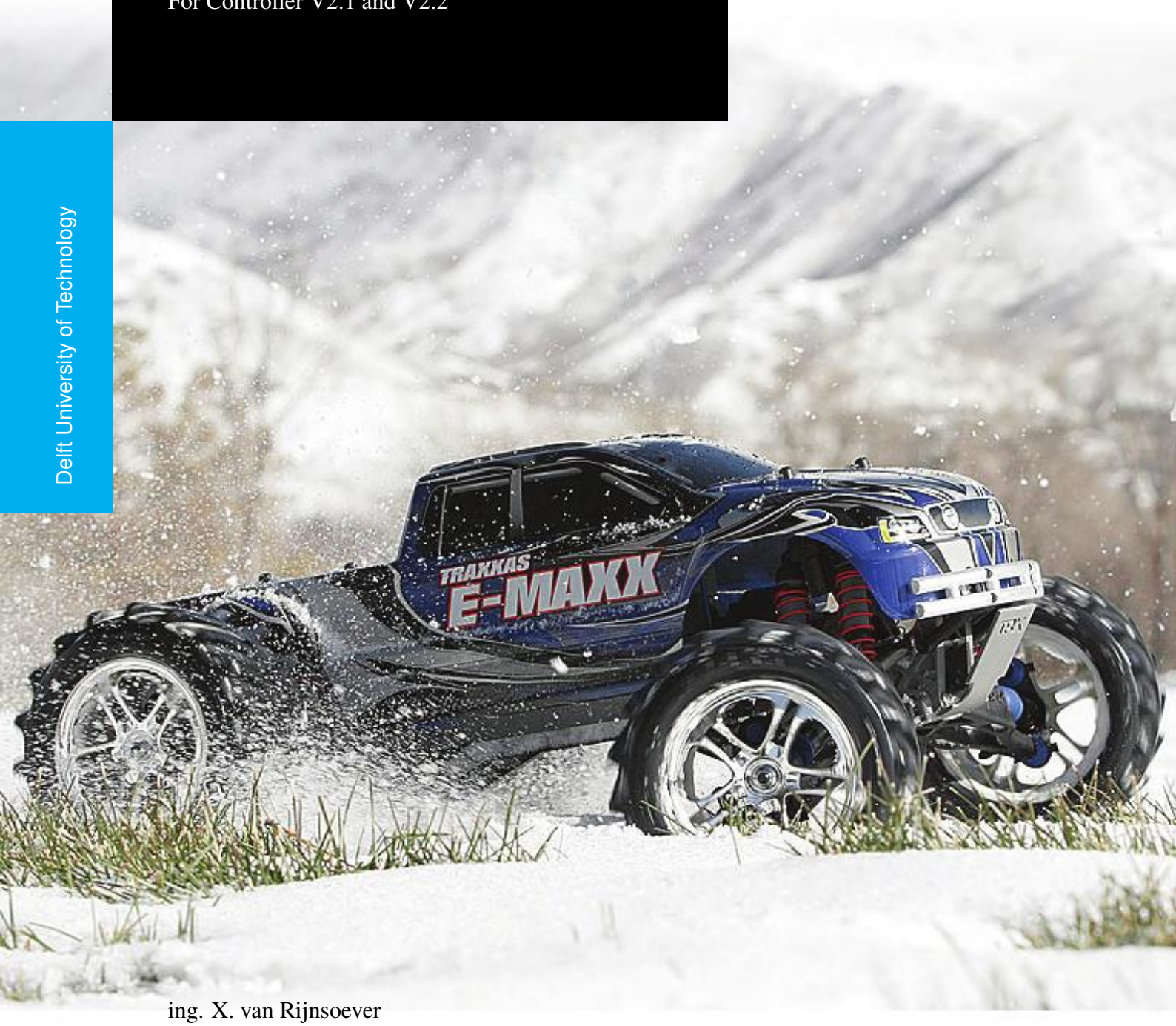
Code EE2L21

EPO4 Car Documentation and Test Manual

For Controller V2.1 and V2.2



ing. X. van Rijnsoever

Delft University of Technology

TU Delft Delft University of Technology

Challenge the future

Contains work by M.E. Weskin M.Sc.

EPO4 Car Documentation and Test Manual For Controller V2.1 and V2.2 V1α (Friday 22nd December, 2017)

This manual is typeset with LaTeX in Nimbus Roman 10 pt. The diagrams are created with Dia, the photos and screenshots are edited with GIMP and ImageMagick.

Contents

Part I

Assembly

Chapter 1

EPO-4 Car Mechanical Modifications

1.1 Introduction

This chapter describes the mechanical modifications applied to the EPO-4 car. The information in this chapter might be inaccurate, since no journal was kept during modification of the original EPO-4 cars. In case a new car is build it might be helpful to compare the car to one of the already modified EPO-4 cars. Please note that some modifications can only be finished properly *after* the wiring of certain parts has been finished and vice versa!

1.2 Suspension

The original suspension of the car is too soft for the lab: when braking hard, the nose of the car tends to dive. This will lead to wrong readouts of the distance sensors.

Front Suspension

The front suspension is stiffened by replacing the default 2 mm spacers with pieces of 16 mm (5/8") PVC pipe of 1.5 cm long. This is denoted in the exploded view in Appendix D.2 (see the Shock Assembly block on the left) as No. 3769.

Rear Suspension

The back suspension is stiffened by replacing the default 2 mm spacers with pieces of 16 mm (5/8") PVC pipe of 2.5 cm long. This is denoted in the exploded view in Appendix D.2 (see the Shock Assembly block on the left) as No. 3769.

1.3 Mounting Components

In order to mount the power board, the controller board, and the motor controller, mounting holes have to be drilled. Figure 1.1 shows the mounting locations of the different components.

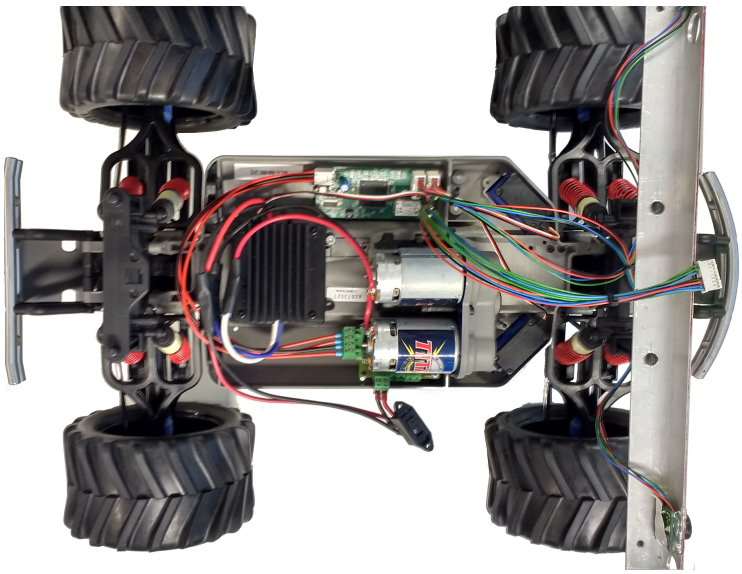


Figure 1.1: Photo of the EPO-4 car showing the mounting locations of the boards

1.3.1 Removing Obsolete Parts

Before the power board, the controller, and the motor controller can be mounted, some parts of the original car have to be removed. See Appendix D.1:

- On both sides, remove the components that make up the battery compartiment:
  - Remove brackets and foam blocks 3927;
  - Remove bracket holders 3928 (front and back);
- Remove the original motor controller 3019R;
- Remove the original controller board assembly, everything inside and including 3924;

1.3.2 Power Board

The power board is mounted in the right battery compartiment (as seen from the back of the car), see also Figure 1.1. The power board has four mounting holes, however, the upper right mounting is very difficult to reach, since it is blocked by one of the motors. The power board can firmly be fixed using the remaining three mounting holes. Mount the board using 10 mm plastic standoffs and 20 mm M3 machine screws with a lock washer and a nut.

1.3.3 Controller Board

The controller board is mounted in the left battery compartiment (as seen from the back of the car), see also Figure 1.1. Two holes have to be drilled, the upper right mounting hole in the controller board is not used. The board is mounted using 10 mm plastic standoffs and 20 mm M3 machine screws with a lock washer and a nut.

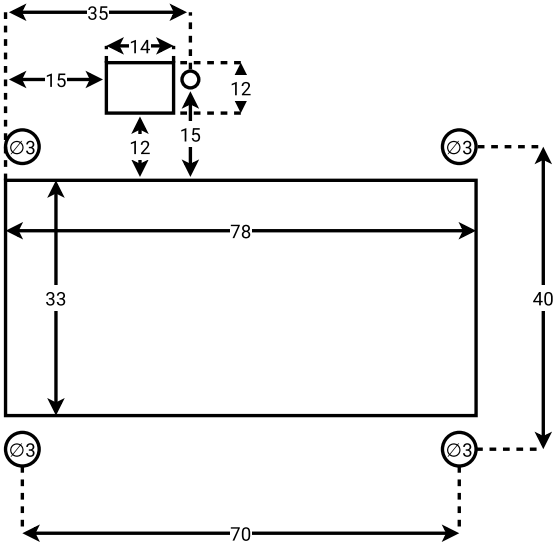


Figure 1.2: Cutouts for display, leds, and reset button. All measurements are in cm

1.3.4 Motor Controller

The original motor controller has to be removed. The more powerful new motor controller is too big to fit in the same space. It is mounted on the same spot, however, new mouting holes have to be drilled. The motor controller is mounted with the wires coming out of the sides: the control wire has to point left, when seen from the back of the car. This way the control wire can be connected to the controller board. Mount the controller with 2×15 mm wood screws.

1.4 Top Cover

The top cover will hold the display, the speaker, some leds, and the reset button. Furthermore, some recesses have to be made in order to make it easier to unplug the power connector.

1.4.1 Display Module

The display module contains the display, four indicator leds, and the reset button. The module is placed in the middle of the "window" of the car, such that the cutout for the leds is just above the "window". The cutouts and holes to drill are shown in Figure 1.2.

1.4.2 Speaker

The speaker is mounted on top of the cover. The speaker requires a hole with a diameter of 5 cm and three holes of 3 mm diameter located at the nodes of a equilateral triangle with edges of 5 cm.

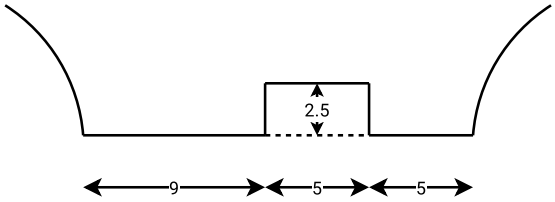


Figure 1.3: Cutout in the side of the top cover (measurements in cm)

1.4.3 Power Supply Recesses

Some recesses have to be cut in the side of the cover to enable easy removal of the power connectors. The cutouts have to be made in the right side of the car. Figure 1.3 shows the exact location of the cutout.

1.5 Sensor Bar

Wire sensors first!
Wire the sensors *before* fixing them into place on the sensor bar! Otherwise it may be impossible to solder the wires to the sensors.

The ultrasonic acoustic sensors are mounted onto a piece of aluminium angle bar. See Appendix E for dimensions. Mount the sensors as follows:

- Tape off the bottom part of the bar underneath the spot where the sensors will be mounted;
- Place the sensor transceiver into the hole in the bar;
- Fix the sensor in place by applying hot glue onto both the top of the transceiver and the bar and onto the bottom of the PCB.

Figure 1.4 shows the final result. The sensor bar can now be mounted using the existing mouning points on the hood of the car.

1.6 Battery Holder

The battery holder is mounted on the back of the car using modified metal mounting hardware. The battery holder is designed to hold two stacked 8.4 V NiMh battery packs.

1.6.1 Individual Battery Holder

The dimensions of a single battery holder can be found in Appendix F. The battery holder is placed onto the car using two L-brackets. A photo of a battery holder is shown in Figure 1.5.

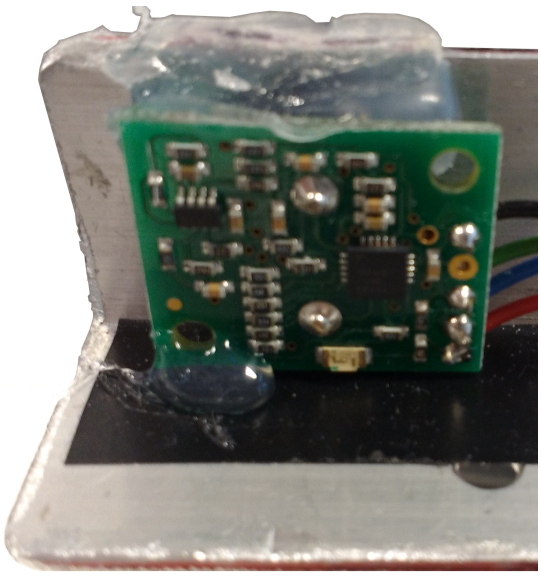


Figure 1.4: Photo of a mounted ultrasonic acoustic sensor

Figure 1.5: Photo of the battery holder

1.6.2 Constructing the Battery Holders

The battery holders are made by building a box and sawing that in quarters. From a single 1/4 sheet of 4 mm MDF it is possible to construct five boxes and thus 20 battery holders.

The following parts are required to build 20 battery holders:

#	Size [mm]	Description
10	332×104	Top and bottom
10	324×50	Long sides
10	104×50	Short sides
40	50×20	Front sides battery holders

Appendix F contains a layout of these parts on a 1/4 sheet of MDF. Use the following cut plan:

- Cut the long 50 mm strip of the long side;
- Cut a 324 mm wide piece of the short side of the remainder;
- Cut the remaining part in strips of 104 mm each;
- Cut two pieces of 50 mm out of each of the 104 mm strips;
- Cut ten pieces of 50 mm out of the 324 mm wide piece;
- Now cut two pieces of 332 mm length from the remainder of the 104 mm pieces;
- Finally, cut up the long 50 mm strip in 20 mm wide pieces.

With all materials cut, construct the boxes:

- Glue the 104×50 mm and 324×50 mm pieces onto the 332×104 mm pieces;
- Glue the top pieces on top and let the glue set for at least an hour.
- When the glue is dry, sand the boxes;

Cut the finished boxes over the X and Y axes in order to create the battery holder. Glue two of the small 50×20 mm pieces on the front of the battery holder. Use black spray paint as finish.

1.6.3 Mounting the Battery Holder

The battery holder is mounted using two 40 mm $\times$ 40 mm L-brackets (stoelhoek) of which one of the innermost holes is enlarged from 5 mm to 8 mm. The brackets are mounted on the battery holder with hot glue. Use the following steps to mount the battery holder on the car:

- Position the battery holder on the back of the car;
- Position the enlarged holes of the L-brackets over the mounting pins on the back of the car;
- Use a hot glue gun to glue the L-brackets to the battery holder.

1.6.4 Problems and Proposed Solution

The batteries have a little slack in the battery holder. When the cars accellerate at full speed, the batteries can move and knock out the small 50×20 mm pieces of the battery holder.

A possible solution is shown in Appendix F.4. The two small pieces are now replaced with an 15 mm piece of MDF, 44×50 mm. This piece is inset into the battery holder and glued with wood glue. After the glue has dried 8 mm dowels can be inserted by drilling holes through the bottom and the sides of the battery holder. The combination of the increased contact area for the glue and the addition of the through-hole dowels is expected to be able to withstand the impact of the moving batteries.

Note that increasing the thickness of the piece and insetting the piece will require the length of the battery holder to increase to 165 mm.

Use the following steps to construct and the new end-piece of the battery holder:

- Cut pieces of 44×50 mm out of a sheet of 15 mm MDF;
- Drill an 18 mm hole 25 mm from the side, 18 mm from the bottom;
- Cut away the material from the top to the hole, 16 mm from each side;
- Glue in the piece into the battery holder;
- After the glue has set, drill two 8 mm holes, 15 mm deep on both sides, 20 mm from the top, halfway in the material of the end-piece;
- Drill two 8 mm holes, 20 mm deep in the bottom, 12 mm from the sides, halfway in the material of the end-piece;
- Glue 8 mm dowels into the holes;
- After the glue has set, use a flush trim saw to trim the dowels;
- Sand as required;
- Use black spray paint as finish.

Chapter 2

EPO-4 Car Electrical Wiring

2.1 Introduction

This chapter describes the electrical wiring of the EPO-4 car.

2.2 Wiring Overview

An overview of the complete wiring of the EPO-4 car is shown in Figure 2.1.

The sections below each describe a part of the wiring. Appendix B describes the wiring of the different connectors on the controller board.

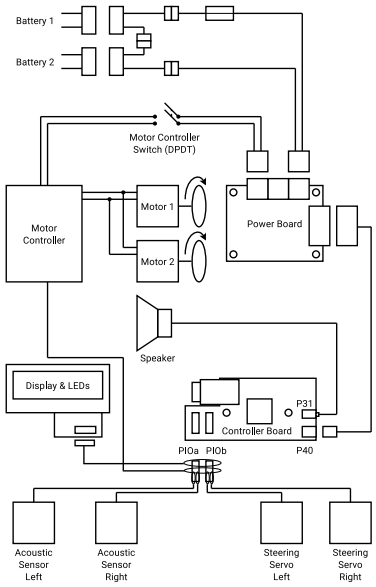


Figure 2.1: Overview of the EPO-4 car wiring

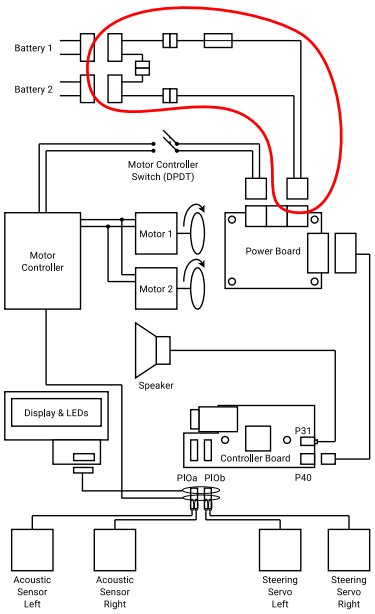


Figure 2.2: Location of the battery cable assembly

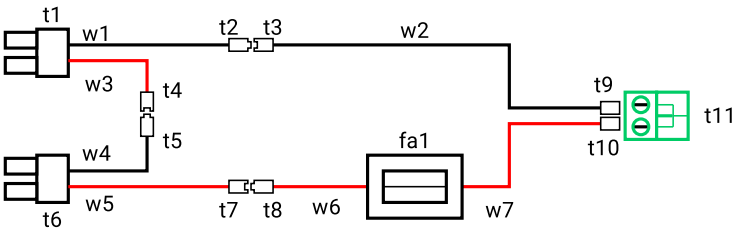


Figure 2.3: Wiring diagram battery cable assembly

2.3 Battery Cable Assembly

This section describes the battery cable assembly that is used to connect two batteries to the EPO-4 car. Figure 2.2 shows the location of the battery cable assembly in the wiring overview.

2.3.1 Components

The battery cable assembly consists of the following components:

Amount	Description	Designators	Origin
2	Traxxas Connector M	t1, t2	Cut from original motor controller
3	Quick Conn. Crimp M	t2, t5, t8	
3	Quick Conn. Crimp F	t3, t5, t7	
2	Ferrules	t9, t10	
1	Fuse Assembly	fa1	Farnell No. 319480
1	Fuse 10 A		Farnell No. 1360862
1	Terminal Block 2-way	t11	Farnell No. 3705353

2.3.2 Wiring

Figure 2.3 shows the diagram of the assembly of the battery cable assembly.

2.3.3 Construction

- Gather all components and cut all cables to length, and strip the cable ends;

- Screw ferrules t9 and t10 into pluggable terminal block t11.

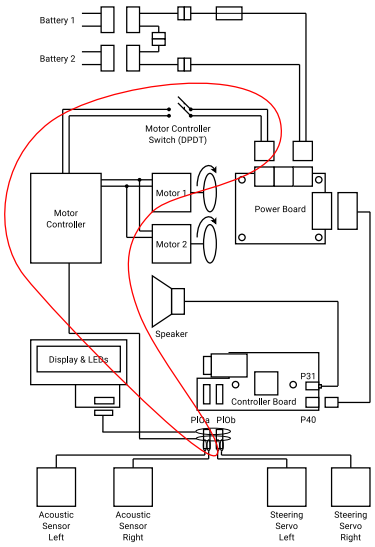


Figure 2.4: Location of the motor controller

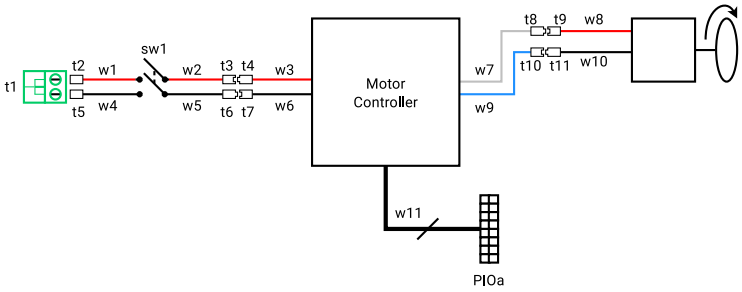


Figure 2.5: Wiring diagram motor controller

2.4 Motor Controller

This section describes the wiring of the motor controller. Figure 2.4 shows the location of the motor controller in the wiring overview.

2.4.1 Components

The motor controller wiring consists of the following components:

Amount	Description	Designators	Origin
1	Motor Controller		Robbe 540 R
1	Terminal Block 2-way	t1	Farnell No. 3705353
2	Ferrules	t2, t3	
1	Slide switch DPDT	sw1	Farnell No. 147911
4	Quick Conn. Crimp M	t4, t6, t8, t11	
4	Quick Conn. Crimp F	t3, t7, t9, t10	
3	JST SPUD-001T-P0.5		RS No. 906-0275
1	JST PUDP-16V-S	PIOa F	RS No. 906-0256

2.4.2 Wiring

Figure 2.5 shows the diagram of the assembly of the motor controller.

2.4.3 Construction

- Gather all components and cut all cables to length and strip the cable ends;
- Cut the connector of the control wire of the motor controller;

- Solder w1, w2, w4, and w5 to switch sw1 and crimp heatshrink over all terminals;
- Screw w1 and w4 into pluggable terminal block t1.

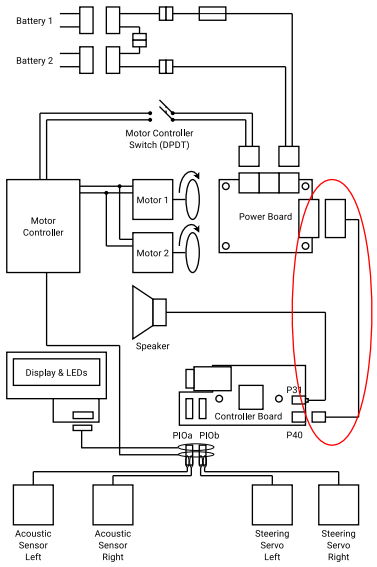


Figure 2.6: Location of the power board to controller cable

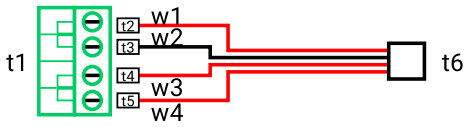


Figure 2.7: Wiring diagram power board to controller cable

2.5 Power Board to Controller Cable

This section describes the wiring of the power board to controller cable. Figure 2.6 shows the location of the cable.

2.5.1 Components

The power board to controller cable consists of the following components:

Amount	Description	Designators	Origin
1	Terminal Block 4-way	t11	Farnell No. 3705353
2	Ferrules	t2, t3, t4, t5	
3	JST SPUD-001T-P0.5		RS No. 906-0275
1	PUDP-08V-S	t6	Farnell No. 183-0796

2.5.2 Wiring

Figure 2.7 shows the diagram of the assembly of the power board to controller cable.

2.5.3 Construction

- Gather all components and cut all cables to length and strip both cable ends;
- Crimp JST SPUD-001T-P0.5 connectors onto one cable end and plug into connector t6;
- Crimp ferrules t2 – t5 onto the other cable ends;
- Screw w1 – w4 into pluggable terminal block t1.

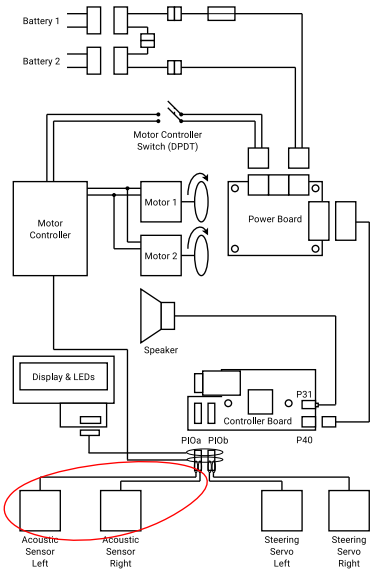


Figure 2.8: Location of the sensors

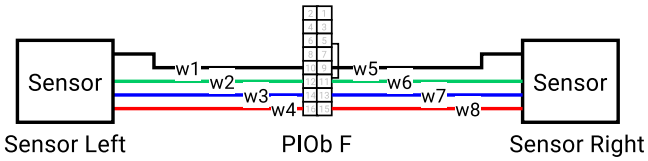


Figure 2.9: Wiring of the acoustic sensors.

2.6 Sensors

This section describes the connection of the two acoustic sensors. Figure 2.8 shows the location of the sensors in the wiring overview.

2.6.1 Components

Amount	Description	Designators	Origin
2	Ultrasonic Ranger SRF02	Sensor Left, Sensor Right	Antratek No. srf02
8	JST SPUD-001T-P0.5		RS No. 906-0275
1	JST JST PUDP-16V-S	PIOb F	RS No. 906-0256

2.6.2 Wiring

Figure 2.9 shows the connection between the controller and the sensors.

2.6.3 Construction

- Gather all components and cut all cables to length;
- Strip both ends on all cables and tin one end;
- Solder the tinned cable ends to the sensor PCB;
- Crimp JST SPUD-001T-P0.5 connectors onto the other cable ends and plug into connector PIOb F;

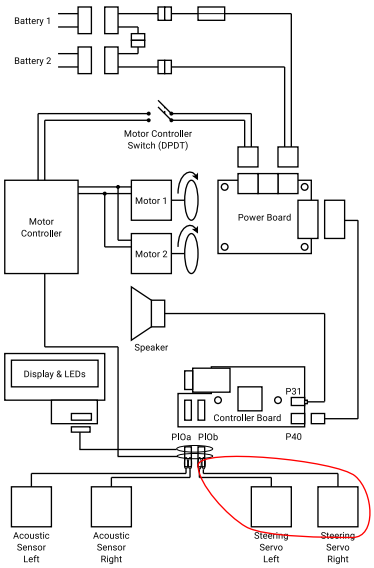


Figure 2.10: Location of the servos

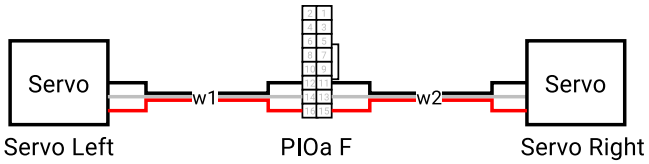


Figure 2.11: Wiring of the servos.

2.7 Servos

This section describes the connection of the two steering servos. Figure 2.10 shows the location of the servos in the wiring overview.

2.7.1 Components

Amount	Description	Designators	Origin
2	High-Torque Servo 2056	Servo Left, Servo Right	In the car: Traxxas servo 2056
6	JST SPUD-001T-P0.5		RS No. 906-0275
1	JST JST PUDP-16V-S	PIOa F	RS No. 906-0256

2.7.2 Wiring

Figure 2.9 shows the connection between the controller and the servos.

2.7.3 Construction

- Gather all components;
- Cut the servo connectors from cables w1 and w2;
- Strip the cable ends;
- Crimp JST SPUD-001T-P0.5 connectors onto the cable ends and plug into connector PIOa F;

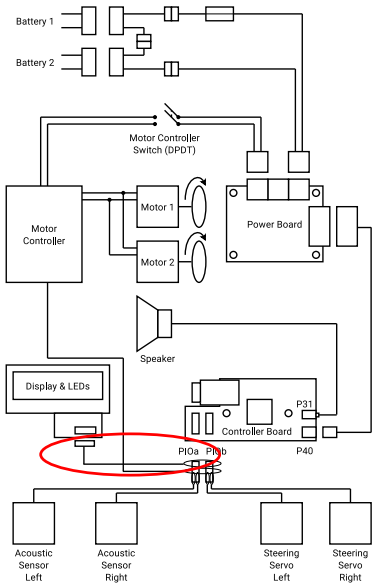


Figure 2.12: Location of the display cable

2.8 Display and LEDs cable

This section describes the display cable that is used to connect the display board to the controller board. Figure 2.12 shows the location of the display cable in the wiring overview.

2.8.1 Components

The display cable consists of the following components:

Amount	Description	Designators	Origin
24	JST SPUD-001T-P0.5		RS No. 906-0275
3	JST PUDP-16V-S	PIOa F, PIOb F, P_IN F	RS No. 906-0256

2.8.2 Wiring

Figure 2.13 shows the diagram of the assembly of the display and leds cable.

2.8.3 Construction

- Gather all components and cat all cables to length;
- Crimp JST SPUD-001T-P0.5 connectors onto all cable ends;
- Plug all cables into the JST PUDP-16V-S connectors.

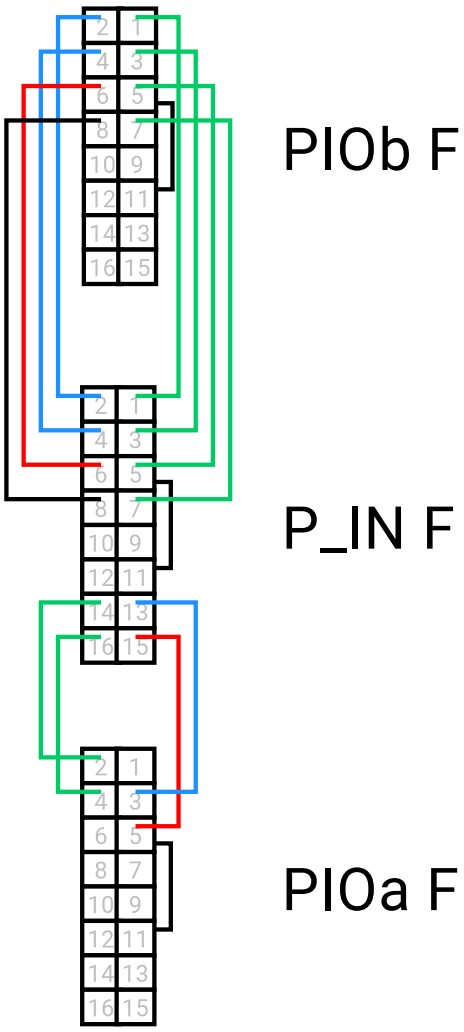


Figure 2.13: Wiring diagram display and leds cable

Chapter 3

EPO-4 Car Firmware and Programming Controller

3.1 Introduction

This chapter describes the firmware and the process of programming the controller board.

3.2 Firmware Structure

This section describes the structure of the firmware of the EPO-4 car controller.

3.2.1 Hardware

The controller used is the ARM LPC4337FBD144. This microcontroller is a dual cpu M0 and M4 with a shared memory. The main controller functionality, including the sensors readout, servo control, and motor control, is implemented in the M4 core. The audio beacon is implemented in the M0 core.

3.2.2 M0 Core: Audio Beacon

The M0 core is used to implement the audio beacon. The M0 core runs completely independent of the M4 core. On startup the M0 core will read in the configuration values for the audio beacon from a part of the shared memory of both cores. The motor PWM is used to generate the carrier frequency, timer 3 is used to control the motor PWM based on the audio code and the repeat counter.

On a change of the audio beacon configuration the main controller in the M4 core can restart the M0.

3.2.3 M4 Core: Main Controller

The M4 core runs the main controller. This controller will handle the following tasks:

- Communicating with the Bluetooth controller;

- Reading the acoustic distance sensors;
- Reading the battery voltage;
- Controlling the steering servos;
- Controlling the motor controller;
- Handling user input;
- Output status information on the display and the LEDs.

Bluetooth: UART0

Bluetooth communication is handled by the UART. In `config.c` the bluetooth pins and the UART are configured. In `bluetooth.c` the actual UART communication functions can be found. Incoming bytes are handled by an interrupt handler (`interrupt.c`) and placed in the read buffer. In the main loop input from the Bluetooth module is read in `epo4_cmd.c` and subsequently processed.

Acoustic Sensors: I2C

The acoustic sensors are read out via I2C. This is handled via the Keil I2C driver. Initialization is done in `peripherals.c`. The sensors are read out using polling on the I2C communication (`sensors.c`). These routines are in turn called every 200 ms via the timer 0 interrupt handler (`interrups.c`). The read in values are stored for later use in updating the display or reporting to the user via the Bluetooth communication.

Battery Voltage: ADC

The battery voltage is monitored via the ADC. The ADC is initialized in `configs.c`. The ADC is used with polling from the interrupt handler of timer 0 (`interrupts.c`).

Steering Servos: Motor PWM

The steering servos are actuated by adjusting the value in match register MAT0 of the motor PWM block in `actuators.c`.

Motor Controller: Motor PWM

The motor controller is controlled with a PWM signal. This signal is also generated by the motor PWM block, now by adjusting the value in match register MAT1 (`actuators.c`).

User Input

User input is received in the UART interrupt handler and stored in the receive buffer (`interrupts.c`). The main loop (`main.c`) calls the function CMD_ReadFromUart() in `epo4_cmd.c` in which a command is read in. When a valid command is found, it is parsed in CMD_doCmd().

Status Output: Display and LEDs

The display is used to show status info of the car. The display update function (`actuators.c`) is called from the main loop (`main.c`). The power LEDs are direcly connected to the incoming power lines. The Bluetooth LEDs are directly driven by the Bluetooth module.

3.3 Firmware Building and Programming

This section describes the software to build the firmware and upload the code to the EPO-4 car controller.

3.3.1 Building the Firmware

The firmware is build in the μVision environment from Keil. A screenshot of the program with an open project is shown in Figure 3.1.

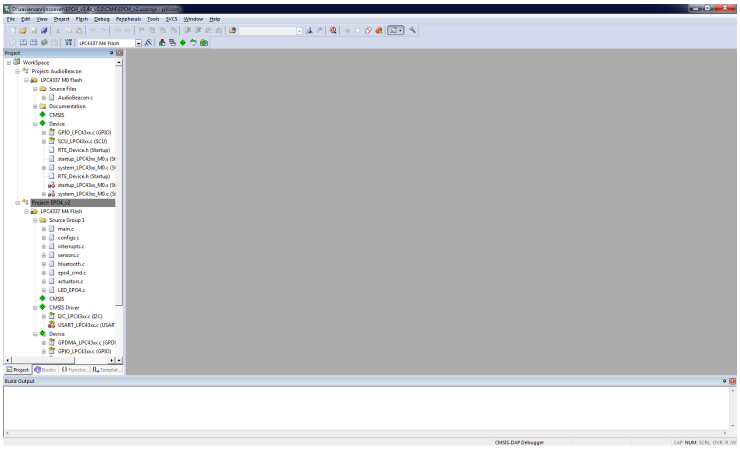


Figure 3.1: Screenshot of μVision

Load the project file (`EPO4_v2.uvmpw`) via Project→Open Project....

The left most part of the program shows the project tree. The EPO-4 firmware contains two programs:

- One project for the main controller that will execute on the M4 core;
- One project for the audio beacon that will execute on the M0 core.

The currently active project is denoted by a gray bar with the project title. You can change the active project by right-clicking the project name and selecting Set as active project. This is shown in the screenshot in Figure 3.2.

The currently selected project can be build using the Build target files button (or by pressing F7). This is shown in the screenshot of Figure 3.3.

Make sure to build *both* projects!

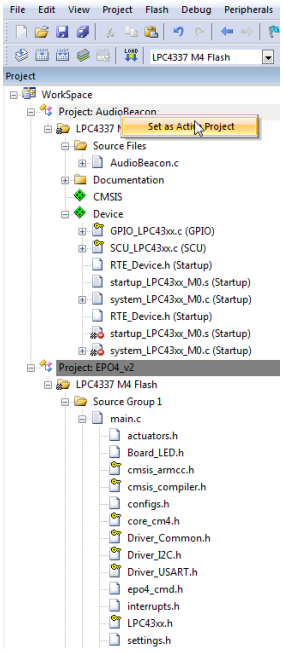


Figure 3.2: Screenshot of μVision: setting the active project

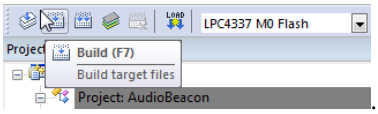


Figure 3.3: Screenshot of μVision: build target files

3.3.2 Programming the Firmware

When the both projects are succesfully build, you can program the microcontroller. Connect the NXP LPCXpresso LPC-Link 2 to the controller board, see the photo in Figure 3.4. Also connect power to the controller board via either a lab power supply or the power PCB in the EPO-4 car.

Unkeyed header on new PCBs
Please note that the new PCBs are not equiped with a keyed programming header, so take extra care to properly connect the programming cable!

Select the project you want to upload to the microcontroller and press the Download button (or press F8). This is shown in the screenshot of Figure 3.5.

Make sure to program both cores by activating both projects and uploading the program to the microcontroller.

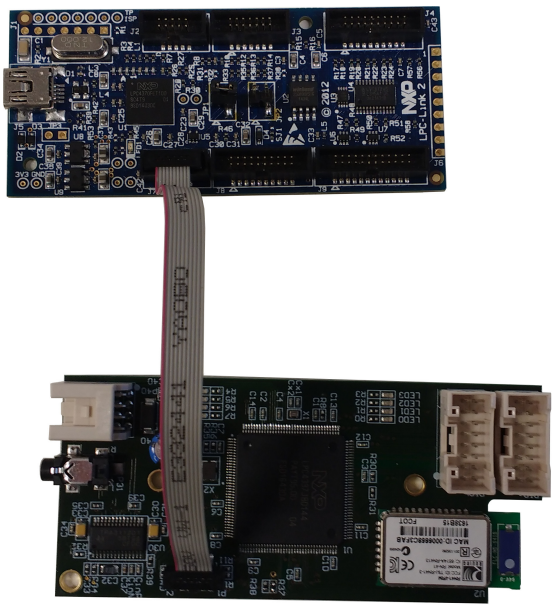


Figure 3.4: Photo of the connection between the programmer (on top) and the controller board

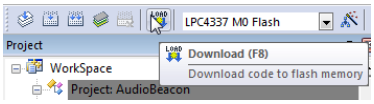


Figure 3.5: Screenshot of μVision: programming the microcontroller board

Part II

Testing

Chapter 4

Controller Communications Setup

4.1 Introduction

The communication with the controller takes place over a serial connection over Bluetooth. This chapter explains the settings and the steps (both hardware and software) required to setup a connection between the host computer and the controller board.

4.2 Specifications

The controller uses the following setup:

Parameter	Setting
Baud rate	115.200 bps
Data format	8.1: 1 start bit, 8 data bits, 1 stop bit, no parity
Flow control	Hardware flow control: RTS/CTS

Commands are terminated with a linefeed (LF, ascii 0x10). For commands accepted by the controller, see Chapter 5.

4.3 Setting up Communications under Linux

Root access
To setup a connection to the controller board under the Linux OS, root access (sudo) is required!

4.3.1 Pairing Bluetooth Devices

Power the controller board and plug in the Bluetooth module into a USB port on the host PC. Now open the Bluetooth program on the host PC via Application→System Tools→Bluetooth. This should open the Bluetooth program (see Figure 4.1).

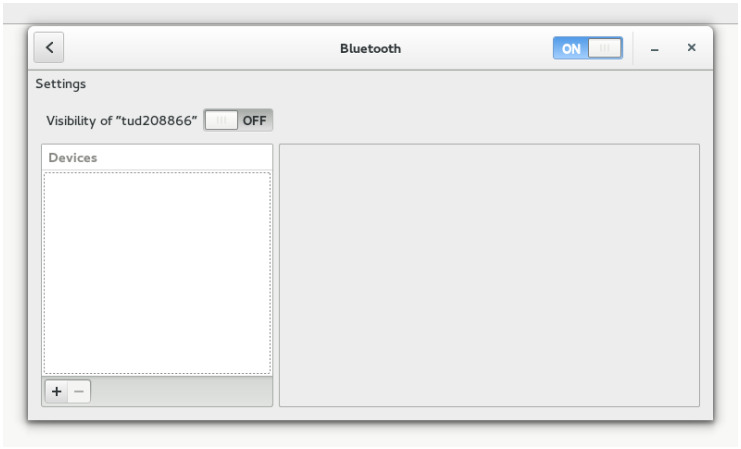


Figure 4.1: Screenshot of the Bluetooth program

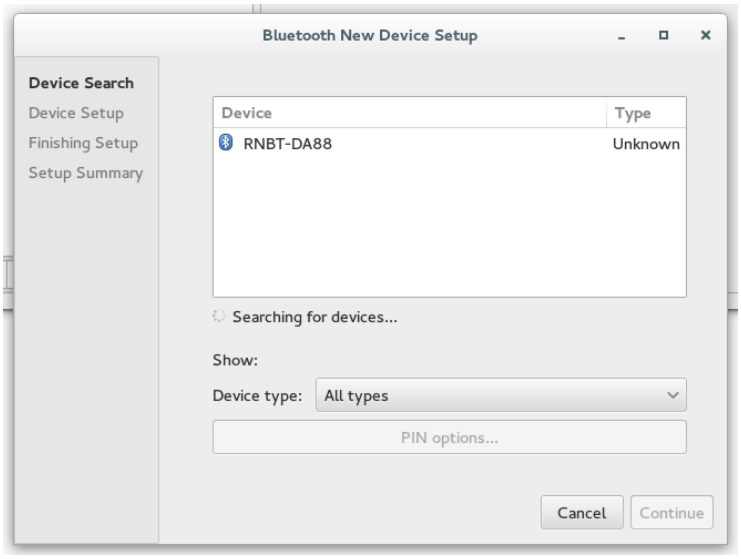


Figure 4.2: Screenshot of the Bluetooth New Device Setup: Device Search dialog

Press the + button in the lower right corner to add a new Bluetooth device. This will open the dialog shown in Figure 4.2.

The host computer will now search for new devices and show any devices found in the dialog. Select the appropriate device and press Continue to proceed. The host computer will now try to pair with the selected Bluetooth device. If connecting is succesful, you will be shown a similar dialog as in Figure 4.3 in which you have to confirm the PIN.

Press Matches to confirm the PIN and proceed. This will complete the pairing procedure and, if succesful, present you the dialog of Figure 4.4.

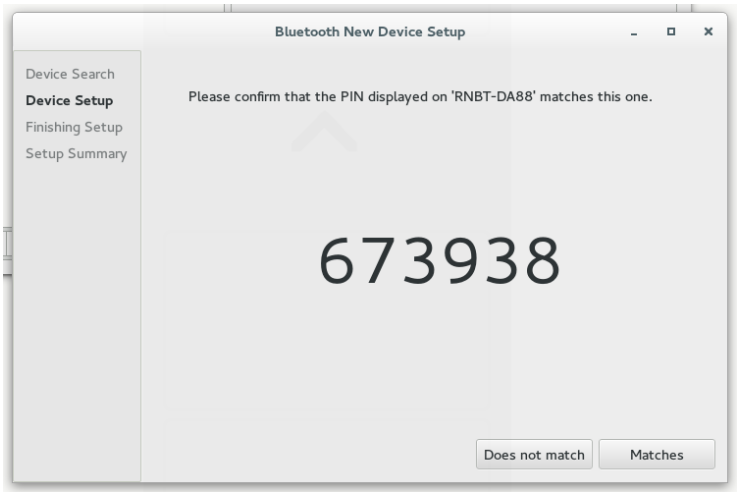


Figure 4.3: Screenshot of the Bluetooth New Device Setup: Device Setup dialog

Figure 4.4: Screenshot of the Bluetooth New Device Setup: Setup Summary dialog

4.3.2 Opening a Serial Connection over Bluetooth

Now that the Bluetooth device is paired, you can open a serial connection to the controller.

- Open a terminal window;
- Use the following command to open the serial connection:
  `sudo rfcomm connect /dev/rfcomm?  <MAC-ID>`
  In this command, use a number for `?` and replace `<MAC-ID>` with the MAC-ID of the paired device.

If the command is succesful, you will see a message like:

```
Connected /dev/rfcomm?  to <MAC-ID> on channel 1
Press CTRL-C for hangup
```

you can proceed to §4.3.3. If, however, the command fails with the following message:

```
Can't create RFCOMM TTY: Address already in use
```

increment the number and retry (as many times as required).

4.3.3 Communicate over the Serial Connection with a Serial Terminal Program

Now that the serial connection has been opened, you can use a serial terminal program to communicate with the controller. Any serial terminal program will do, provided the correct settings are used (see §4.2). In this manual, the program `picocom` will be used, see Appendix **??** for more information on this program.

- Open a *new* terminal window;
- Use the following commano to start `picocom`:
  `sudo picocom --baud 115200 --flow h /dev/rfcomm?  --omap crlf --imap lfcrlf --emap crcrlf --echo`
  Replace the `?` with the number used to open the connection.

This will open serial communications according to the setup described in §4.2 with local echo (`--echo`) You can now type in command for the controller. You can exit the program with Ctrl a - x .

4.4 Setting up Communications under Windows

Admin access
To setup a communications with the controller board under the Windows OS, additional software has to be installed. This requires administrative rights!

4.4.1 Pairing Bluetooth Devices

Power the controller board and plug in the Bluetooth module into a USB port on the host PC. Now add the Bluetooth device on the host PC via the systray (lower right corner) Bluetooth icon→Add a device. This should open the dialog of Figure 4.1.

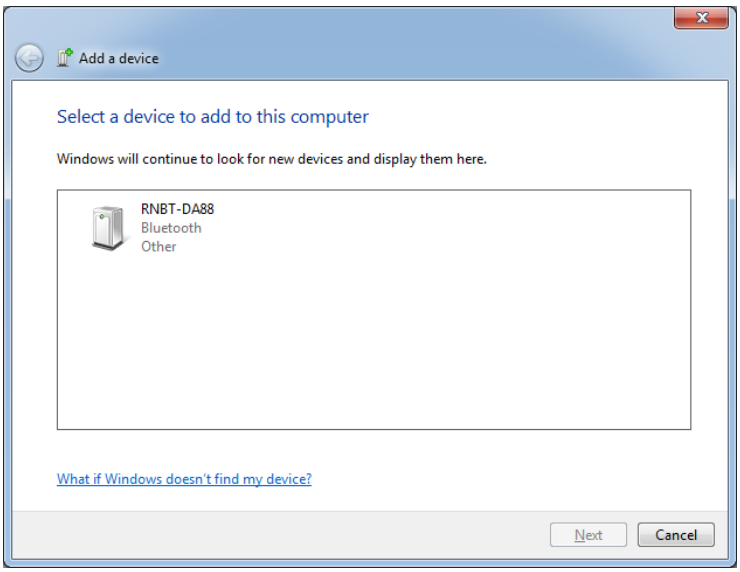


Figure 4.5: Screenshot of the Bluetooth Add a device dialog

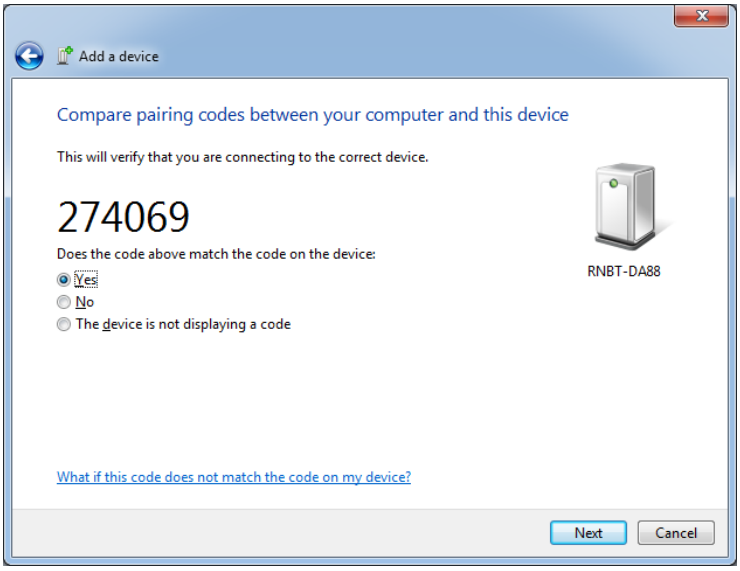


Figure 4.6: Screenshot of the Bluetooth Add a device: Confirm PIN dialog

The host computer will now search for new Bluetooth devices and show any device in the dialog. Select the appropriate device and press Next to proceed. The host computer and the controller will now try to pair with the selected Bluetooth device. If connecting is succesful, you will be shown a similar dialog as in Figure 4.6 in which you have to confirm the PIN.

Press Next to confirm the PIN and proceed. This will complete the pairing procedure and, if succesful, present you the dialog of Figure 4.7

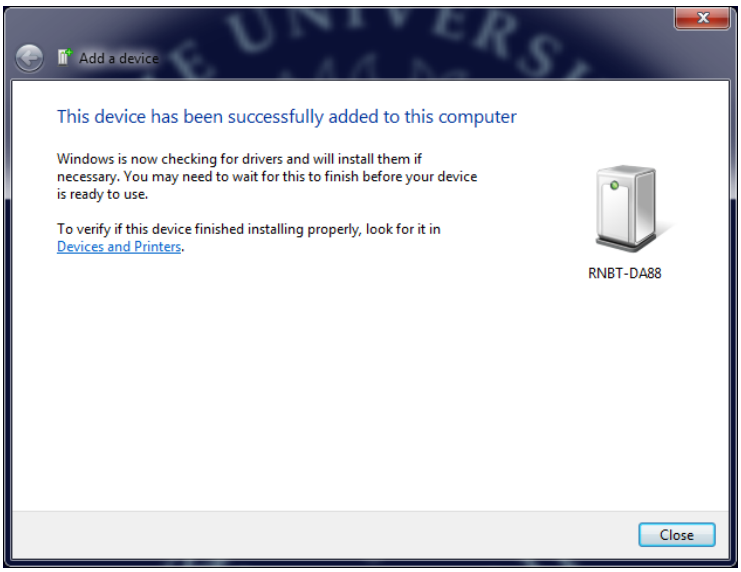


Figure 4.7: Screenshot of the Bluetooth Add a device: Finished dialog

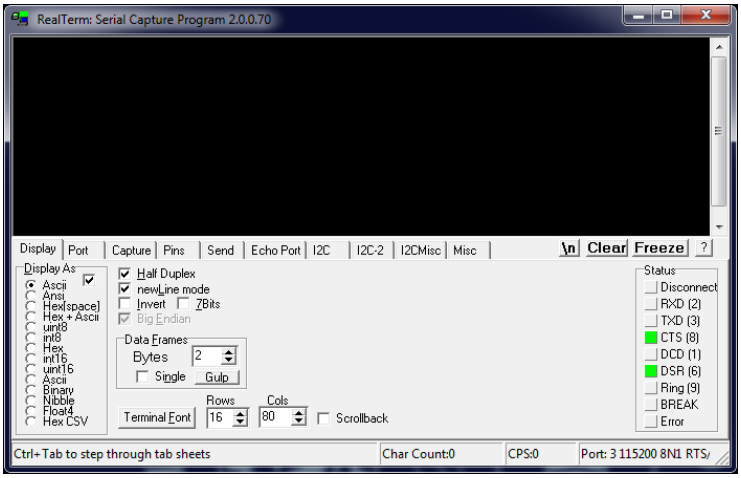


Figure 4.8: Screenshot of the `RealTerm` program with the Display tab shown

4.4.2 Communicate over the Serial Connection with a Serial Terminal Program

For communications, the program `RealTerm` is used. A number of settings in the program have to be adjusted. The first settings are on the Display tab as shown in Figure 4.8. Change the following settings:

- Tick Half Duplex;
- Tick newLine mode.

In order to setup the port correctly, you have to know which ports are assigned to the Bluetooth adapter. To see these ports, select on the systray Bluetooth icon→Open

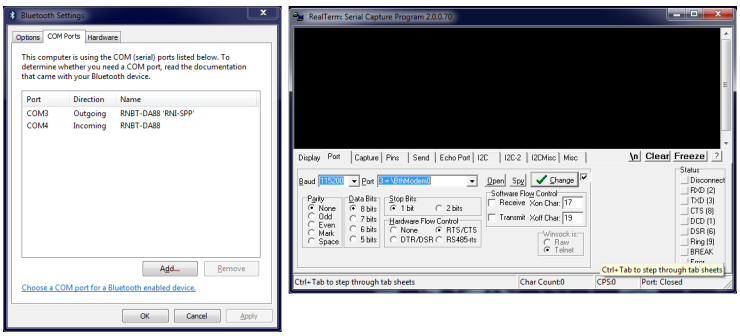


Figure 4.9: Screenshot of the Bluetooth serial ports dialog and `RealTerm` program with the Port tab shown

Settings and select the tab COM Ports. Figure 4.9 shows the settings on the Port tab in `RealTerm` and next to the COM Ports dialog. In `RealTerm` select the outgoing port number. Use the following settings:

- Set Baud to 115200;
- Set Parity to none;
- Set Data Bits to 8 bits;
- Set Stop Bits to 1 bit;
- Set Hardware Flow Control to RTS/CTS;

Now the port can be opened with the Open button.

With the port opened, the Send tab can be used to send commands to the controller. On the Send tab tick the boxes labeled +LF in order to terminate a command with a newline character (see Figure 4.10. Now you can type in some commands in the one of the two text boxes and send the command to the controller with the Send ASCII button. Replies from the controller are shown on the terminal window.

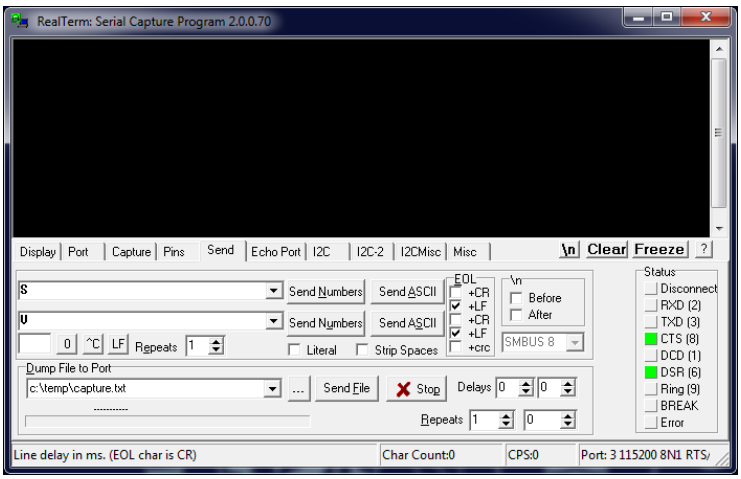


Figure 4.10: Screenshot of the `RealTerm` program with the Send tab shown

Chapter 5

Controller Commands

5.1 Introduction

Communication with the controller board is done via a serial channel over Bluetooth. This chapter shows the commands that can be send to the controller.

5.2 Command Structure

Each command has the following format:

```
C dddd \n
```

The different parts have the following meaning:

Part	Description
`C`	Command
`dddd`	An optional parameter of 0 to 4 bytes
`\n`	Terminating newline character (ASCII 10)

5.3 Available Commands

The following commands are available:

`A`	Audio enable/disable Enable or disable the audio beacon
`Ax\n`	char `x`=`1` enable `x`≠`1` disable
`B`	Set bit frequency Audio beacon setting: set bit frequency of beacon code
`Bxx\n`	uint16_t `xx` is bit frequency in Hz (default = 5000)
`C`	Set audio code Audio beacon setting: set 32 bit audio code

`Cxxxx\n` uint32_t
`xx` is the audio code (default = 0x00000000)

`D` Set direction PWM
Set PWM value of steering servos
`Dxxx\n` char[3]
`xxx` is the PWM value as promille (100-200, default = 150)

`F` Set carrier frequency
Audio beacon setting: set the carrier frequency of the beacon
`Fxx\n` uint16_t
`xx` is the carrier frequency in Hz (default = 15000)

`M` Set motor controller PWM
Set PWM value of motor controller
`Mxxx\n` char[3]
`xxx` is the PWM value as promille (135-165, default = 150)

`R` Set repetition counter
Audio beacon setting: set repetition counter. This value represents the number of bit transmission times the beacon halts before retransmitting the code. $f_{rep} = \frac{f_{bit}}{\text{rep count}}$
`Rxx\n` uint16_t
`xx` is the repetition counter value (min. = 32)

`S` Get status
Get status of the EPO-4 car, including audio beacon settings, PWM values for steering servos and motors, and the values returned by the sensors
`Sx\n` char
`x='d'` only report sensor values:
`USLxxx`
`USRxxx`
Values returned in cm. Value of 999 denotes overflow.

`x='v'` only report battery voltage:
`VBATTxx.xV`
Value returned in V.

`x=''` full status report

```
*************************
* Audio Beacon:  on
* c:  0x00000000
* f_c:  xxxxx
* f_b:  xxxxx
* c_r:  xxx
*************************
* PWM:
* Dir.  xxx
```

```
* Mot.  xxx
*************************
* Sensors:
* Dist.  L xxx R xxx
* V_batt xx.x V
*************************
```

V Get version

Get version number and revision date of the firmware for the main controller and the audio beacon

```
******************************
*            EPO-4           *
******************************
*  KITT Firmware Rev.  3.4b   *
* Audio Firmware Rev.  2.0b   *
******************************
*   Author:M.E. Weskin B.Sc.  *
*   Date:      May 9, 2016    *
******************************
```

Chapter 6

Testing the EPO-4 Controller Board Functions

6.1 Introduction

The previous chapters describe the methods to connects to the controller board (Linux in §4.3 and Windows in §4.4) and the commands that are available (§5.3). This chapter briefly describes steps to test the functionality of the controller board and the EPO-4 car.

6.2 Testing the connection

The commands `V` and `S` can be used to test the connection with the controller board.

6.3 Testing the Sensors

The commands `S` or `Sd` can be used to get the current value of the sensors. The sensors can detect an object at a minimum distance of 15 cm. Note that the display also shows the distance of both sensors.
If the readout seems wrong, check the following:

- Are all cables connected (connect any missing/broken cables);
- Do the sensors have a different address? This can be tested by disconnecting one of the sensors (remove the wires from the connectors on PIOb). If both sensors work this way, then both sensors have an identical address. Make sure the left sensors has address 112 and the right sensor has address 113.

6.4 Testing the Battery Voltage Readout

The commands `S` or `Sv` van be used to get the current value of the battery voltage. If the value is incorrect, most likely the ADC connection to the microcontroller is damaged. It is not possible to repair this damage.

6.5 Testing the Audio Beacon

Before enabling the audio beacon, a code must be set first:

- Use the command `C1234` to specify a suitable code;
- Enable the beacon with `A1`;
- Disable the beacon with `A0`.

The `S` command shows the internal state of the audio beacon. This enables you to verify if the audio beacon is working properly if no sound is audible. If the beacon is enabled but no sound is produced, either:

- the M0 core of the controller is not programmed (program the M0 core);
- the audio beacon is broken (check signal on cable).

With the commands `B`, `F`, `R` parameters of the audio beacon can be modified. The `S` command can be used to verify the current settings.

6.6 Testing the Steering Servos

The steering servos can be tested with the `D` command:

- `D100` turns the servos maximally right;
- `D200` turns the servos maximally left;
- `D150` set the servos in straight posistion;

6.7 Testing the Motors

The motors can be tested with the `M` command. Make sure the motor controller is powered by sliding the slide switch in the correct position.

Make sure the EPO-4 car wheels can move freely when testing the motors. The car is very fast and if the wheels are on the groud, the car might quickly accelerate into a wall!

- `M165` drives the motors full-speed forwards;
- `M150` stops the motors.
- `M135` drives the motors full-speed backwards;

The motor controller has a dead zone implemented, meaning that very small deviations from 150 will not turn the motors on at all. Furthermore, the car has a high inertia that makes it impossible to start the car with a low PWM value.

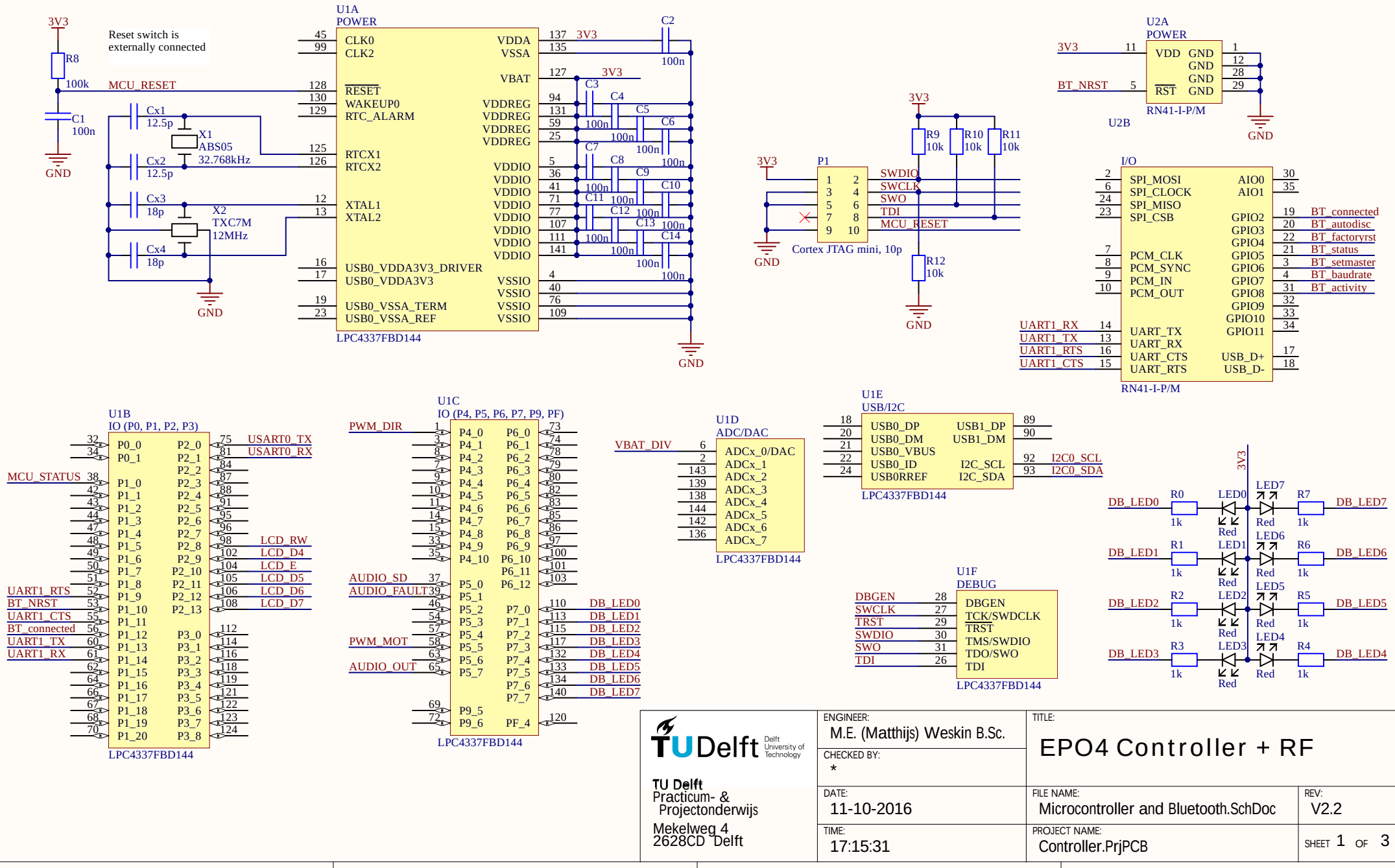
Part III

Appendices

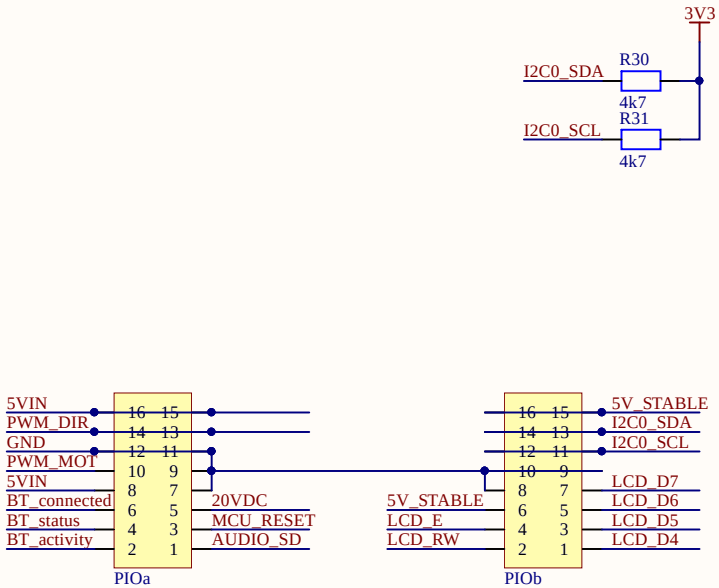
Appendix A

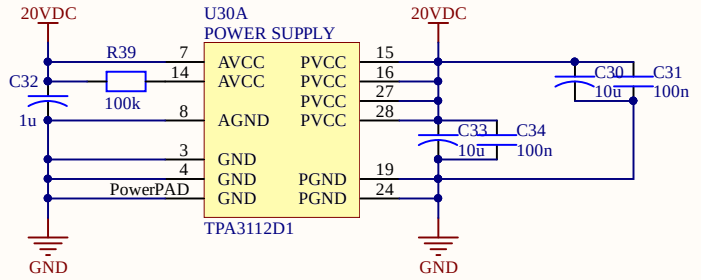
Schematic of the Controller V2.2

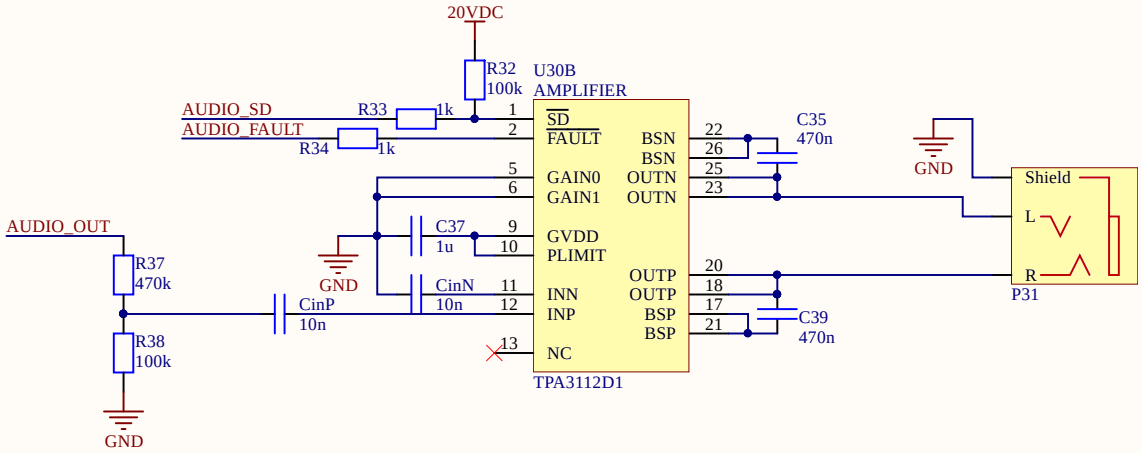
These pages contain the schematics of the controller v2.2.



TU Delft Delft University of Technology TU Delft Practicum- & Projectonderwijs Mekelweg 4 2628CD Delft	ENGINEER: M.E. (Matthijs) Weskin B.Sc. CHECKED BY: *	TITLE: EPO4 Controller + RF	
	DATE: 11-10-2016 TIME: 17:15:31	FILE NAME: Microcontroller and Bluetooth.SchDoc PROJECT NAME: Controller.PrjPCB	REV: V2.2 SHEET 1 OF 3

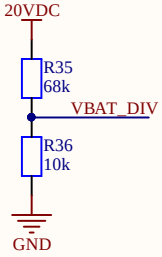






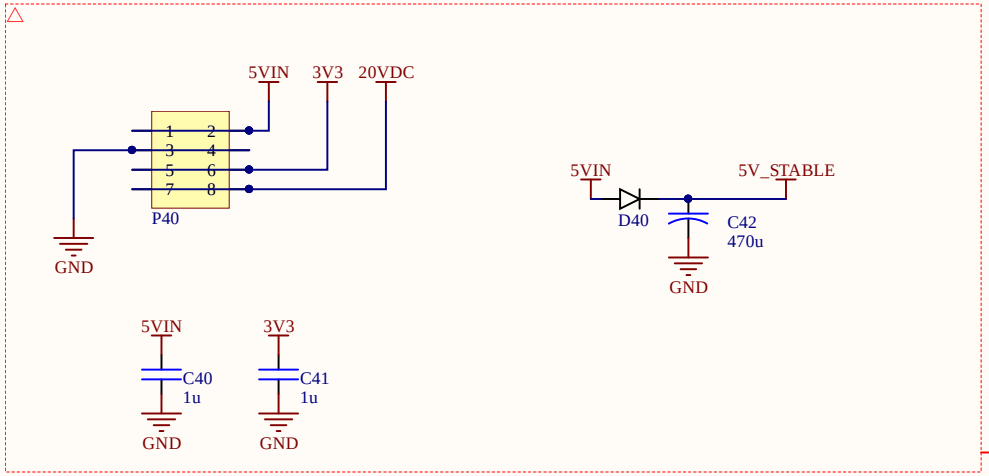
Resistive divider to scale the battery voltage (0-20VDC) to the maximum input voltage of the analog-to-digital converter (0-3.3V).

High resistors must be used in order to keep te current flow as low as possible.



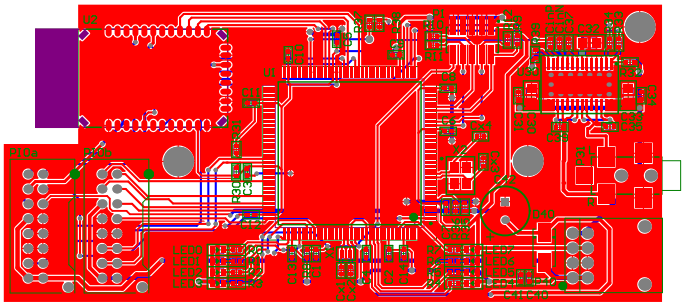
TU Delft
Delft University of Technology
TU Delft
Practicum- & Projectonderwijs
Mekelweg 4
2628CD Delft

ENGINEER: M.E. (Matthijs) Weskin B.Sc.	TITLE: EPO4 Sensors & Actuators	
CHECKED BY: *		
DATE: 11-10-2016	FILE NAME: Sensors and Actuators.SchDoc	REV: V2.2
TIME: 17:15:32	PROJECT NAME: Controller.PrjPCB	SHEET 2 OF 3



TU Delft Delft University of Technology TU Delft Practicum- & Projectonderwijs Mekelweg 4 2628CD Delft	ENGINEER: M.E. (Matthijs) Weskin B.Sc. CHECKED BY: *	TITLE: EPO4 Power Management	
	DATE: 11-10-2016 TIME: 17:15:33	FILE NAME: Power Management.SchDoc PROJECT NAME: Controller.PrjPCB	REV: V2.2 SHEET 3 OF 3

Layer	Name	Material	Thickness	Constant	Board Layer Stack
1	Top Overlay				
2	Top Solder	Solder Resist	0,40mil	3,5	
3	Top Layer	Copper	1,40mil		
4	Dielectric 1	FR-4	12,60mil	4,8	
5	Bottom Layer	Copper	1,40mil		
6	Bottom Solder	Solder Resist	0,40mil	3,5	
7	Bottom Overlay				



TU Delft

Practicum– & Projectonderwijs
Mekelweg 4
2628CD Delft

ENGINEER: M.E. (Matthijs) Weskin B.Sc.

CHECKED BY: M.E. (Matthijs) Weskin B.Sc.

DATE: 11–10–2016

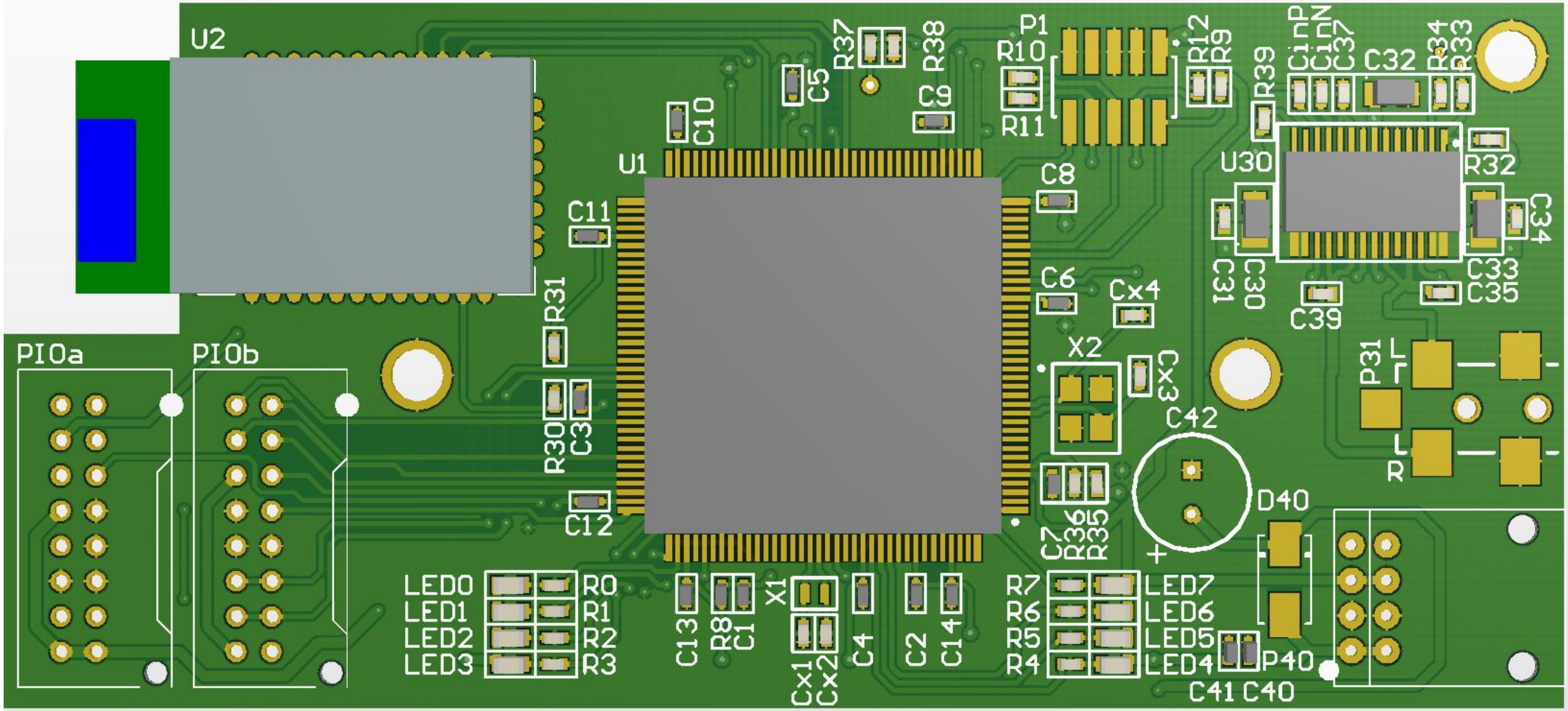
TIME: 17:15:34

TITLE: **EPO4 Main Controller**

FILE NAME: Controller.PcbDoc

REV: 40

SCALE: SCALE: 1.00


U2
U1
PIOa
PIOb
R37
R38
C5
C9
C10
C11
C12
C3
R30
R31
P1
R10
R11
R12
R9
R39
CinP
CinN
C37
C32
R34
R33
R32
U30
C8
C6
Cx4
Cx3
X2
C42
D40
C7
R36
R35
C31
C30
C33
C35
C34
C39
P31
L
R
LED0
LED1
LED2
LED3
R0
R1
R2
R3
C13
R8
C1
X1
Cx1
Cx2
C4
C2
C14
R7
R6
R5
R4
LED7
LED6
LED5
LED4
P40
C41
C40

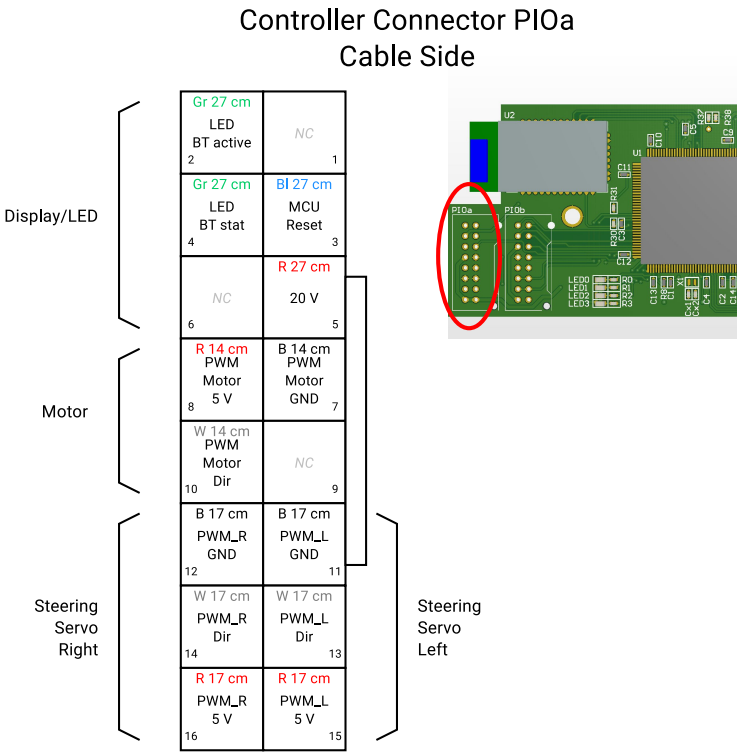
Appendix B

Controller Board Connectors

This appendix describes the connections of the different connectors on the controller board.

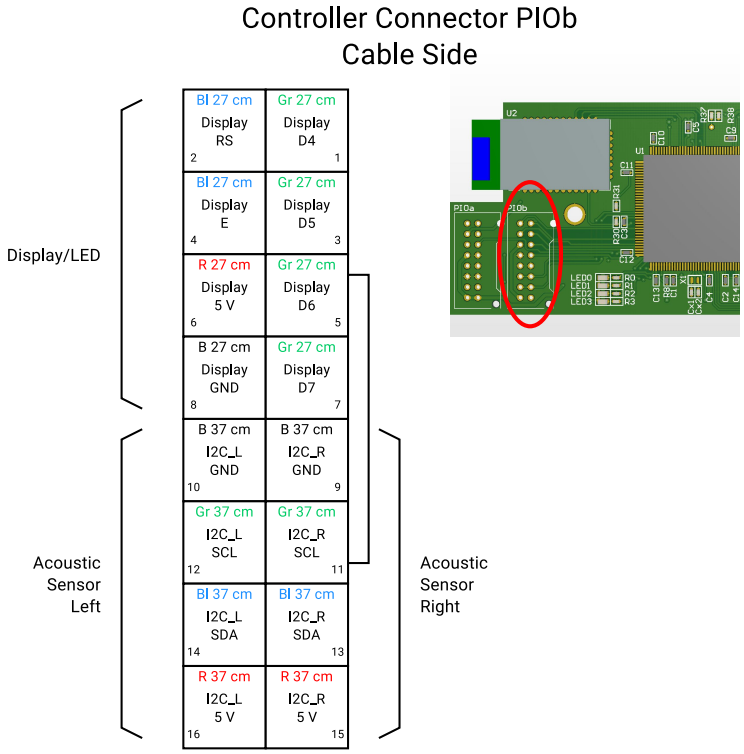
B.1 Connector PIOa

Connector PIOa is used to connect the display, the motor controller and both acoustic sensors.



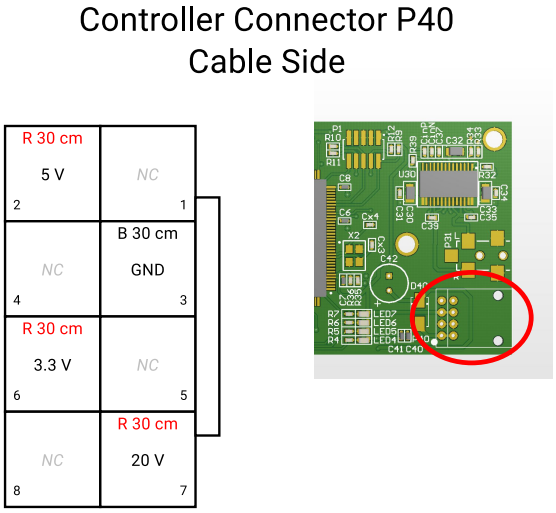
B.2 Connector PIOb

Connecotor PIOb is used to connect the display and both steering servos.



B.3 Connector P40

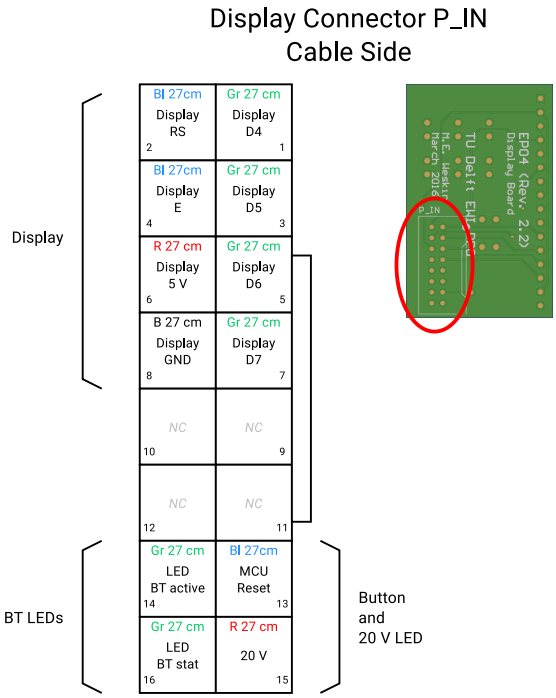
Connector P40 is the power supply connector.



Appendix C

Display Board Connector

This appendix describes the connections of connector P_IN on the display board.

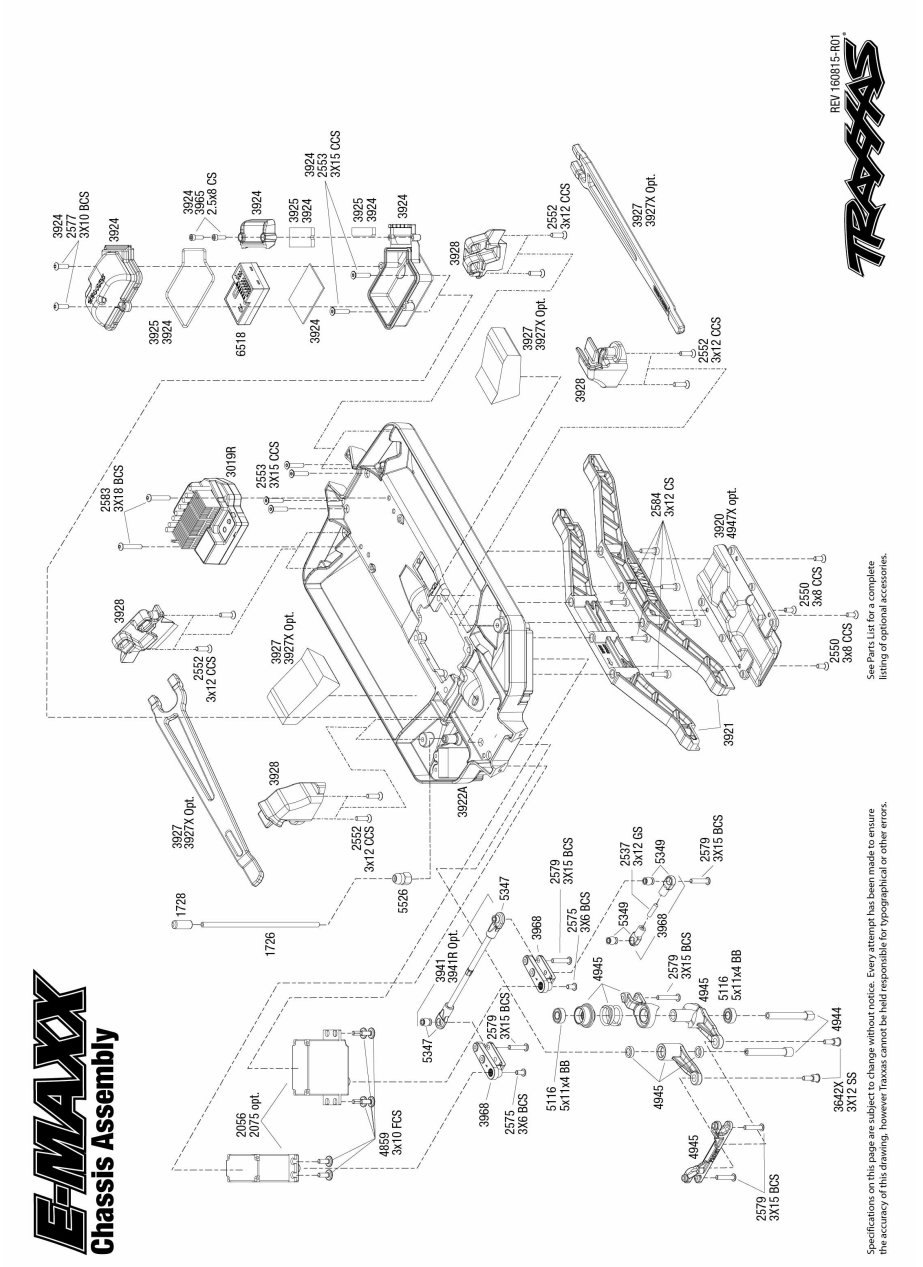


Appendix D

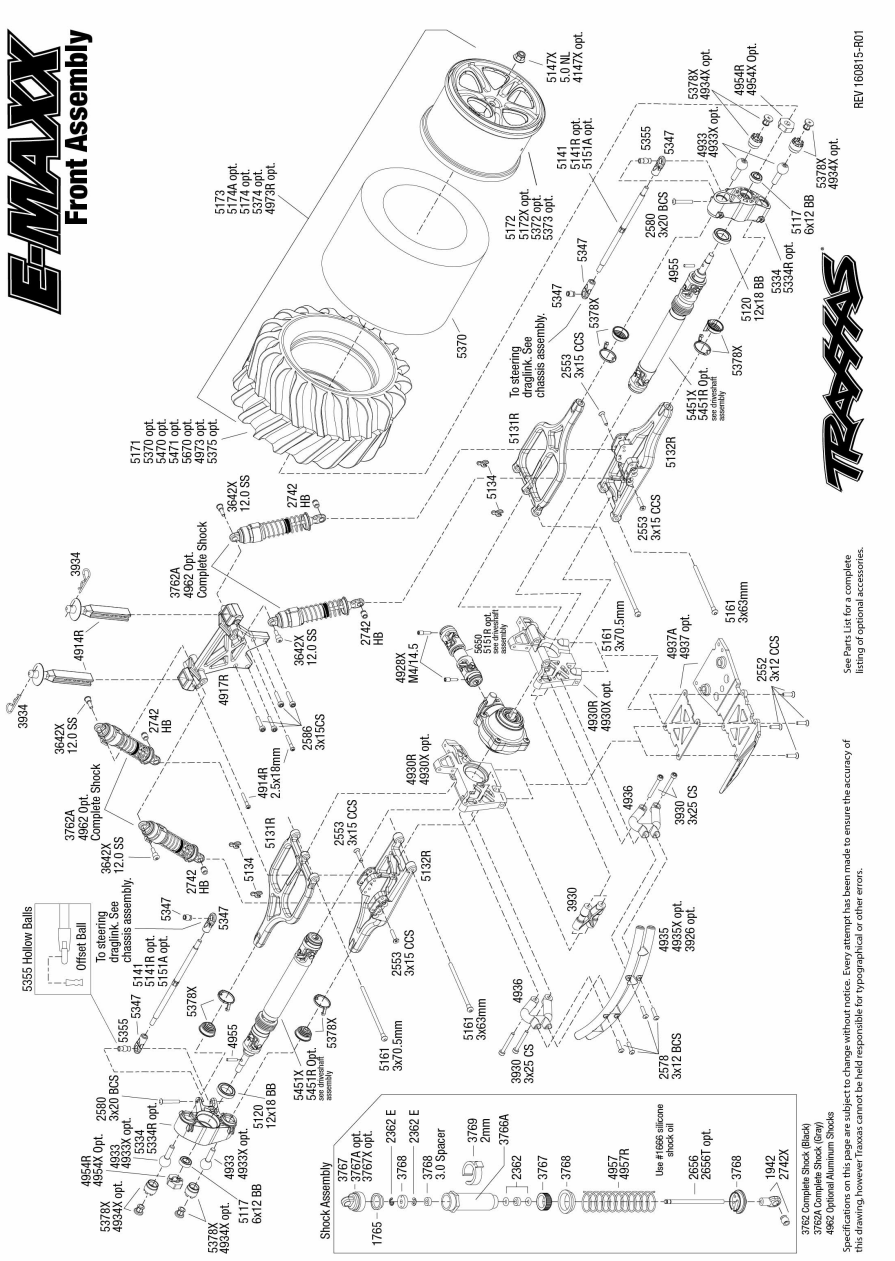
Exploded Views Traxxas E-Maxx

This appendix contains the exploded views of the Traxxas E-Maxx car.

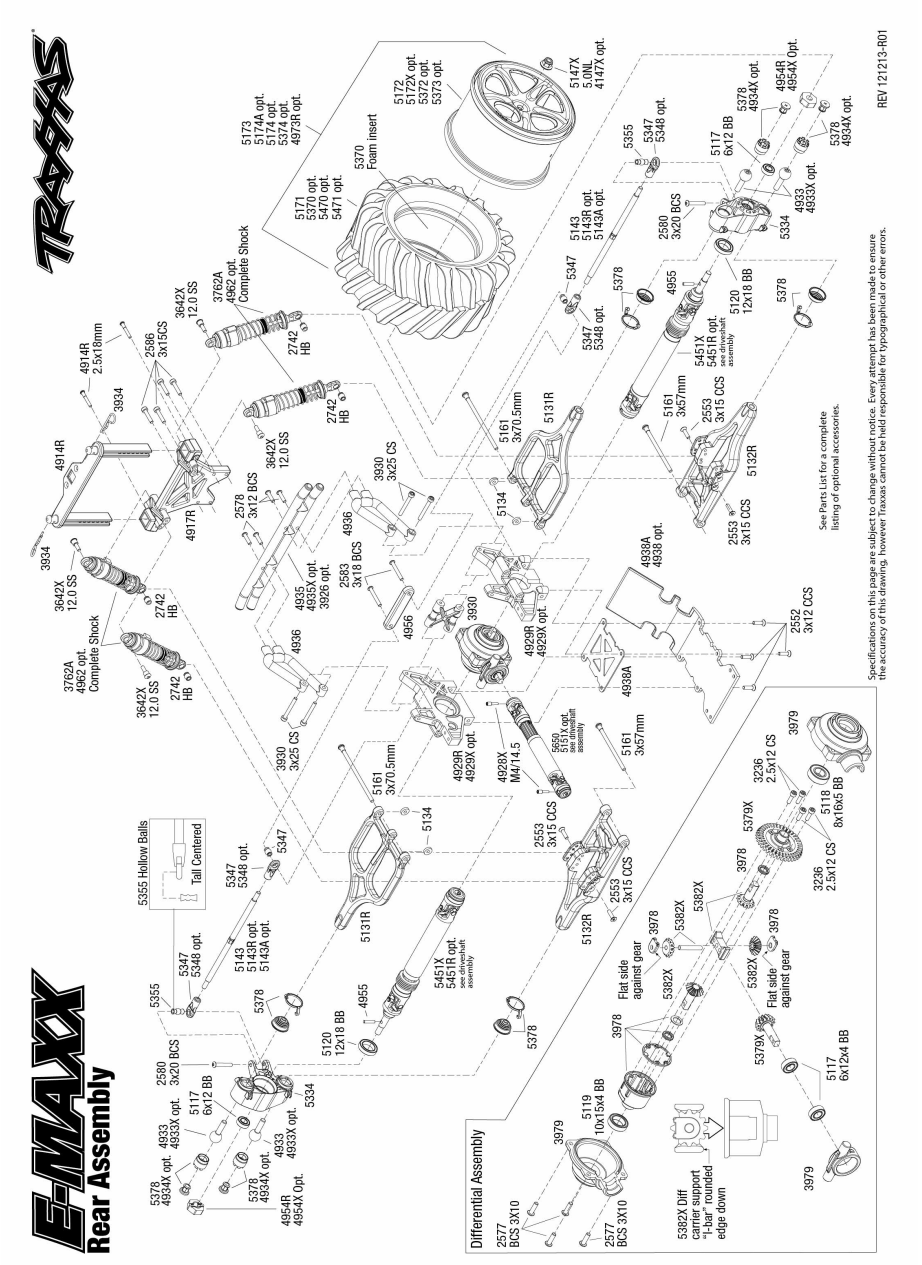
D.1 Chassis Assembly



D.2 Front Assembly



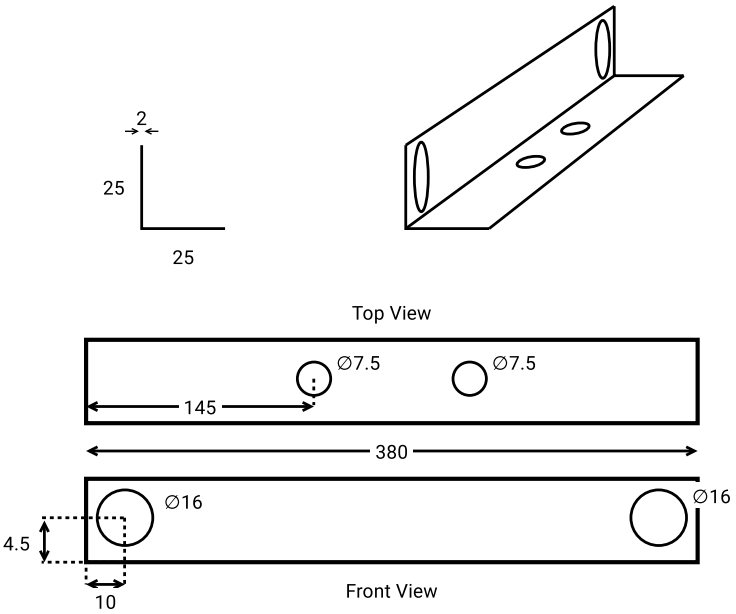
D.3 Rear Assembly



Appendix E

Sensor Bar

This appendix contains the mechanical drawings of the sensors bar. The bar is made out of a piece of aluminium angle bar. All measurements are in mm.



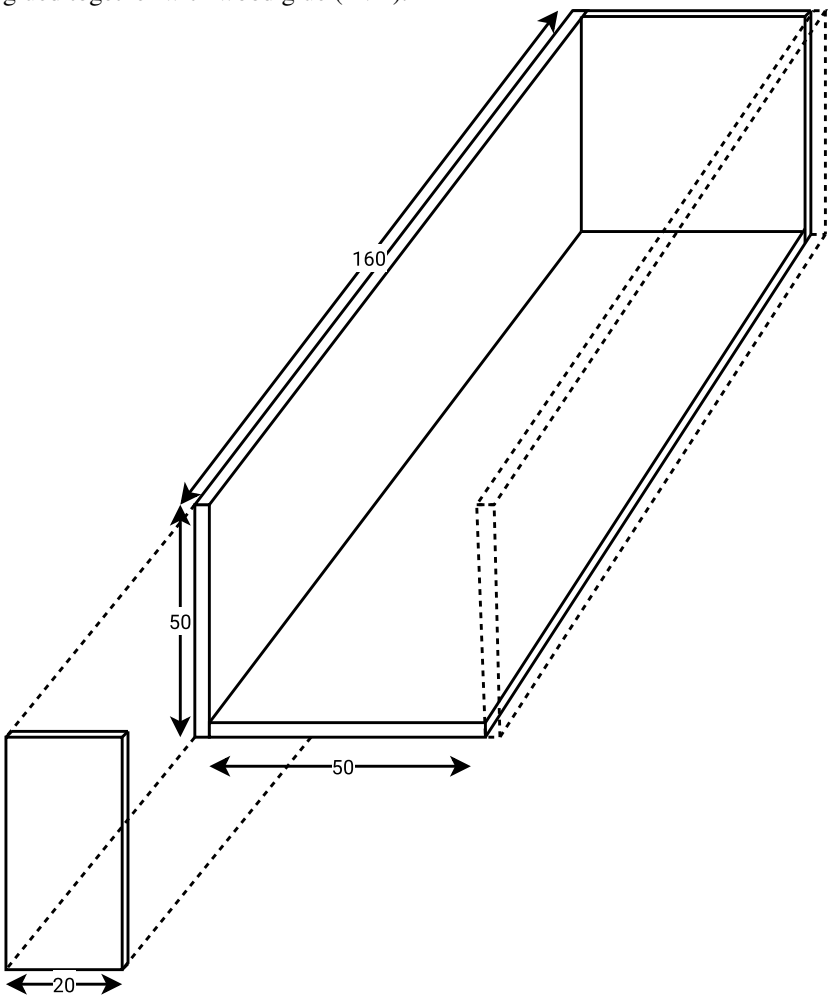
Appendix F

Battery Holder

This appendix contains the mechanical drawings of the battery holder. The battery holder is made out of 4 mm MDF.

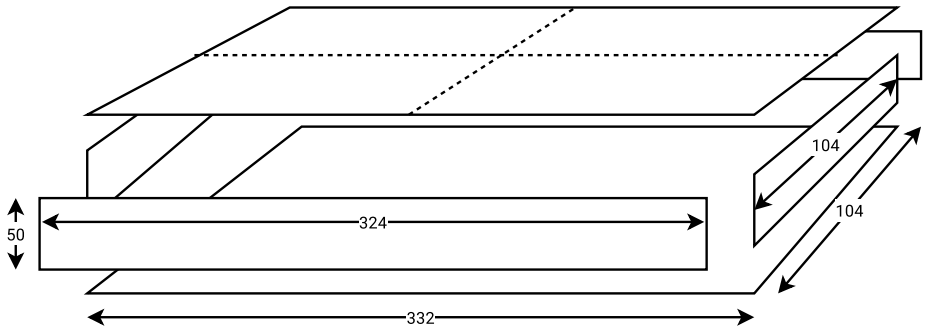
F.1 Single Battery Holder

These are the dimensions of a single battery holder. All dimensions are in mm. Joints are glued together with wood glue (PVA).



F.2 Box for Four Battery Holders

The battery holders are made as a box that contains four single battery holders. All dimensions are in mm. Joints are glued together with wood glue (PVA).

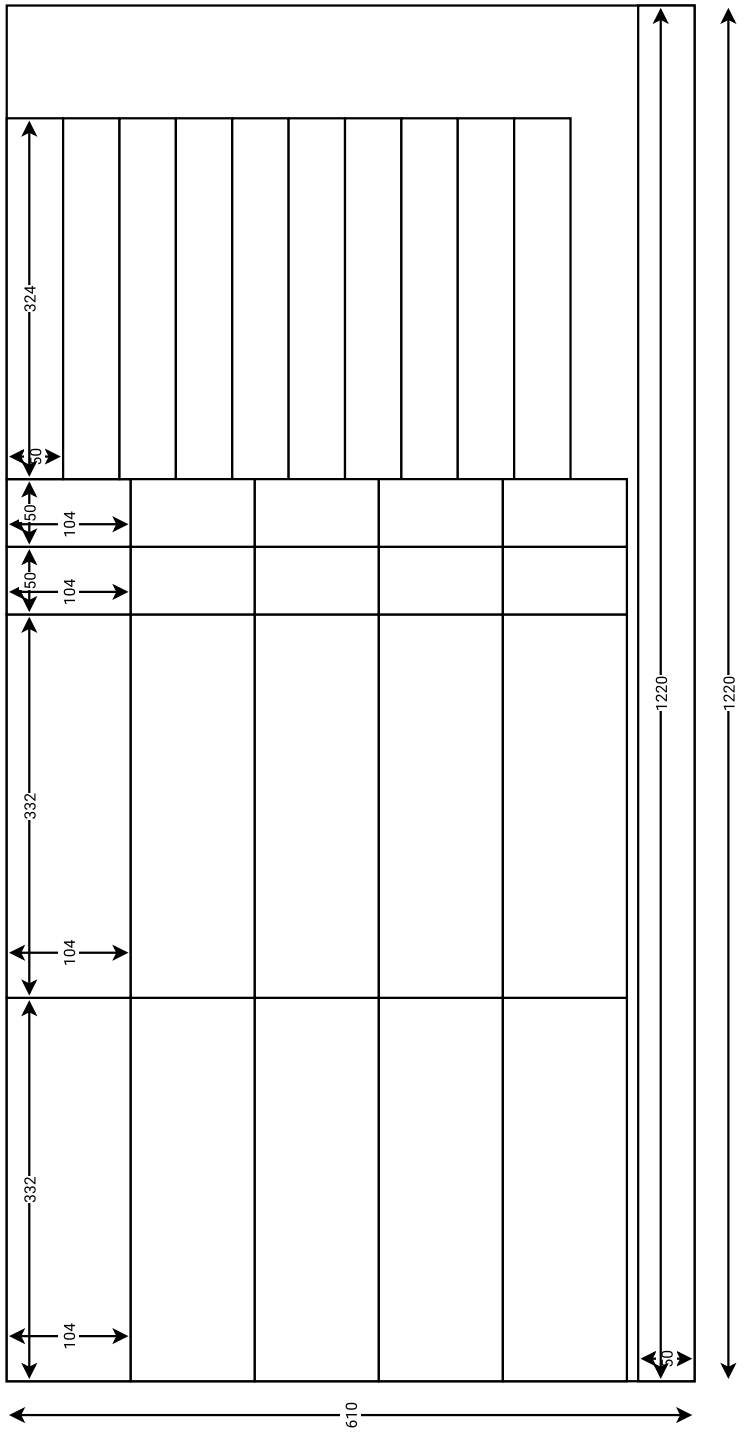


F.3 Cut List and Layout

It is possible to make 20 battery holders from a single 1/4 sheet of 4 mm MDF. The following parts are required:

#	Size [mm]	Description
10	332×104	Top and bottom
10	324×50	Long sides
10	104×50	Short sides
40	50×20	Front sides battery holders

Cut according to this layout (all dimensions in [mm]):



F.4 Updated Version of the Battery Holder

This the updated version of the battery holder that is designed to withstand the force of the moving batteries under full accelleration of the EPO-4 car. All dimensions are in mm, the end-piece is made of 15 mm MDF.