collect data

This software runs on the Raspberry Pi. You can get it on the donkeycars webpage.

It is supposed to

- open a control interface, enabling you to control the car
- collects camera, steering and throttle data (RC controlled mode)
- collects camera data and set steering and throttle accordingly (autonomous mode)
- send steering and throttle signals

5.2 Software to control the motor and steering servo

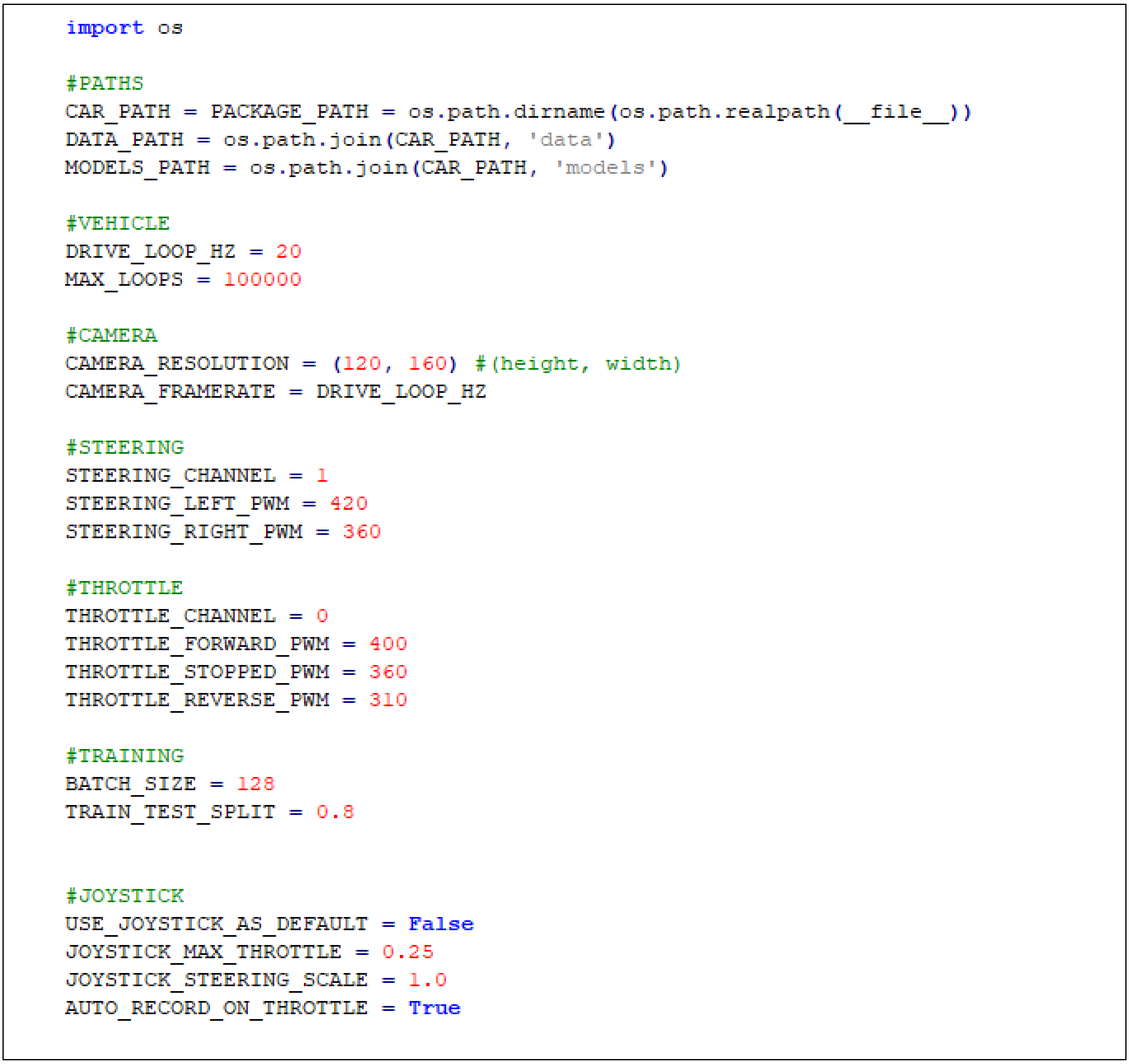
This software runs on the XMC1100. It is programmed to control the steering servo and the DC Motor according to the signals sent by the Raspberry.

5.3 Software to train the neuronal network

This software run on a PC or a server. It uses the collected data to train a neuronal network that will later on control the car while the autonomous drive. This software part is also included in the donkeycar project.

6 Software configuration

Please use the documentation provided by the donkycar website to get started.
You will get to a chapter called “Calibrate your Car”. You can use the following setup of the config.py file as starting point. Combined with the given code for the XMC100 this will set the maximum motor output to 40% of the maximum possible output power. We saw this as a speed that is good to handle for beginners.
According to your motor and servo you might have to to change the values a bit.



```
import os

#PATHS
CAR_PATH = PACKAGE_PATH = os.path.dirname(os.path.realpath(__file__))
DATA_PATH = os.path.join(CAR_PATH, 'data')
MODELS_PATH = os.path.join(CAR_PATH, 'models')

#VEHICLE
DRIVE_LOOP_HZ = 20
MAX_LOOPS = 100000

#CAMERA
CAMERA_RESOLUTION = (120, 160) #(height, width)
CAMERA_FRAMERATE = DRIVE_LOOP_HZ

#STEERING
STEERING_CHANNEL = 1
STEERING_LEFT_PWM = 420
STEERING_RIGHT_PWM = 360

#THROTTLE
THROTTLE_CHANNEL = 0
THROTTLE_FORWARD_PWM = 400
THROTTLE_STOPPED_PWM = 360
THROTTLE_REVERSE_PWM = 310

#TRAINING
BATCH_SIZE = 128
TRAIN_TEST_SPLIT = 0.8


#JOYSTICK
USE_JOYSTICK_AS_DEFAULT = False
JOYSTICK_MAX_THROTTLE = 0.25
JOYSTICK_STEERING_SCALE = 1.0
AUTO_RECORD_ON_THROTTLE = True
```

Figure 15 config.py

7 How to drive everywhere

You need a local WiFi network to get connected to your car. To collect data and drive the car wherever you want you can use your Smartphone as local hotspot. A simple way to implement this is to change the name and SSID of your mobile hotspot to the same as you use to control the donkeycar in you local WiFi network.

The donkeycar will connect automatically to your phone. Use a terminal app to start the donkeycar and start driving wherever you want.

Revision history

Document version	Date of release	Description of changes
V1.0	20.09.2018	Initial release

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2018-09-20

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AppNote Number

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.