Technical University of Denmark

Henning Si Høj

Platform Integration for Autonomous Self-Driving Vehicles

Master's thesis, June 2017

DTU Electrical Engineering
Department of Electrical Engineering

PLATFORM INTEGRATION FOR AUTONOMOUS SELF-DRIVING VEHICLES
INTEGRATION AF PLATFORM TIL AUTONOMT SELVKØRENDE KØRETØJ

Report written by:
Henning Si Høj

Advisor(s):
Ole Ravn
Jens Christian Andersen
Nils Axel Andersen

DTU Electrical Engineering
Automation and Control
Elektrovej
Building 326
Technical University of Denmark
2800 Kgs. Lyngby
Denmark
Tel: 4525 3576

studieadministration@elektro.dtu.dk

Project period:	30 January 2017 – 30 June 2017
ECTS:	30
Education:	MSc
Field:	Electrical Engineering
Class:	Public
Remarks:	This report is submitted as partial fulfilment of the requirements for graduation in the above education at the Technical University of Denmark.
Copyrights:	

Abstract

In the coming years, autonomous vehicles will bring a paradigm change in the way automobiles are operated. With the propagation of these self-driving vehicles, new challenges arise for which innovative solutions must be found.

The Shell Eco-Marathon will launch a competition for fuel-efficient Autonomous UrbanConcept vehicles in 2018. As part of this master's thesis, a platform of hardware and software has been integrated into the DTU Dynamo car. Sensors and actuators have been designed, implemented and tested, augmenting the car with autonomous capabilities. LIDARs, GPS, IMUs and wheel sensors form the basis for the car's ability to sense its surroundings. The performance of the various sensors has been proven through data collected at the Shell Eco-Marathon 2017, of which a selection is also presented in this thesis. Combined with closed loop control of the actuators, the car can control its speed and heading. The platform created as part of this thesis allows the car to drive a predetermined path and will, with further improvements, enable DTU Dynamo to achieve full autonomy and participate in the upcoming competition.

Keywords

Autonomous automobiles, Self-Driving cars, Automated vehicles, Sensors, Actuators, Remotely operated vehicles, Driverless cars, Mobile Robots, Fuel-efficient driving

Contents

Section Overview

1 Introduction and Motivation – introduces the background and the motivation behind the project and describes the goals that need to be reached.

2 Analysis and Design – describes the challenges of the project, analyzes existing work and research, and deduces the choice of solutions to overcome these challenges.

3 Implementation – describes the integration of the various solutions into a coherent platform in practice.

4 Testing and Results – displays the achieved results and analyzes a selection of data collected from the real-world application of the system.

5 Future Work and Improvements – identifies areas where further work is required to improve the system and expand its capabilties.

6 Summary and Conclusion – summarizes the findings and provides conclusions from the project and this thesis.

Platform Integration for Autonomous Self-Driving Vehicles

1 Introduction and Motivation

1.1 Background

1.1.1 General

This master's thesis was performed under DTU Automation and Control in cooperation with DTU Roadrunners and DTU Department of Mechanical Engineering on the DTU Dynamo ecocar with the goal of preparing the car for the Autonomous UrbanConcept category of the Shell Eco-Marathon 2018.

1.1.2 Shell Eco-Marathon

The Shell Eco-Marathon is an annual competition about fuel efficiency held at three separate events in Asia, America and Europe [1]. In 2017, Shell Eco-Marathon Europe took place May 25-28 at the Olympic Park in London with over 200 teams from European schools and universities participating. The goal of the competition is to drive a certain number of laps on the race track while consuming as little energy as possible. Teams can choose to compete in three different energy classes: electric, hydrogen and internal combustion (including Gasoline, Diesel, CNG, and Ethanol). Two car classes are competing, the UrbanConcept class and the Prototype class. The UrbanConcept class should resemble a small city car with four wheels, where the driver needs to sit upright in an adequately sized cabin. On the other hand, the Prototype class does not have as many requirements, often resulting in a smaller three-wheeled vehicle which barely fits a driver laying down. Figure 1 shows all the participants at the event in 2016 with cars from the different classes represented in the front.

Figure 1: Group photo at the Shell Eco-Marathon 2016

Figure 2: DTU Dynamo 13.0 at the Shell Eco-Marathon 2017

1.1.3 DTU Roadrunners

DTU Roadrunners (Figure 3) is a student-driven team at the Technical University of Denmark, whose goal it is to construct fuel-efficient ecocars and participate in the Shell Eco-Marathon each year. The team consists mostly of engineering students from the mechanical and electrical departments. Since 2005, the team has participated with DTU Dynamo in the UrbanConcept class and has won its category 10 times. The current version of the car can be seen in Figure 2. It has a custom carbon fiber monocoque and runs on a specially designed ethanol powered combustion engine.

Figure 3: The DTU Roadrunners team after receiving the prize for runner-up in Internal Combustion UrbanConcept at the Shell Eco-Marathon 2017

1.1.4 The Autonomous UrbanConcept

With fully autonomous vehicles expected to be generally available from major manufacturers before 2020 [2], it is a field that is currently in rapid expansion. For the organizers of the Shell Eco-Marathon, the event is an opportunity to showcase the innovation that is possible in today's personal mobility. Starting in 2018, they plan to introduce a new category to the competition: the autonomous UrbanConcept. Even though exact details of this category are still unknown, it is expected that all technologies will be allowed and the main goal will be to manage to drive one or more laps on the track autonomously without any other cars or obstacles on the track. In subsequent competitions, more challenges will be added to showcase the vehicle's ability to handle various real-world situations.

1.2 Problem Statement

The following problem statement was established as the basis for this master's thesis:

"The aim of this project is to design, implement and test sensors and actuators to augment the DTU Ecocar with autonomous capabilities. The thesis will focus on the control aspects and prepare the necessary platform for the integration of a driverless car. The data collected will facilitate the participation in the Shell Eco-Marathon 2018 competition for self-driving vehicles."

Original problem statement

As stated, the focus of this thesis is the implementation of the self-steering feature as well as the various sensors for localization. Data from these sensors will be collected for later processing and analysis. The systems and functionalities constituting this platform are to be integrated in practice on the DTU Dynamo vehicle. Planning, navigation and advanced obstacle detection have not been pursued and are not within the scopes of this project.

1.3 Budget

A budget estimate was established at the beginning of the project. Table 1 shows the comparison between the original budget and the actual amount spent. Based on the original estimate, a budget of 75 000 DKK from the Department of Mechanical Engineering was approved, while DTU Automation and Control agreed to provide the gyroscope and a Teensy 3.6. The GPS system was sponsored by Swift Navigation, and the laser scanner was sponsored by Lightware Optoelectronics. This resulted in

a total amount of 13243 DKK spent on the project. Because of the success to obtain these sponsorships and the decision to use a lower cost laser scanner, the final cost was significantly below the budget.

Item	Link to product	Budget [DKK]	Actual [DKK]	Notes
PC	Link	2800	2792	
SSD	Link	700	639	
RAM	Link	450	437	
Teensy	Link	270	270	Provided by AUT
GSM Modem	Link	415	332	
GPS	Link	13800	962	Sponsored at a value of 13300 DKK
Gyroscope	Link	1400	1400	Provided by AUT
Wheel Encoder	Link	500	172	
Brake motor	Link	750	676	
Brake controller	Link	750	649	
Brake encoder	Link	150	179	
Steering motor	Link	750	676	
Steering controller	Link	750	649	
Steering potentiometer	Link	150	34	
Steering encoder	Link	150	179	
Extra components		3752	3197	*
Laser scanner (LIDAR)	Link	50000	0	Sponsored at a value of 7057 DKK
Total		**77537**	**13243**	

All prices are in DKK exclusive of VAT inclusive of shipping

*Extra components include: Battery, DC-DC adapters, SIM card and subscription, Website hosting, 2x LIDAR Lite v3, PCB production and PCB components

Table 1: Comparison of original budget (item designations amended for comparison) and actual cost (not including the value of sponsored items)

2 Analysis and Design

2.1 General

The main features of an autonomous vehicle can be divided into seven categories [3]:

- Self-steering
  - Ability to set the front wheels in any direction within the available degrees of freedom
- Braking
  - Ability to apply the brakes in a controlled manner
- Acceleration
  - Ability to control the throttle within the engine parameters
- Perception (obstacle detection)
  - Ability to detect any obstacles in the predicted path
  - (Advanced) Ability to track moving obstacles and predict their position
- Localization
  - Ability to use sensors to accurately determine the pose and velocity
- Navigation
  - Ability to use data from the sensors to navigate in the environment
- Planning
  - Ability to plot the required path to move from its position to a specified location and avoid obstacles

The fact that the vehicle, in this case, is specifically designed to compete in the Shell Eco-Marathon makes it possible to reduce the complexity of some of these categories. The competition takes places on a predefined race track that is closed off with barriers on both sides of the track, and no other traffic or obstacles are expected. In 2017, the total track length was 1577 m with a general track width of 7.5 m and a minimum track width of 6.2 m. The maximum speed allowed on track was 50 km/h.

For steering and braking, motors are used to respectively turn the wheel and apply the brakes. Acceleration is controlled by software with a separate engine control unit installed in the car. The CAN bus is used to communicate to this unit. Due to the optimization of the engine design towards fuel efficiency, the engine can only be operated at full throttle or be turned off.

To determine the vehicle's position, heading, and velocity relative to its starting position, sensors can be installed to measure the speed of the wheels and the heading of the vehicle (see Section 2.11 and 2.12). Furthermore, for global localization, satellite navigation systems such as GPS can be used (see Section 2.9).

Even though no obstacles are expected, basic obstacle detection is still desirable for safety concern. Several sensors exist to accomplish this task, but generally, a laser scanner (LIDAR) is used (see Section 2.10). Combined with adequate software, this sensor can also be used to detect the barriers of the race track and thus assist with localization.

Since there are currently no requirements for any image recognition, no camera based computer vision systems are used. While vision based solutions can contribute to localization and obstacle detection, for example using visual odometry or stereoscopic imaging, the high complexity of these solutions combined with the general lack of commercial availability of such solutions meant that other solutions were preferred.

2.2 Literature and existing research

Extensive research has already been done on autonomous robots and vehicles. With automated and driver assistance systems to be found in many modern cars and several manufacturers promising fully autonomous cars will arrive before 2020, many innovative solutions are currently being tested. Most notably, Tesla and Waymo (formerly Google's self-driving project) have captured the public's interest with their autonomous advances.

The most commonly used classification for autonomous vehicles is standardized by SAE International as J3016 [4]. It describes six levels of autonomy based on the amount of driver attention and intervention required, summarized in Table 2.

Level 0	No Automation	Automated system issues warnings but driver has control.
Level 1	Driver Assistance	Automated system and driver share control in some situations (for example Lane Keeping Assistance).
Level 2	Partial Automation	Automated system can take full control, but driver must be ready to immediately intervene.
Level 3	Conditional Automation	The driver can turn their attention away from the road, but must be able to take control within some delay if needed.
Level 4	High Automation	Driver attention is never required for safety. In some situations, the automated system will request the driver to take control or pause the drive to continue.
Level 5	Full Automation	Automate system has full-time control in all situations with no human intervention required.

Table 2: Levels of self-driving autonomy standardized by SAE International

Tesla has equipped their cars with what they claim to be "full self-driving capabilities" using 8 cameras, 12 ultrasonic sensors and radar that together can deliver 360-degree field of view. Their current cars are autonomous between level 2 and 3, but Tesla promises to deliver software updates to enable a higher level of autonomy in the coming years. Waymo is aiming for level 5 with their self-driving car, which has already driven over 4 million kilometers, using cameras, LIDARs and radar sensors. Other manufacturers, such as Audi, BMW, and Volvo, have also presented their self-driving vehicles, using similar sensor suites.

The scope of this project is limited to the Autonomous UrbanConcept, and it is thus of interest to analyze the autonomous vehicle "Kate" developed by the Oxford Robotics Institute under contract from Shell [5] (see Figure 4). It uses the Selenium Autonomy system, developed at Oxford University, which is a software stack focused on self-driving and machine learning. In its current state, the car would be classified under level 2 within its known environment. "Kate" has a vertically mounted 2D LIDAR (Sick NAV245) and three stereoscopic cameras for localization and visual odometry. For obstacle detection, it uses three 3D LIDARs (Velodyne VLP-16). Moreover, it has two side cameras for visualization. The main processing takes place on an onboard Mac Pro, and an iPad is used as interface. A second offboard Mac Pro is used to visualize the sensor data. No GPS is used.

The system uses the camera and LIDAR data to create a 3D map. Using machine learning techniques, it can then localize itself in this map. The heavy reliance on cameras creates challenges for driving in environments with changing lighting conditions, however, it enables a very flexible solution. The platform can, in theory, be ported to any vehicle. After "learning" the map, any vehicle equipped with this platform would be able to drive autonomously within this map. According to tests on the Oxford University campus, up to 32 manually driven laps were required until the entire track had been sufficiently "learned". For the track in London, around 10 "learning laps" were needed.

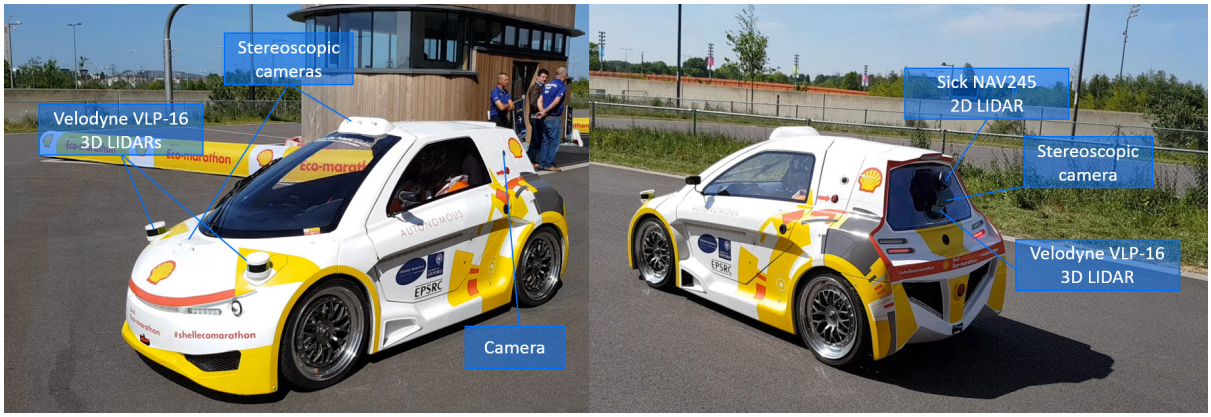


Figure 4: The Autonomous UrbanConcept prototype "Kate" developed at Oxford

2.3 DTU Dynamo

DTU Dynamo is the car that is to be expanded with autonomous capabilities (Figure 5). The car is 3.387 m long, and 1.298 m wide at its widest point, with 0.800 m between the rear wheels, 1.084m between the front wheels, and 1.516 m between the front and the rear wheels. The wheel diameter including tires is 0.559 m (22 inches).

Apart from the engine control unit, a National Instrument CompactRIO, all electronics boards are custom made. There are 5 of these boards, called Motor, Mini, Steering, Front Light, and Back Light. Apart from the analog Mini board, all boards communicate through a CAN bus. The Motor board is the main hub, interfacing with the CompactRIO and controlling the gear and starter motor. The Steering board contains a 3.5" display that acts as the main interface for the driver. It has 18 buttons which the driver can actuate. The Front Light board is responsible for the driving light, turning light, horn, fan and windshield wiper. The Back Light board in the back of the vehicle controls braking light and lights that indicate whether the starter motor is running. One 12 V battery with a capacity of 14 Ah supplies power to all electrical systems and is fused. There are two emergency shutdown switches, one external and one internal.

For a more detailed overview of the electronics in the car, I refer to "Design and Implementation of Electronics in the DTU Ecocars Dynamo and Innovator" (see Appendix B).

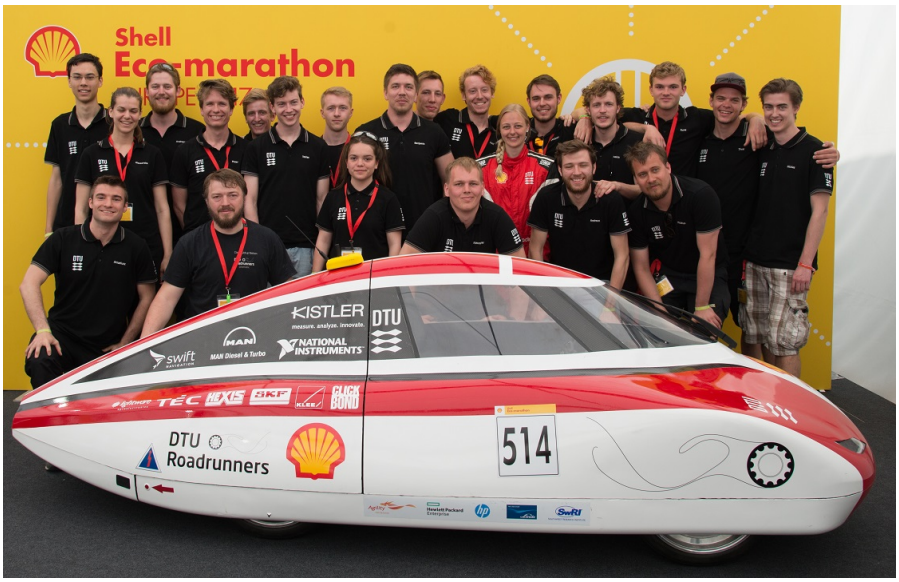


Figure 5: DTU Dynamo 13.0 and the DTU Roadrunners team at Shell Eco-Marathon 2017

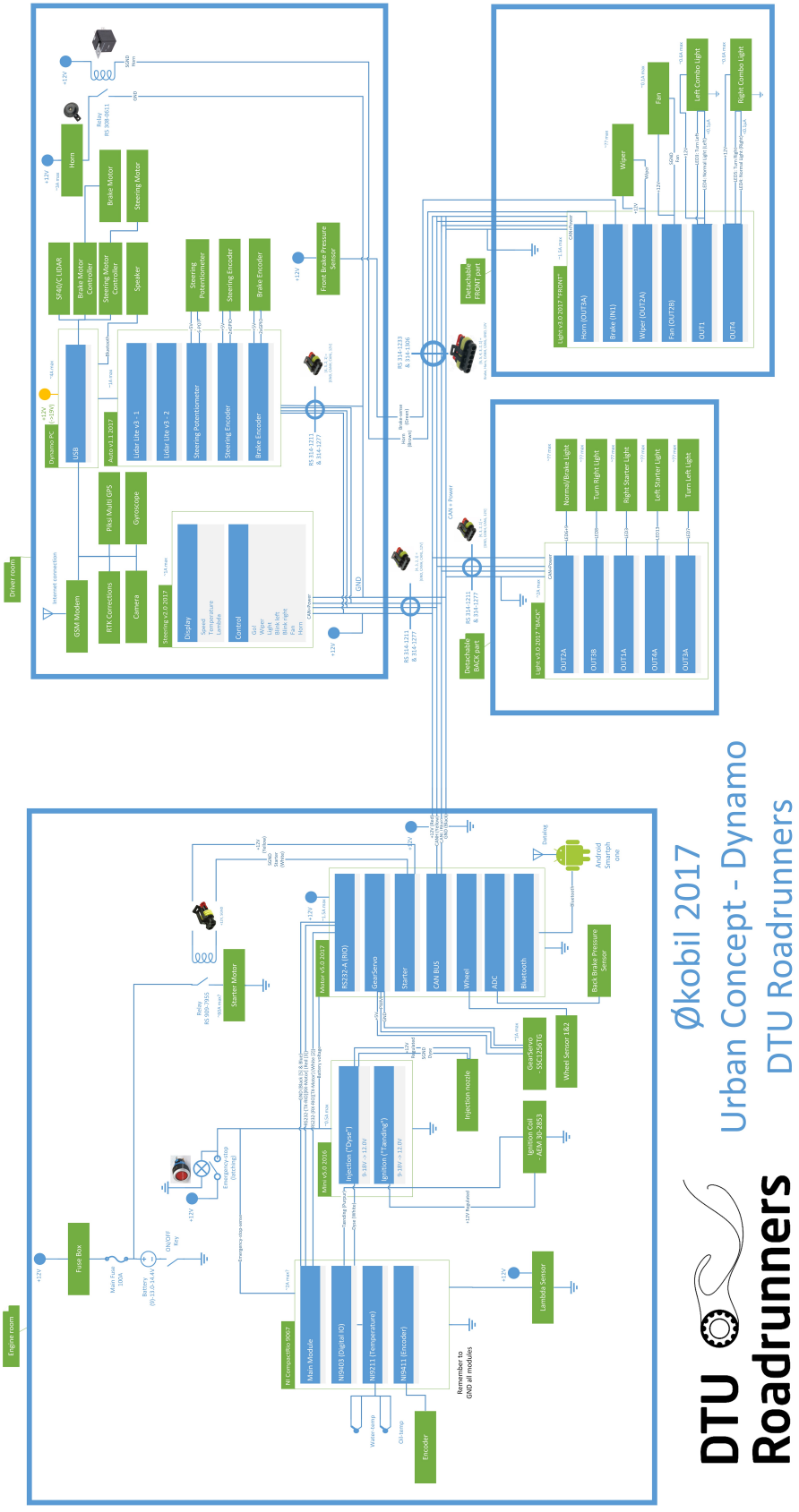


Figure 6: Overview of the electronics and electrical systems in DTU Dynamo 13.0 (high resolution version can be found at dtucar.com/diagram2017.pdf and in the digital version of this document)

2.4 Computer

The computer is where the main processing takes place. Four specifications are of interest, and for the selection, the following prioritization is chosen:

Physical size < Power consumption < Computational power < Price

Figure 7: BRIX GB-BKi5A-7200 (Intel Core i5-7200U (2 x 2.5 GHz), Intel HD Graphics 620, M.2 SSD, 2x DDR4, 802.11ac, Bluetooth 4.2)

During operation, the computer is always turned on and thus always consumes power. Since no advanced vision processing is desired, the requirements for the computational processing power are quite low. Due to this, no dedicated graphics card is required. Notebook processors of the current generation provide sufficient processing power for all the expected tasks while maintaining low power consumption. The desired form factor is based on Intel's Next Unit of Computing, also known as "small-form-factor" and several iterations available. Narrowing in on units using the newest, 7th generation Intel processors, and checking for availability, the choice came down to two variations of the Gigabyte Brix, differing only in the CPU: Intel Core i5-7200U or Intel Core i7-7500U. The difference in performance between the two lies at around 10 %, while the version with the i7 costs 1000 DKK more. With this in mind, the Gigabyte Brix GB-BKi5A-7200 was chosen (see Figure 7). According to online benchmarks, it should consume 13.3 W in idle, 43.8 W at peak and it requires a 65 W power supply.

The chosen computer is delivered as a barebone. This means that storage and RAM need to be purchased separately. The biggest expected dataset to be processed is the laser scan. For this, 8 GB of RAM should be more than sufficient. Should the requirements change, this is also an easily upgradable part. The computer uses DDR4 RAM, where one of the advantages is reduced power consumption compared to older versions.

For the storage unit, the computer has an M.2 2280 slot. An SSD with high performance and low power consumption is desired. The first candidate for these criteria was the Samsung 960 EVO, but since it was not in stock anywhere, the second best option was selected, the Samsung 850 EVO. It is 25 % slower than the 960 EVO but also uses 20 % less power while being 19 % less expensive. The version with the lowest amount of storage, 250 GB, was deemed sufficient for this application. Should more storage space be needed, for example for storing log files, an external storage device can easily be connected.

2.5 Steering Motor

The steering motor needs to fulfill several criteria, and its specifications can be divided into six categories:

Torque, Speed, Power, Angular resolution, Ease of use, Price

In order to meet the ease of use requirement, the motor should be powered by 12 V and easily controllable. Experimentally, it was determined that turning the steering wheel with the vehicle immobile on asphalt required around 3 Nm of torque.

$$\omega_{expected} = 3 \text{ Nm}$$

The datasheet for motors often specify a maximum holding torque, but often provide insufficient data concerning the motor's abilities depending on load and speed. Additionally, many factors can influence both the motors performance and the torque required to turn the wheels. For these reasons, it was decided to use a safety factor of 5, resulting in:

$$\tau_{design} = 15 \text{ Nm}$$

It is desired to be able to turn the wheels from full left to full right in less than one second. The motor is attached to the steering rod through a gear with a radius of 26 mm. Given the full travel distance of the rod of 71 mm, this results in the following desired rotational speed of the motor[1]:

$$\omega_{expected} = \frac{s_{rod}}{1\text{s} * 2\pi r_{gear}} = \frac{0.071\text{m}}{1\text{s} * 2\pi * 0.026\text{m}} = 0.435 \text{ RPS} \Rightarrow 26 \text{ RPM}$$

This number assumes infinite acceleration, which of course is unrealistic. The datasheets of motors again provide insufficient data to calculate the maximum acceleration and exact behavior of the motors under varying loads. Thus, again, a safety factor of 5 is used to give the desired maximum rotational speed of the motor:

$$\omega_{design} = 130 \text{ RPM}$$

Out of a selection of DC motors, servos, and stepper motors, only one fulfilled all criteria: the Phidget 3333_0 Bipolar Stepper with 15:1 Gearbox (see Figure 8). Being a bipolar stepper motor, it requires precisely timed control of the poles, which in return allows exact position control. For ease of use, the 1067 PhidgetStepper Bipolar controller (see Figure 9) is used. The chosen motor with gearbox has a step angle of 0.12°, a maximum speed of 116 RPM and a maximum output torque of 14.7 Nm. Its speed is slightly below the design speed, but a safety factor of 4.5 is deemed to still be sufficient. Compared to other contenders, the motor has the benefit of its low weight at 1.3 kg and its low maximum power consumption at 34 W. It has an exposed rear shaft, where an encoder can be mounted to accurately track the motor's position relative to its starting point. The HKT22 optical rotary encoder (see Figure 10) is designed to be mounted on the chosen stepper motor and provides 300 counts per revolution with full quadrature cycles.

Figure 8: Phidget 3333_0 NEMA-23 Bipolar Stepper with 15:1 Gearbox

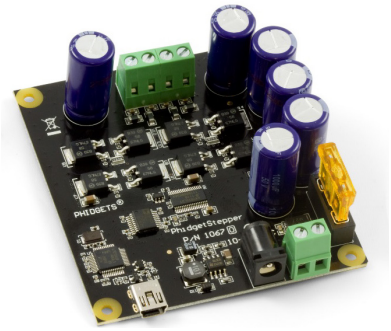


Figure 9: Phidget 1067_0 PhidgetStepper Bipolar HC controller

Figure 10: HKT22 Optical Rotary Encoder for the stepper motor

[1] RPS is rotations per second, RPM is rotations per minute

2.6 Steering Feedback

A sensor needs to provide absolute position feedback to support accurate positioning of the front wheels. This can be done multiple ways, for example with an absolute rotary encoder or a potentiometer. What is important for this sensor, is that the absolute position of the wheels is measured after the gearing of the steering rod since the gear can exercise backlash. A slide potentiometer was chosen (see Figure 13) because it can attach directly to the steering rod. Due to the fact that the resistance in a potentiometer can vary from unit to unit, as well as by its temperature and age, it is used as an adjustable voltage divider. Since it can be assumed that the change in resistance happens uniformly over the entire potentiometer range, the chosen design reduces the impact of this drawback of using potentiometers.

Figure 13: Bourns PTB 10k 100mm potentiometer

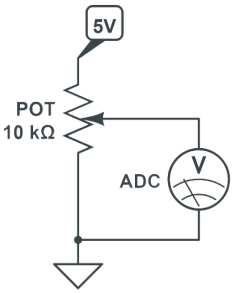


Figure 13: Potentiometer as an adjustable voltage divider

2.7 Brake Motor

DTU Dynamo was scheduled to be fitted with an all new brake system by a group in the DTU Roadrunners team for the race in 2017. This meant that the motor for the brake actuation could not be procured before the new system was in place. Because the implementation of the new braking system was delayed several weeks, there was not much time for analyzing the requirements of the brake motor. A test was conducted where the same motor as the one used for steering successfully actuated the braking system. For the sake of simplicity, it was thus decided to use the same motor. Its torque and speed ratings enable actuation of the brakes at an acceptable rate, but for fast braking at maximal power, a motor with a higher torque rating will have to be acquired.

Figure 13: Danfoss AKS 32 Pressure Transmitter

2.8 Brake feedback

As feedback for the brake control, hydraulic pressure sensors are installed in the two independent hydraulic circuits. They can provide absolute feedback about the performance of the brakes. The Danfoss AKS 32 pressure transmitter (Figure 13) was provided by the Department of Mechanical Engineering. It can operate up to 55 bar, is supplied with 12 V and the pressure is returned as an analog voltage with a response time of 4 ms. The hydraulic system of the brakes is designed so that the pressure never exceeds 50 bar.

2.9 GPS

GNSS (Global Navigation Satellite System) is widely used for global positioning in vehicles, ships, and planes. The system is based on receiving broadcasts from at least four satellites containing data about their orbital location (ephemeris) and the precise clock. With the known speed of propagation (the speed of light), the position can be triangulated. The most widely used variant, GPS, can usually deliver accuracies in the range of 1 to 5 m, depending on the number of satellite signals that can be received and the quality of these signals. At these accuracies, combined with precision in the same range of 1 to 5 m, GPS would only be usable to obtain a rough global position estimate.

The main sources of this error in position are the transmission delays due to ionospheric and tropospheric interference, as well as errors in the satellites' position data and internal clocks. Both the accuracy and precision can be dramatically improved by using a base station with a fixed position to send correctional data to the moving unit (often called the rover). A simple implementation of this concept is differential GPS (DGPS), which broadcasts the difference between the known fixed position and the position obtained by the satellite system. DPGS can improve the accuracy to 0.5 to 1 m, which is still insufficient for localization.

A more advanced technique is using Real Time Kinematics (RTK). The GPS implementation, called Carrier-Phase Enhancement GPS, measures the phase of the GPS signal's carrier wave. A fixed base station with known position broadcasts real-time corrections based on that measurement. The base station needs to be in the vicinity of the rover, specifically within a distance of about 10 km, so that it can see the same satellites as the rover and measure the same atmospheric disturbances. Using this technique, accuracies within a few centimeters can be obtained.

In addition to this, the accuracy can be further improved by using a multiband receiver which can receive signals from the L2 band in addition to the traditional, most commonly used L1 band. This adds redundancy to the measurements and can help compensate for signal interference and ionospheric error, resulting in sub-centimeter accuracy. Furthermore, the robustness can be improved by tracking additional GNSS constellations such as GLONASS and Galileo simultaneously in addition to GPS.

There exist public networks of base stations that broadcast their correction data over the internet, notably the Continuously Operating Reference Station (CORS) network focused on the United States and the EUREF Permanent GNSS Network focused on Europe. The base stations are however coarsely distributed, and not all stations transmit real-time data. For example, the nearest public station to the London race track is located at a distance of 50 km. This means a base station would have to be erected to ensure relevant and compatible correction data.

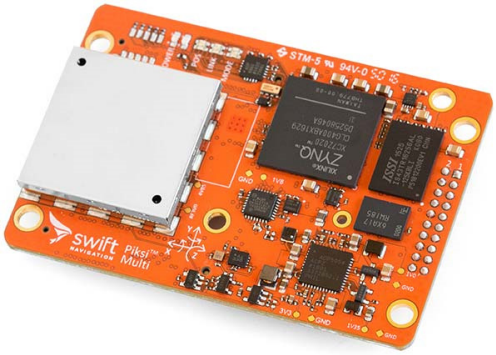


Figure 14: The SwiftNav Piksi Multi RTK GNSS module

Three products were considered for the GPS solution: Emlid Reach RTK, U-Blox C94-M8P RTK, and SwiftNav Piksi Multi (see Figure 14). The Piksi Multi has the best accuracy and convergence time of the three, being the only one capable of multi-band as well as multi-constellation operation. The convergence time determines the time between initialization until the receiver can provide RTK based location. According to its specifications, it is accurate to 2 cm horizontally and 6 cm vertically on short baselines with a good sky view. Its accuracy degrades at a rate of 1 mm horizontal and 3 mm vertical for each kilometer between the base station and the rover. The time until convergence of the RTK fix is less than a minute, and the update rate is 10 Hz. After communicating with Swift Navigation, the company graciously agreed to sponsor the Piksi Multi evaluation kit, including two modules, radios, and high-gain antennas. The standard implementation uses 2.4 GHz radios to transmit the RTK correction data. Since these radios have a limited range, communication over the internet using the 3G or 4G GSM networks should be implemented.

2.10 LIDAR

A laser scanner or scanning LIDAR (Light Detection and Ranging) is an active exteroceptive sensor that scans the environment using a laser and calculates the range based on the time needed for the light

to reach the target and return. The LIDAR can be used for localization, but its main purpose here is for obstacle detection. One of the challenges when using LIDARs for localization or mapping is to ensure accurate feature detection. Depending on the mounting point of the sensor, bumps in the road or changes in elevation will cause the measurements to become unusable, and algorithms need to be able to handle those situations.

Scanning LIDARs can be characterized by the following specifications:

Maximal range, Angular resolution, Scanning speed,
Horizontal field of view, Vertical field of view, Accuracy

The most common LIDARs scan in a single 2D horizontal plane, whereas more advanced ones scan several horizontal planes, essentially providing a 3D image. The 3D LIDAR provides a far more detailed scan, which can aid in localization and mapping of the track as well as compensate for bumps in the road or changes in elevation.

The maximum speed that DTU Dynamo can reach is 50 km/h. Braking from this speed at maximum braking power should yield a braking distance of no more than 20 m according to the design of the brakes. Adding a safety margin to this, we can specify that the desired sensor should be capable of measuring at a range of at least 50 m. The sensor should capture at least an 180° window in front of the car to include all the desired information. Power consumption and weight should be as low as possible. Although a vertical field of view can be useful, it is not necessary. The rotation rate, angular resolution, and accuracy simply need to be good enough for obstacle detection. Table 3 shows a comparison of the specifications of three selected options.

Initially, the Velodyne VLP-16 was chosen, since its vertical field of view, fast rotation rate and good resolution meant it could later also be used for detailed localization and mapping. However, it was impossible to procure the device, with expected delivery in September 2017. As a replacement, focused on obstacle detection, the Lightware SF40/C (Figure 15) was chosen, and Lightware Optoelectronics agreed to offer the sensor in a sponsorship towards the project.

	Velodyne VLP-16	**Sick LMS511**	**Lightware SF40/C**
Price	$7999	$7000	$999
Range	100 m	80 m	100 m
Horizontal Field of View	360°	190°	360°
Vertical Field of View	± 15°	N/A	N/A
Power consumption	8 W	22 W	4.5 W
Weight	830 g	3700 g	270 g
Rotation Rate	5 – 20 Hz	25 – 100 Hz	1 – 4.5 Hz
Horizontal Angular Resolution	0.1 – 0.4°	0.167 – 1°	0.18°
Accuracy	3 cm	1 – 3 cm	3 – 12 cm

Table 3: Comparison of LIDARs considered

Depending on the placement of the scanning LIDAR, it can be difficult to detect the distance to the sides of the vehicle. Knowing this distance can be useful to localize the vehicle with respect to the side barriers of the track. Additionally, it can be used to detect obstacles to the sides of the vehicle, which can become relevant when turning. For this purpose, two Garmin LIDAR Lite v3 sensors (see Figure 16) have been chosen. These side LIDARs are not rotating and always measure the same area to the

left and the right of the car, respectively. They are small, have low power consumption and can measure with an accuracy of 2.5 cm up to 40 m distance at a rate of 500 Hz. The beam divergence can be expressed by the beam diameter:

$$D_{beam} = \frac{distance}{100}$$

With the general track width of 7.5 m and the car positioned in the center of the track, the expected distance to the barriers is 3.1 m. At this distance, the beam diameter is 31 cm.

Figure 15: Lightware Optoelectronics SF40/C laser scanner

Figure 16: Garmin LIDAR Lite v3 optical distance measurement sensor

2.11 Inertial Measurement

An Inertial Measurement Unit (IMU) often combines an accelerometer, a gyroscope, and a magnetometer to provide data about the vehicle's acceleration, angular velocity and orientation. For this use case, the most important data desired is the heading of the vehicle. The relative change in heading can be provided by a single axis gyroscope, while a magnetometer can provide a point of initialization. Magnetometers measure the Earth's magnetic field, like a compass, and are notoriously sensitive to external disturbances. Furthermore, the gyroscope's measurements are also not perfect and can drift over time. Both drawbacks can be compensated through regular updates from the accurate GPS sensor, which can provide accurate heading data while the vehicle is in motion.

Figure 17: The Cruizcore XG1010 single axis gyroscope

The Cruizcore XG1010 single axis gyroscope (Figure 17) was provided by DTU Automation and Control. It has a bias drift of 10 degrees per hour and can provide accurate information about the orientation with 0.1 degrees per second of rate noise at 50 Hz.

2.12 Wheel odometry

To measure the speed and the distance of the vehicle, the simplest solution is to install a sensor on one more of the wheels. The measurement accuracy will depend on the accuracy of the measured wheel diameter, which can vary based on tire pressure and vehicle load. Additional errors can occur if the wheel slips or loses contact to the ground. Many rotary encoders exist that could be used in this application. Optical encoders were excluded since they can be sensitive to dirt and dust, which is present around the wheels. Magnetic and inductive encoders are more suitable for these conditions. The associated disks come in different shapes, some with gear-like forms, others with holes. To be able to measure direction and at the same time increase resolution, two inductive sensors were chosen. A resolution of maximally 5 cm is desired. Given the available water jet cutter at the DTU Department

of Mechanical Engineering, a steel disk with 30 holes was designed by the DTU Roadrunners group responsible for the brake system. The mount was designed so that it could fit directly on the brake disk (see Figure 18). With 30 holes and 2 sensors, 120 edges can be measured, resulting in a resolution of 1.44 cm.

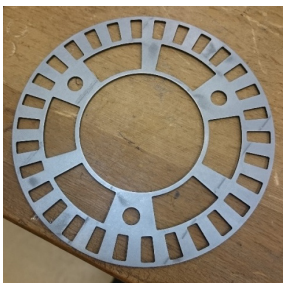


Figure 18: Rotary encoder disk for wheel odometry

2.13 Software

An autonomous system consists of many parts, as described in the previous sections. The software is tasked with binding the parts together. It needs to take data in, process it and control the actuators. These tasks can be divided into different independent blocks that communicate through public interfaces. Separating code into modules allows parallel development of functionalities and facilitates serviceability of the code. Furthermore, it allows code to execute in parallel at runtime, increasing the system's performance.

The Robot Operating System (ROS) is a free and open source framework which is built specifically to facilitate the development of modular software for robotics applications. Using a widely used framework such as ROS also makes it easier for future development of the system. New developers can focus on a single aspect of the system and are only required to have knowledge about the public interfaces between the different modules. ROS contains many functionalities, and thousands of ROS compatible packages are available online, although the main feature used by this project is its messaging system. This system allows asynchronous inter-process communication between modules, known in ROS terminology as nodes.

Figure 19 illustrates the typical setup of the ROS messaging system with two nodes. The advertiser node publishes a message to a specified topic with a predefined message type. The subscriber node registered to this topic receives the message in a callback. There can be multiple advertisers and subscribers for each topic. The communication is handled by the ROS core through a TCP/IP protocol, ensuring arrival and integrity of the message.

The most current version of ROS as of April 2017, ROS Kinetic Kame, was used and installed on the most recent Long-Term Support Ubuntu distribution, Ubuntu 16.04 LTS.

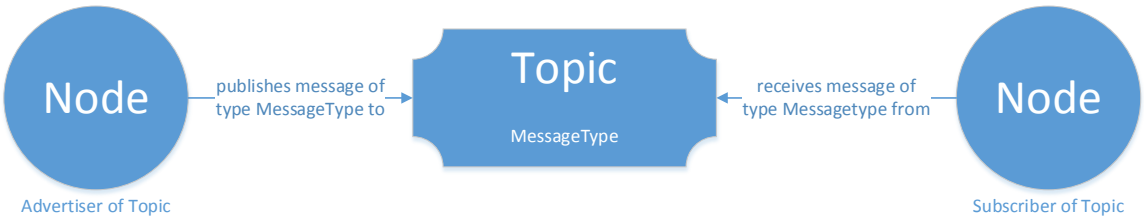


Figure 19: Simplified overview of the ROS messaging system

3 Implementation

3.1 Overview

The source files for the ROS software, Android software, embedded software, EAGLE designs, 3D prints, and website created as part of this master's thesis can be found in Appendix A.

All the devices described in Section 2 need to be installed and interfaced with the computer. Some sensors and controllers can be interfaced directly through USB while other use I2C or require an analog interface. To interface with these other devices, the Auto board was specifically designed for this application. (see Section 3.2).

A block diagram of all the interfaces between the devices can be seen in Figure 20.

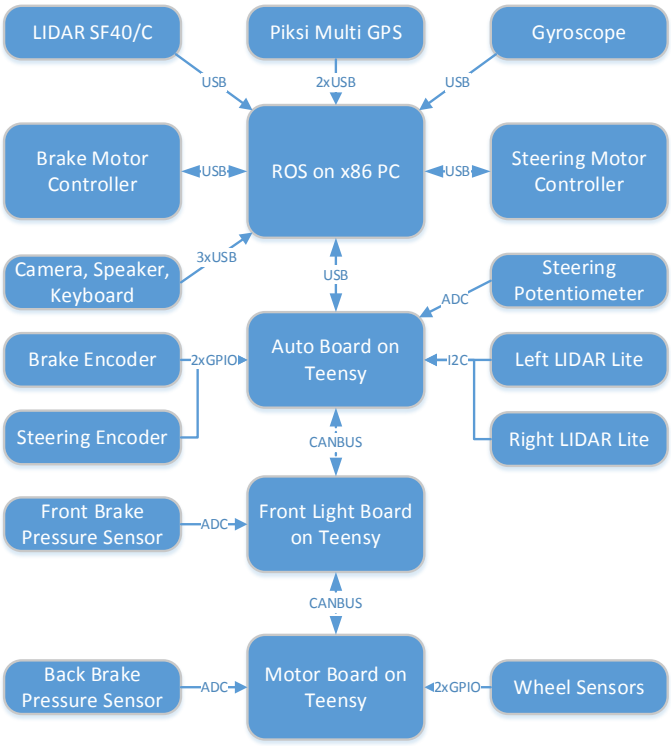


Figure 20: Block diagram of devices in the vehicle relevant for the autonomous system

3.2 The "Auto" board

The "Auto" board (Figure 21) is a custom PCB (Printed Circuit Board) designed in EAGLE to act as an interface between the computer and the devices that cannot be connected directly by USB. The main microcontroller unit (MCU) is an NXP MK66 with a 32-bit ARM Cortex-M4 core running 180 MHz at 3.3 V. It is used as part of the Teensy 3.6 development board, similarly to all other electronics boards in the car. Programmable over USB and with a high number of GPIOs, ADCs, I2C, SPI, PWM, CAN ports and FPU, this MCU has more than enough features for this application. All other electronics boards in DTU Dynamo are using the same microcontroller, thus compatibility is ensured.

The schematic and board design can be found in Appendix D and Appendix E and respectively.

The board is connected to the CAN bus of the vehicle and supplied with 9-15 V. A DC-DC converter converts this to a stable 5V. The Teensy board has its own 3.3 V voltage regulator. Since the MCU runs at 3.3 V while most devices it interfaces with run on 5 V, simple MOSFET-based level converters are

used. The two side LIDARs are connected through the I2C bus and supplied with 5 V. The encoders for the steering and braking motors also run on 5 V, and both provide a quadrature signal which is captured by the MCU on interrupt capable GPIO ports. The steering potentiometer acts as a variable voltage divider, supplied with 5 V. This signal is measured after voltage division by an ADC port. An MPU9250 IMU is also added to the board, providing accelerometer, gyroscope and magnetometer data in 9 degrees of freedom. To enable new devices to be connected, the board also has two extra ADCs, an I2C bus and an SPI bus exposed. Two status LEDs can be controlled from the MCU. All analog connections are protected against overvoltage by Zener diodes, and the CAN bus connections are protected against ESD (electrostatic discharge). Polarized Molex Micro-Fit 3.0 connectors are used, that cannot be connected incorrectly. The main voltage supply is protected against reverse polarity and fused. Noise on the power lines is expected when the starter motor runs. All power lines are equipped with large electrolytic capacitors with significant safety margins as well as ceramic capacitors which should increase robustness against noise on the power lines. Similarly, all analog inputs have a small analog low-pass filter to prevent noisy readings.

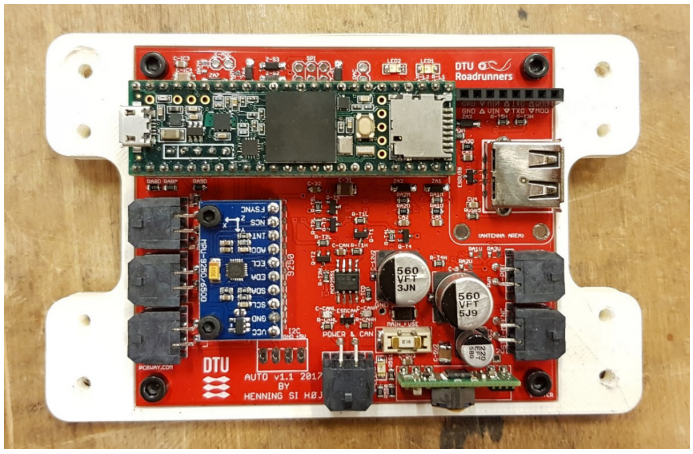


Figure 21: Fully soldered "Auto 2017 1.1" board

3.3 3D Printed Enclosures

To make it easier to install the boards and protect them, 3D printed enclosures for the Auto, Motor controller and Piksi Multi boards were designed. They were drawn in SolidWorks and printed on Ultimaker 2+ with PLA filament. Light guides on the enclosure for the Auto board (Figure 22) make the status LEDs visible from the outside. The motor controller enclosure (Figure 23) has holes on the sides and top of the lid to allow airflow to cool the power components. For the Piksi Multi enclosure, two variations were used, one for the tripod mounted base station (Figure 24), and one for the vehicle.

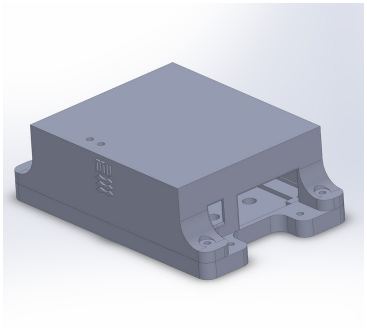


Figure 22: The enclosure for the Auto board

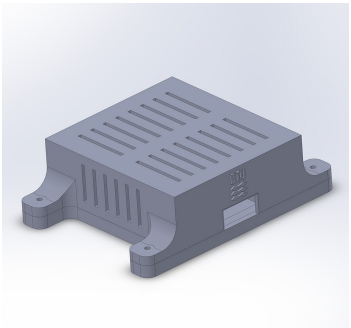


Figure 23: The enclosure for the Phidget Stepper motor controller

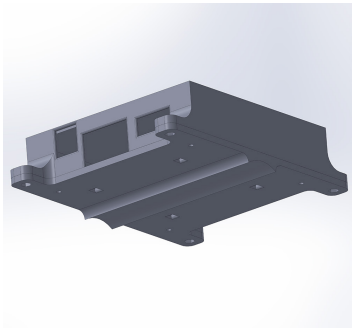


Figure 24: The base station variation of the Piksi Multi enclosure

Enclosures for the GPS "Freewave" 2.4 GHz radios were also produced since these were used for testing before the GSM synchronization was implemented.

3.4 Mounting the Hardware

Since the car was to participate in the fuel mileage challenge of the Shell Eco-Marathon, where low weight is essential, all boards and devices were mounted in a modular fashion (Figure 25). Where possible, Dual-Lock Reclosable Fasteners were used, and where not, simple nut and screw mounts were used. This also minimized interference with other work on the car. For the same reasons, a separate battery was used, even though the system is prepared to run on the same battery as the rest of the car.

As a measure of safety, it was ensured that the driver of the vehicle can override the brake and steering motors at any time. The brake motor mount (Figure 26) was designed so the driver can override it and increase brake power at any time. For the steering motor (Figure 27), a mechanical slip gear was used. If the driver applies sufficient torque to the steering column, the gear slips and the driver overrides the steering action.

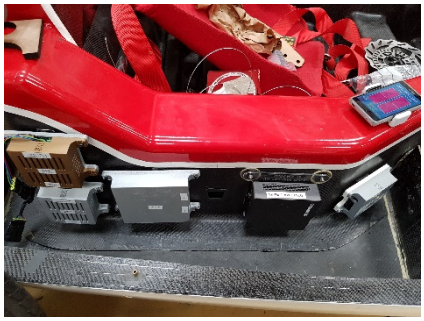


Figure 25: Modular mount for electronics and computer

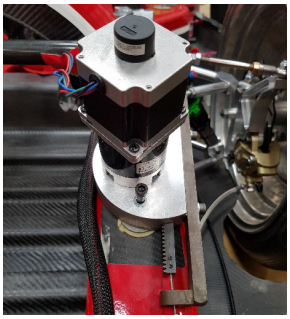


Figure 26: Stepper motor for control of the brakes

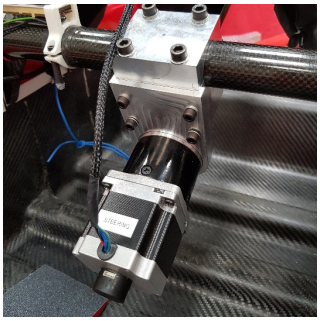


Figure 27: Stepper motor for control of the steering wheel

The GPS antenna was mounted externally to ensure it has an unobstructed view of the sky. While the side LIDARs were supposed to be integrated into the bottom chassis, they were instead mounted on metal pieces sticking out on the left and the right sides to support full modularity. The GPS and side LIDAR mounting can be seen in Figure 28.

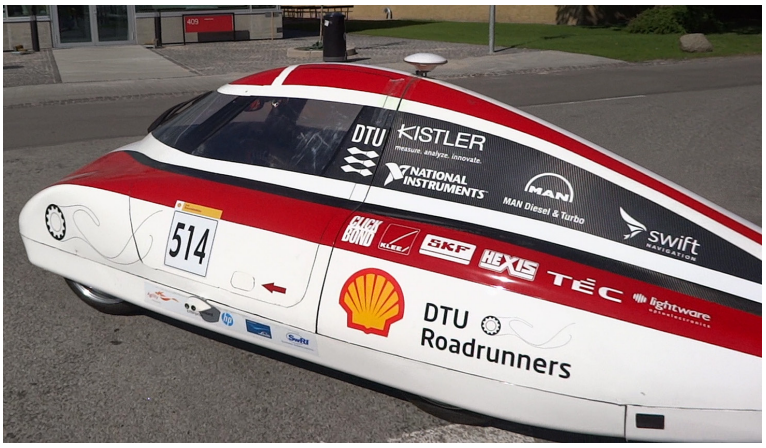


Figure 28: Side view of the car with visible GPS antenna and side mounted LIDAR

The LIDAR was mounted in front of the vehicle, where it has more than 180 degrees field of view (Figure 29). With a mount of 32.7 cm above the ground, at the worst change in inclination on the 2017 race track (just before the hill on the track), the LIDAR has an unobstructed view 20.5 m in front of the car.

The gyroscope was mounted on the rigid part of the monocoque under the seat with some rubber to reduce vertical vibrations (Figure 30).

The potentiometer was attached to the steering rod with a 3D printed mount (Figure 31).

The front pressure transmitters for the hydraulic braking system can be seen in Figure 32.

As intended, the wheel encoder disk was mounted directly to the brake disk (Figure 33).

Figure 29: Scanning LIDAR mounted at the front of our vehicle

Figure 30: Gyroscope mounting under the seat

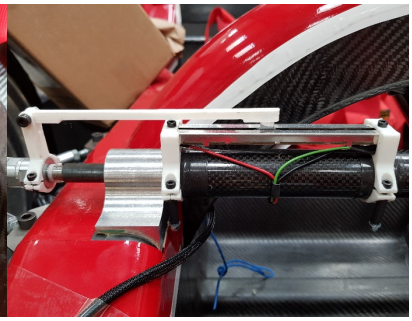


Figure 31: Potentiometer for absolute positioning of the steering wheel

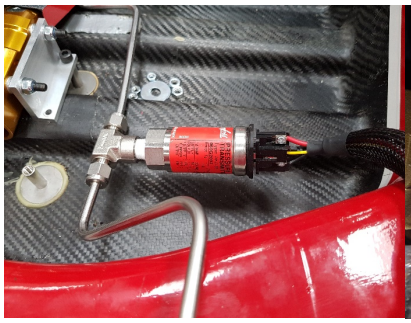


Figure 32: Front pressure sensor for the hydraulic brake circuit

Figure 33: Wheel encoder disk mounted to the right rear wheel

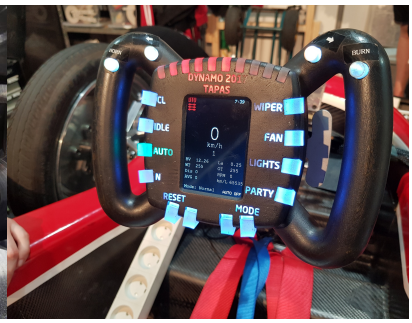


Figure 34: Steering wheel

In Figure 34, the steering wheel can be seen. In addition to its normal features, it can be used to toggle the autonomous mode on and off and indicates the state of the autonomous system.

On the left side of Figure 35, the RTK base station can be seen, mounted to a tripod with the antenna, the Piksi Multi, a battery and a phone. The sensors can also be seen in relation to one another.

Figure 35: RTK base station and DTU Dynamo without top shell

3.5 ROS Software

3.5.1 Dependencies

In addition to the ROS framework, the software has the following dependencies:

- libphidget21 (LGPL) [6]
- libserialport (LGPL) [7]
- libsbp (LGPL) [8]
- jsoncpp (MIT) [9]

All four are open-source libraries. The first three are licensed under the GNU Lesser General Public License 3.0 (LGPL). This is a permissive free software license. When used as a library through interfaces, it may be distributed under any terms and without source code. The JSON library is licensed under the MIT License with no restrictions.

Additionally, code is used in an altered form from the following two repositories:

- lightware_sf40c_ros_driver [10]
- swiftnav_piksi [11]

Both of these repositories as well as the ROS framework itself are open-source and licensed under the BSD-3 license, which permits all uses as long as the license and copyright text are preserved.

Two libraries have been written on which some of the nodes depend.

- The "Serial Tools" library contains a utility function which allows identification of the USB port name to which a device with a specified serial number is attached. When a serial connection is established, a port name in the format "dev/ttyUSB0" is needed. This port name is allocated dynamically depending on the port and the order by which the USB devices are connected. By using this library, the software can identify the name regardless of the connection port, and all USB ports can be used for all USB devices.
- The "MyQueue" library implements a simple circular buffer and is optimized to provide the average of said buffer. Values can be added to the buffer up to a predefined maximum, after which the last entry is overwritten by the next one. The sum is continuously updated so that the running average can be obtained with only a single division operation.

3.5.2 General

Figure 36 shows an overview of all nodes and topics. All nodes and associated functions are written in C++. A node advertising a topic (arrow pointing towards topic) can publish messages to the subscriber of the topic (arrow pointing away from topic). The data contained in the messages types can be seen inside the topics, and outgoing connections from the nodes can be seen under the node names.

All nodes can be launched and initialized simultaneously using launch scripts. They are written to be robust and can handle disconnections and loss of data gracefully by reconnecting to their devices or relaunching the node. Each node can be in three states:

- The "OK" state indicates problem-free execution.
- The "Wait" state indicates the node is not initialized yet or trying to recover from an error.
- The "Fatal error" state indicates the node could not recover from an error and must be re-launched.

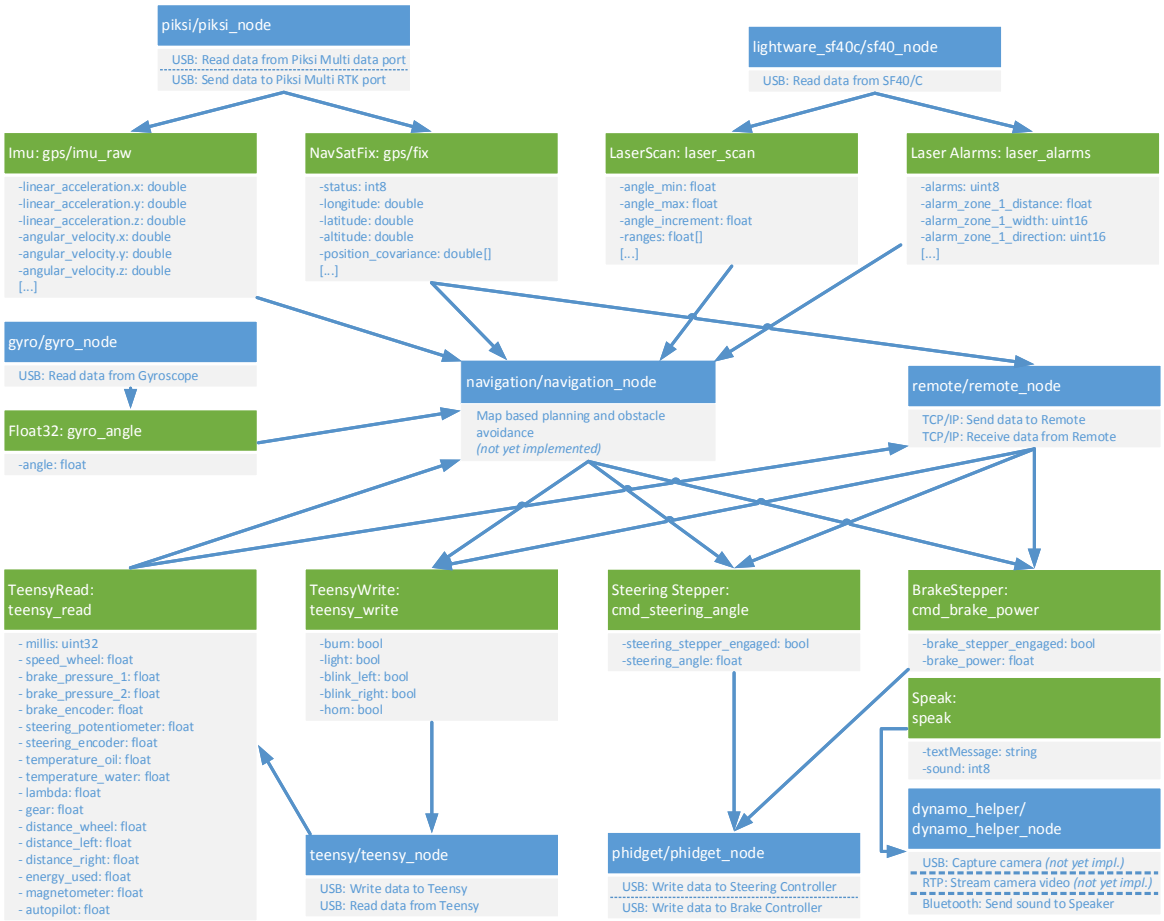


Figure 36: Overview diagram of all nodes (blue) and their communication paths through topics (green) as well as all connection going out to external devices

3.5.3 Gyro Node

The Gyro node connects to the gyroscope through its USB serial port and runs at 100 Hz. Data is received continuously, and each packet begins with two start bytes that are 0xFF. The angular rate, angle, and checksum are each two bytes in length. With the checksum, the data can be checked for errors and discarded if an error is detected. When this node is initialized, the angle is set to 0. Subsequent angle changes are published relative to the initialization angle.

3.5.4 SF40 Node

The SF40 node establishes two-way communication to the LIDAR through its USB serial port. The rotation speed of the LIDAR can be set to 1, 2.25 or 4.5 Hz, which is then used for the loop rate of the node. Laser scan and alarm values are parsed from comma separated streams and published regularly. They can be configured for the area of interest with preset angles and headings. The SF40/C LIDAR also provides the capabilities of simple navigation in the form of a heading, indicating where no obstacles were detected and sufficient clearance exists.

3.5.5 Piksi Node

The Piksi node connects to the Piksi Multi board through its USB serial port and its RS232 RTK port using a USB to RS232 adapter. The libsbp library handles the parsing of the data stream from the Swift

Binary Protocol to GPS and IMU data, which are published from callbacks as soon as the data of interest is received. Communication to the Piksi Multi base station is established through a peer-to-peer socket over the internet. The TCP protocol is used to receive data from the base station at 10 Hz and transmit the received bytes to the RTK port of the onboard Piksi Multi.

3.5.6 Teensy Node

The Teensy node connects to the Auto board through the Teensy's USB serial port. It sends and receives serial data encoded in a JSON format. While this is not the most efficient protocol, it is widely used and very flexible. Objects can easily be added or removed on either side of the transmission line without interfering with the operation. The values that are being sent and received can be seen in Figure 36.

3.5.7 Navigation Node

The Navigation node is intended to be the main processing node of the system. It receives data from all the sensors and then publishes the desired steering angle, brake power and desire to accelerate. This node should contain the preprogrammed map and use the sensor data to localize the vehicle, avoid obstacles and navigate on the track. It has not been implemented yet (see Section 5) since it is outside the scope of this thesis.

3.5.8 Phidget Node

The Phidget node establishes connection to the two Phidget Stepper motor controllers for steering and brake actuation. It subscribes to the "teensy_read" topic to receive encoder, potentiometer and pressure feedback for the control loops. The loops are implemented in two classes, responsible for the steering and brake motor respectively and run at 100 Hz.

In the current implementation, the Phidget node is also responsible for simple autonomous driving by recording data of a manual run to a file and then loading that data and driving the run autonomously based on it.

3.5.8.1 Steering motor control

The steering control class implements closed-loop control for the steering motor. It receives a reference angle and three feedback values. The motor controller reports the number of steps it has tried to impose on the motor. Since the motor can skip steps, for example, if the torque is too high, this value does not indicate the actual position of the motor. For this, the encoder on the rear shaft can be used. It reports the actual number of ticks the motor has moved. Since the motor connects to the steering rod through a slip gear, the motor can turn without the wheels turning, for example, if the driver overrides the motor. For the absolute position of the steering wheels, the potentiometer is used. Due to backlash in the steering, there can be a small inconsistency between the motor movement and the potentiometer feedback. Since adjustments in the potentiometer and wheel mount can have an effect on the calculations in the control loop, a calibration is performed on initialization, turning the wheels from full left to full right.

The Phidget motor controller allows the step target, maximum velocity, and acceleration to be set. It has a built-in control loop that tries to reach the step target at the given velocity and acceleration based on its step count. To ensure the desired wheel position is reached, a simple proportional velocity controller is wrapped around, taking feedback from the encoder. The potentiometer is used to set the step target.

3.5.8.2 Brake motor control

The brake control class is very similar to the steering control class. It receives a reference braking power between 0 and 1 and three feedback values: the number of steps, the encoder value, and the brake pressure. Pressure transmitters from the front and rear hydraulic systems are used to measure the absolute braking power. The brake motor can lose steps if the torque is too high. This can be seen in Figure 37, where the motor slips after 0.7 seconds. After this event, the encoder and step values are no longer equal. Therefore the pressure reading is used to compensate for this.

To ensure the desired brake power is reached, a simple proportional velocity controller is wrapped around the motor controller, taking feedback from the encoder. The brake pressure is used to set the step target.

The formula to convert the brake pressure ADC reading to bar is:

$$P = \left(\frac{n_{ADC}}{2^{16}-1} \cdot \frac{V_{ref}}{\frac{R_1}{R_1+R_2}} - 1 \right) \cdot \frac{P_{max}}{V_{max}} = \left(\frac{n_{ADC}}{2^{16}-1} \cdot \frac{3.3\ \mathrm{V}}{\frac{4700\ \Omega}{4700\ \Omega + 10000\ \Omega}} - 1 \right) \cdot \frac{55\ \mathrm{bar}}{5\ \mathrm{V}}$$

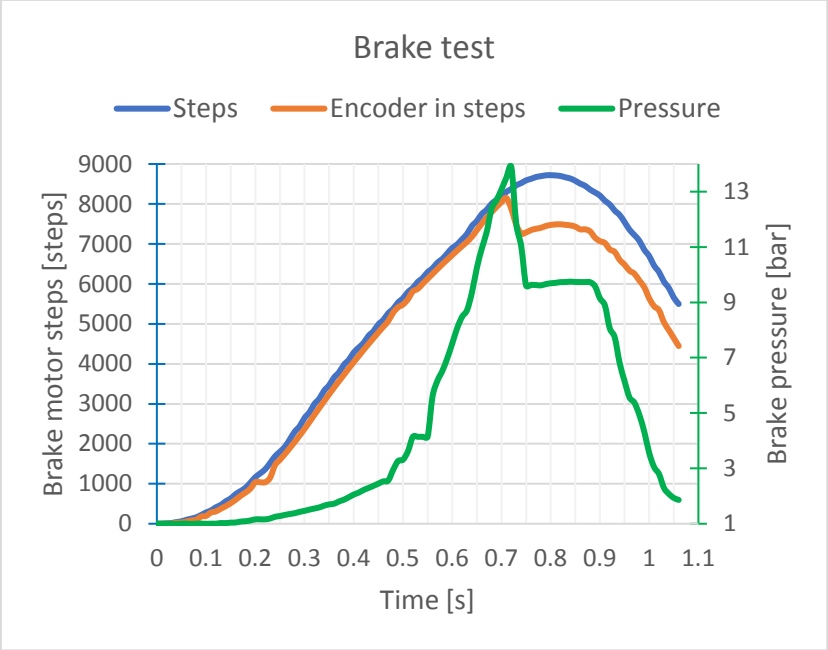


Figure 37: Brake test where maximum motor torque is exceeded

3.5.9 Dynamo Helper Node

The Dynamo Helper node is a node that is intended to interface with all other nodes. It can implement services to check if all nodes are running as intended and restart them if needed. Additionally, it can be responsible for starting the camera and data stream. Currently, this is performed manually, and the node only contains functionality for audio feedback through the speaker.

3.5.10 Remote Node

The Remote node connects to the Remote app (see Section 3.6) through a peer-to-peer socket over the internet. The TCP/IP protocol is used, because it ensures delivery and implements full handshake and data loss handling, with the downside of being slower than UDP. Since the computer acts as a

server and the remote acts as a client, the client needs to know the IP address of the computer. This IP address can change every time a new connection to the internet is established. Therefore, a script publishes the current IP address of the computer to a database, where the remote can fetch it and connect to the computer. The remote sends commands to the Remote node, which publishes them to the associated topics while sending data about the vehicle's performance to the remote (see Figure 36).

3.6 Android Software

3.6.1 Remote App

The Remote app (Figure 38) is written in Java and connects to the server set up by the Remote node on the computer. It uses two libraries:

- The "virtual-joystick-android" library [12], licensed under Apache 2.0, which is a license similar to the before mentioned LGPL.
- The "SlideView" library [13], licensed under the MIT License.

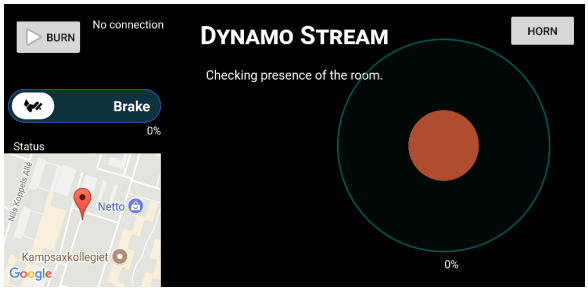


Figure 38: The Android Remote app

The joystick controls the steering wheels, and a slider controls the brake power. There are two press-and-hold buttons: one for acceleration ("burn") and one for the horn. A map indicates the current position of the car, and the speed, distance, and gear are shown in plain text. The app implements a multi-touch activity, such that all buttons can be pressed at the same time and are responsive, indicating their current state. In the background, the "Dynamo Stream" connects to the webcam of the vehicle and gives a live first-person view from the car.

3.6.2 Piksi Base Station App

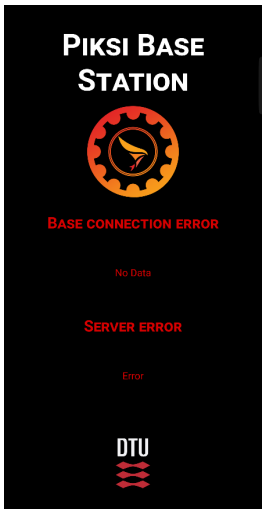


Figure 39: The Android Piksi Base Station app

The Piksi base station app (Figure 39) is written in Java and connects to the server set up by the Piksi node on the computer. It receives the RTK correction data from the Piksi Multi through an RS232 to USB adapter plugged into the smartphone's USB port. This data is then sent directly to the socket of the computer. The smartphone activity displays the status of the USB and the socket connection.

3.7 Embedded software

The embedded software on the Teensy is written in C++ with the Teensyduino framework (MIT License).

In addition to this, three open-source libraries are used:

- MPU9250 [14], MIT License
- ArduinoJson [15], MIT License
- Lidar Lite v3 [16], Apache 2.0 License

The code on the Auto board connects to the I2C sensors (IMU and Lidar Lite) and the CAN bus and sends the relevant data to the computer in a JSON string over USB. It then receives a JSON string from the computer with the commands to publish on the CAN bus.

In addition to the Auto board, embedded software has been written for the Front Light board (to measure the brake pressure), the Steering board (to toggle autonomy and display status) and the Motor board (to measure the brake pressure and the wheel sensors).

3.8 Visualization software

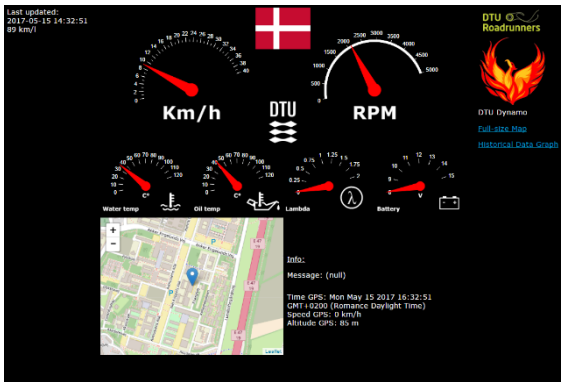


Figure 40: Website with visualization about the car's performance

Visualization of the car's performance also plays an important part and can help capture the attention of visitors and judges at the competition. The "DTUcar" website displays live GPS location, speed, engine data and much more. It uses PHP to upload data to a MySQL database which is visualized with Javascript on an HTML page. Much of the data about the autonomous functions can be added to this website in the future, such as the live webcam feed. It uses a Real-Time Protocol which results in an average delay of 0.3 s for the stream under normal signal conditions.

4 Testing and Results

4.1 General

A week before the competition, the car was able to capture data from all its sensors and drive a predefined path by steering autonomously. Within the predefines environment, the car can be said to be close to level 2 autonomy. While it had the ability to accelerate, this was not activated for safety reasons, and the tests were performed by pushing the car and letting it steer itself.

During the training rounds on the track in London, the autonomous system was installed and captured data from the sensors while the driver steered the vehicle. In the following, some selected data is analyzed and presented. Analyzing and processing the comprehensive set of data collected before and during the competition could be a project on its own and is beyond the scope of this thesis.

As part of the Shell Eco-Marathon 2017, the car also competed for the "Technical Innovation Award" (see application in Appendix C) in which it was mentioned as the runner-up for the award.

4.2 Steering

The steering motor worked as intended and allowed the car to steer itself. Several tests were performed with predefined steering programs. The steering was also tested with the remote-control app. The position of the steering potentiometer was recorded throughout the race attempts in London.

Videos from some of these tests can be seen here: https://dtucar.com/auto.html

4.3 Braking

The brake motor had sufficient torque to perform a deceleration from low speed within reasonable time, being able to reach 10 bar in hydraulic pressure. The speed and pressure curves from a mild deceleration can be seen in Figure 41. Deceleration from higher speeds was not tested.

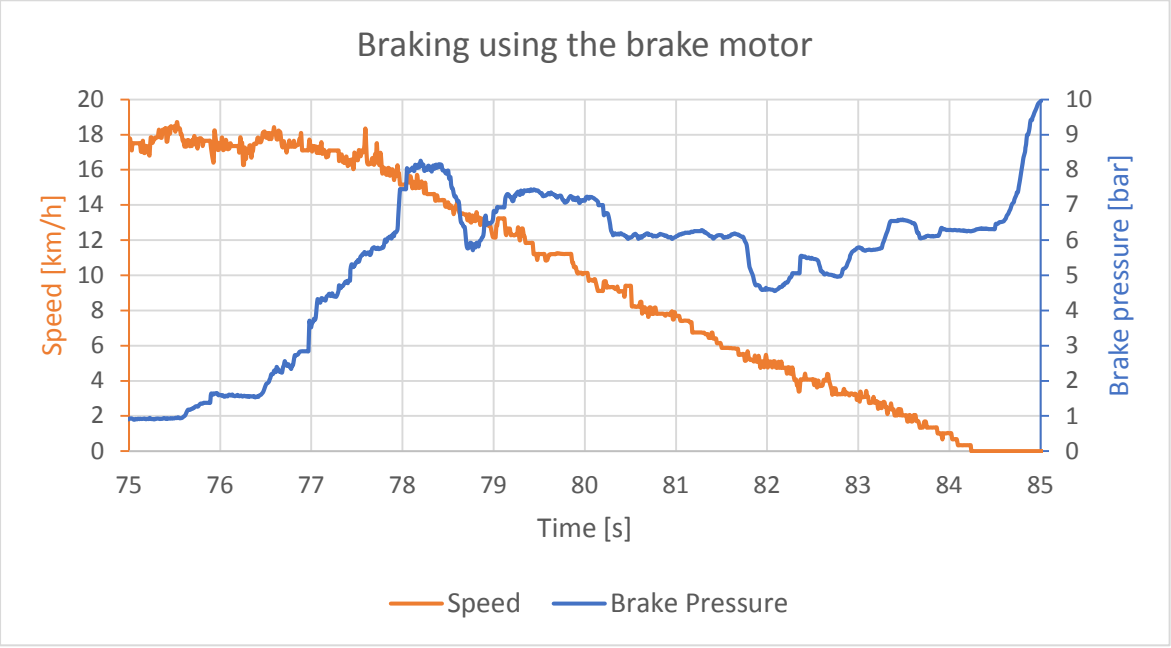


Figure 41: Brake and speed curves from a mild deceleration actuated by the brake motor

4.4 LIDAR

The SF40/C provided consistent laser scans at its maximum rate of 4.5 Hz. One such sweep can be seen in Figure 42, captured while the car was waiting to go on the track. The figure also shows two of the alarm zones. The LIDAR can report up to 7 alarm zones predefined by distance, angular width, and direction. Currently, two alarm zones are enabled: one at 2 m with 90 degrees width and a second one at 5 m with 20 degrees width. Both are pointing forward (0 degrees). Unfortunately, a broken connector meant that the LIDAR data was incomplete and only collected at the beginning of the run.

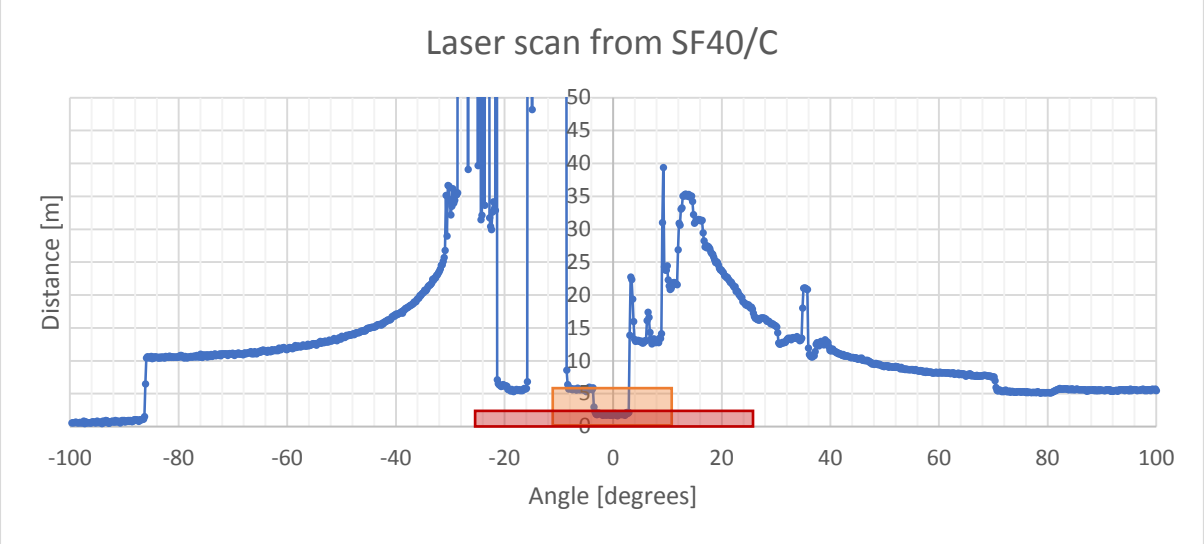


Figure 42: Single laser scan from the LIDAR mount at the front of the vehicle

The side LIDARs provided data about the distance to the sides, as can be seen in Figure 43. For the first 92 seconds, the car is in the narrow start lane, which is 3.2 meters wide according to the organizers. Adding the left and right distances to the width of the car and the mounting plate, the track width at the starting line is measured to be 3.3 meters. Shortly thereafter, the car enters the main track, where the track width is measured to be 7.5 meters, matching the specified general track width.

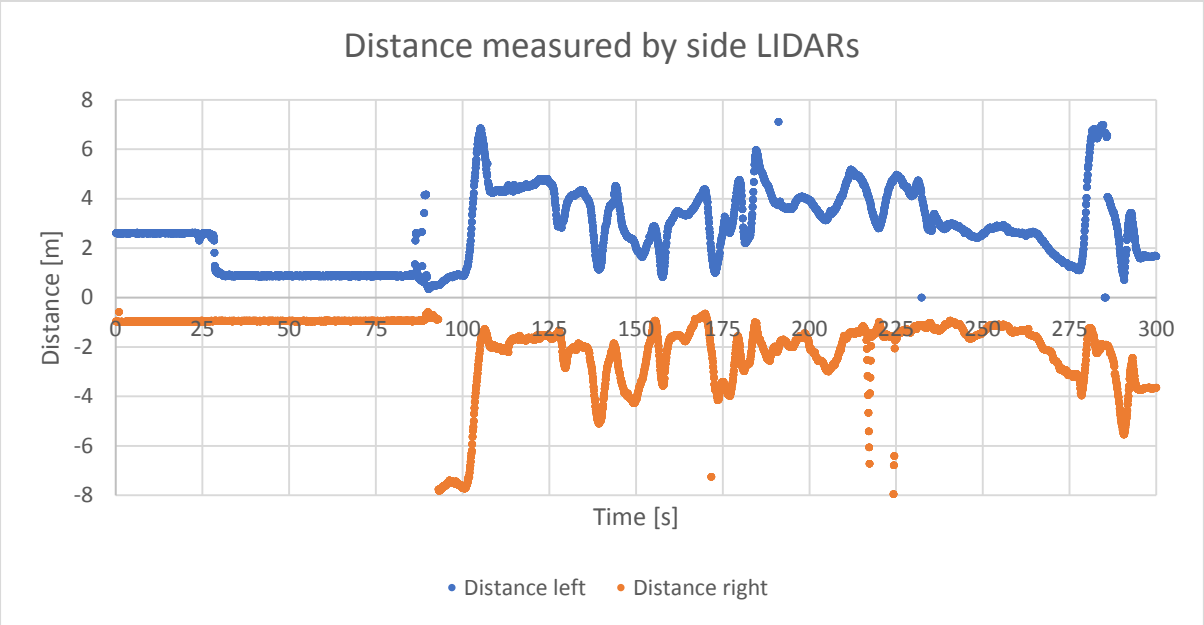


Figure 43: Distance measured by the side LIDARs at the beginning of a race

4.5 GPS

The GPS sensor lived up to its specifications. When an RTK fix was achieved, its best accuracy was measured slightly better than the specified 2 cm horizontal and 6 cm vertical. In the test illustrated in Figure 44, the RTK fix mode was maintained throughout the lap while 9 satellites were in view. This resulted in a perfect trace of the courtyard with the worst accuracy peaking at 6.2 cm. No post processing has been done on the data. In the 5 m x 1 m excerpt of Figure 44, the car was running at 4.2 km/h, and the update rate of 10 Hz was maintained resulting in the plotted data.

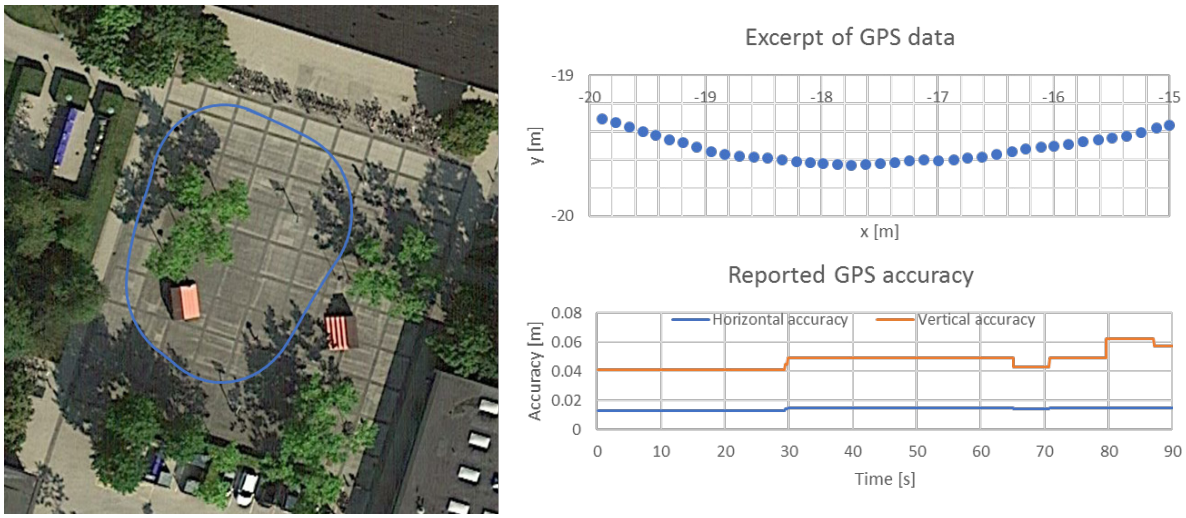


Figure 44: GPS data from a lap on Produktionstorvet in Kongens Lyngby with a 5 m x 1 m excerpt and the reported accuracy over time

In Figure 45, a recording of GPS data from five laps can be seen. The data is colored depending on the status reported by the GPS. The highest accuracy is achieved in RTK Fixed mode, where all data points lie perfectly on the track with a single exception. Shortly after initialization, for a period of 3.7 seconds, the Piksi reported a position 2.9 km away from its actual position and 6500 m below the surface of the earth. All other points are consistent and provide the most accurate data in the system. RTK Float mode should still provide reliable data with a reported average accuracy of around 20 cm. In DGPS mode, the nominal accuracy drops to 1-2 m. In the figure, it can be seen that DGPS can be very inaccurate, reporting locations far outside of the track. It falls back to this mode, or the even more inaccurate SPP (single point precision) mode in locations where the GPS reception is poor, such as under the bridge or close to buildings. Interestingly, the location provided by the Android smartphone continues to update even when the car is under the bridge. This is probably the result of the system combining data from the GPS, the IMU and the GSM network to provide the location, something that could also be implemented on the vehicle's system as part of the Navigation node.

At the time of the competition, the firmware of the Piksi Multi did not have the ability to provide heading or bearing information. It has since been updated to provide this functionality. Figure 47 shows bearings calculated post-event based on the latitude and longitude data provided by the GPS using the "Haversine" formula.

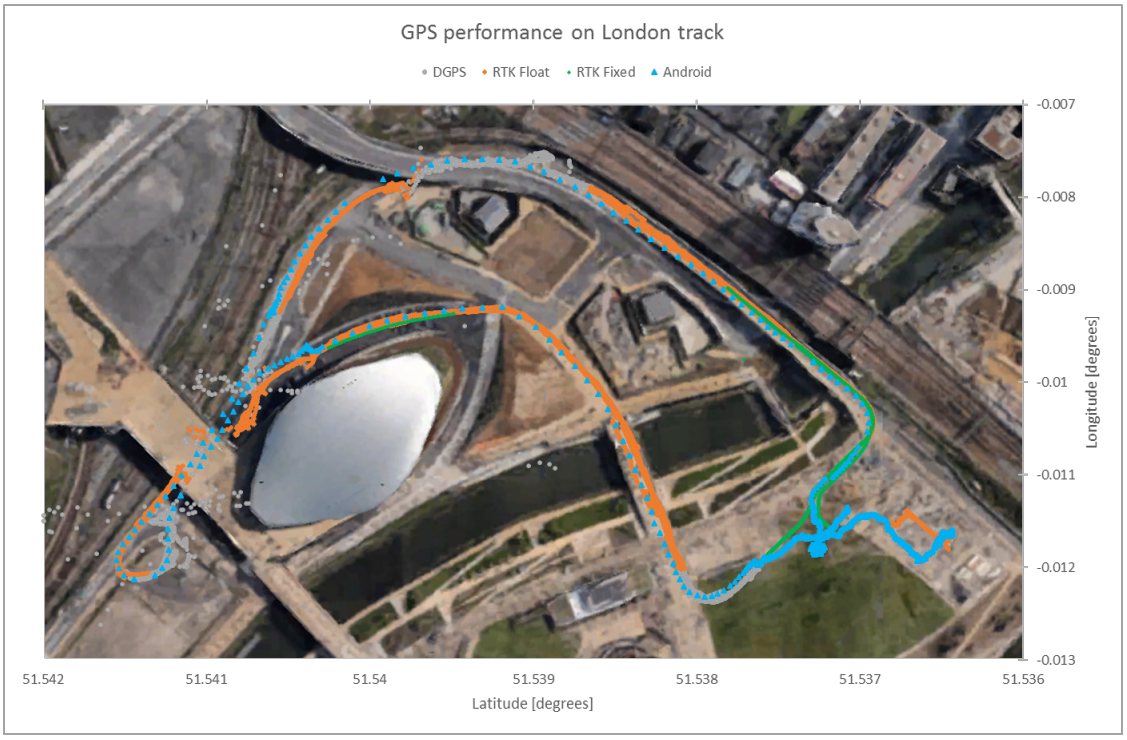


Figure 45: GPS performance of the Piksi Multi and an Android smartphone on the London track

4.6 Odometry

The wheel sensors were intended to function in full quadrature mode, but due to an incompatibility with the wheel mount and too large sensors, the phase difference between the two sensors was not guaranteed to be 90 degrees. Therefore, they were simply used to count the number of edges detected. While this problem affected the speed reading, as can be seen by the noisy data in Figure 46, it does not seem to have severely affected the distance reading. The laps can be recognized by the fact that the car must come to a full stop each time it passes the finish line. The length of the track as measured by the GPS readings is 1583 m, or 6332 m after 4 laps. After 4 laps, the distance as measured by the wheel sensors is 6316 m, only 16 m or 0.25 % less than the GPS distance. This difference could be caused by inaccurate GPS readings or errors in the wheel sensor readings. The difference compared to the official track length of 1577 m can come from the fact that the vehicle does not always run exactly in the middle of the road.

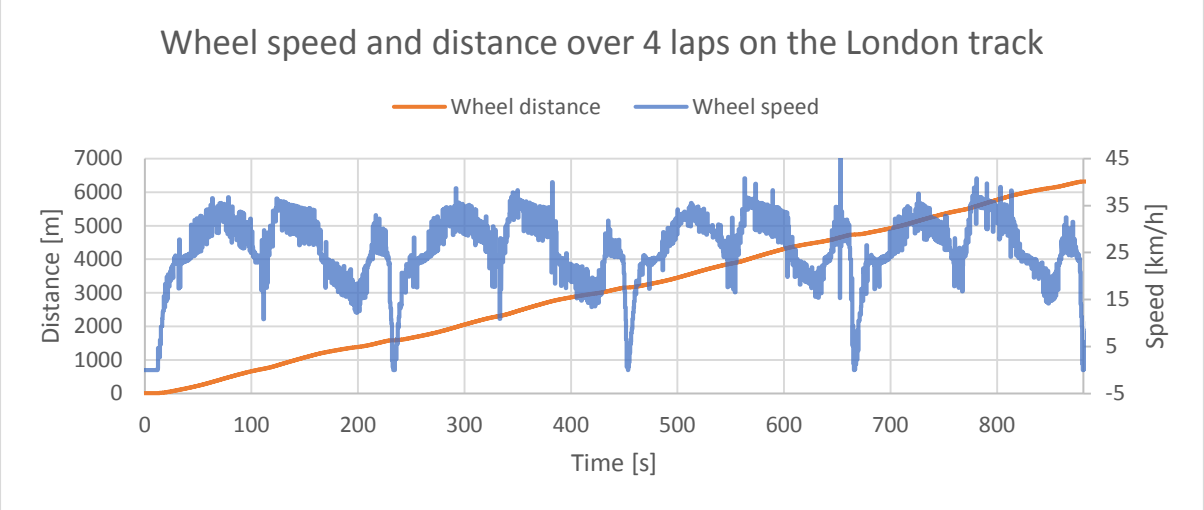


Figure 46: Wheel speed and distance over 4 laps on the London track

Using the same 4 laps to provide data, the performance of the gyroscope can be analyzed. Figure 47 shows the gyroscope data compared with bearing data obtained from the GPS data. The gyroscope data is clearly coherent and mostly matches the GPS bearing. After 4 laps, there is a divergence of 13 degrees, which could be caused by gyroscope drift. To further validate this data, the steering potentiometer could also be included.

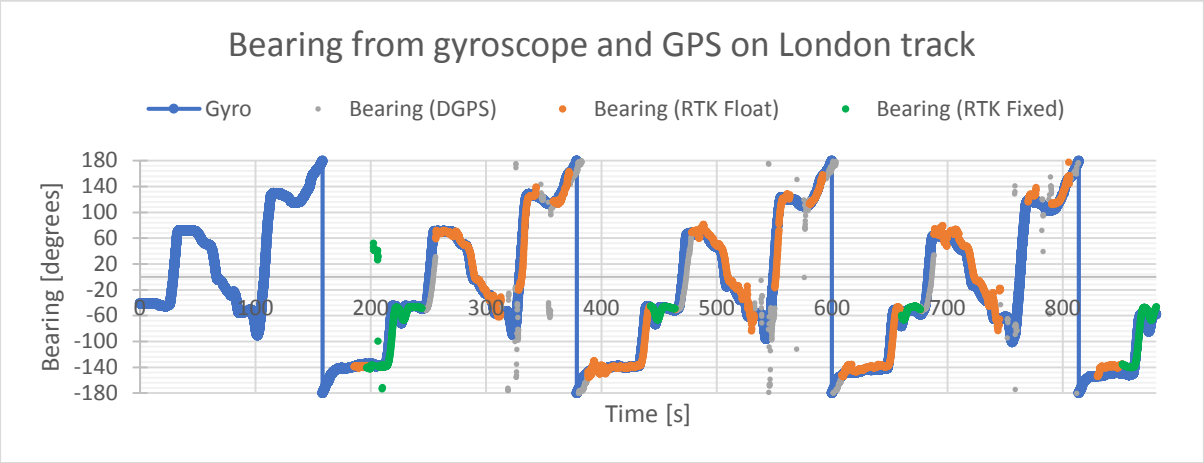


Figure 47: Gyroscope and GPS bearing over 4 laps on the London track

For an estimate of the initial global heading of the system, the magnetometer is to be used. In Figure 48, the magnetometer and gyroscope data is illustrated with the vehicle driving on a track with the shape of a rounded rectangle. The magnetometer data is very noisy, but can, with sufficient samples, be used to roughly determine the initial heading.

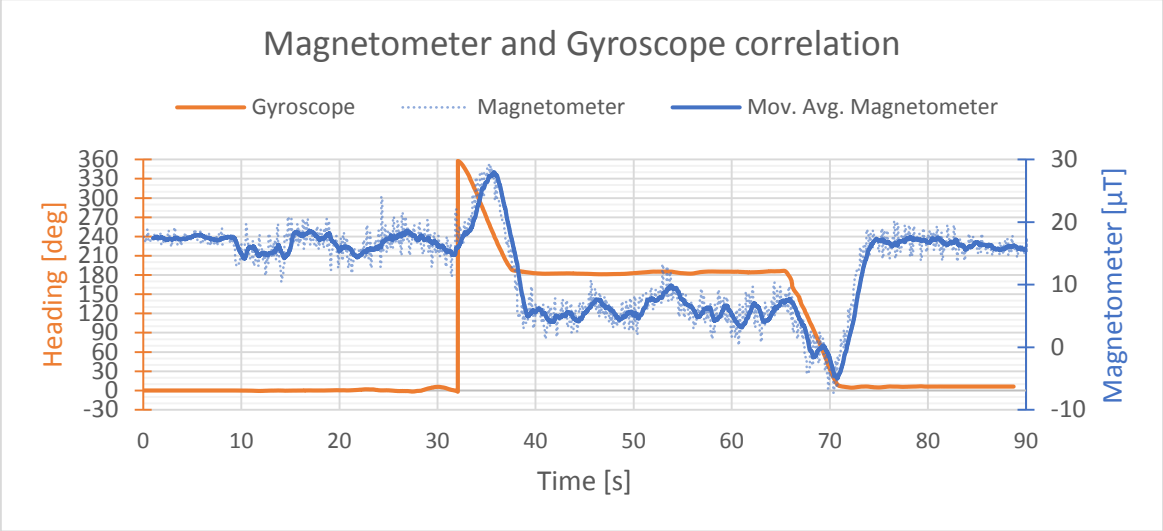


Figure 48: Correlation between magnetometer and gyroscope data for heading determination

4.7 Power

System state	Power [W]
Standby	8.2
Idle	17.6
Ready	38.5
Motors running	51.2

Table 4: Power consumption depending on system load

The power measurements were carried out with an amperemeter connected in series with the voltage supply to the system.

With the computer turned off and all devices connected, and on standby, the system consumed 8.2 W of power. At 2.8 W, the computer is the biggest consumer in this state, as it is also delivering 5 V power to all serial devices connected by USB, for example the 4G modem consuming 0.5 W. It is closely followed by the Piksi Multi board at 2.7 W. The third highest consumer is the Auto board, at 1.4 W, since it does currently not have a standby mode and is constantly supplying the side LIDARs and encoders.

After turning on the computer, the idle system consumes 17.6 W. The computer makes up 7.7 W and the touchscreen display 4.1 W of the total consumption.

With the system ready and all the nodes running, the computer uses an additional 4.6 W. Once the motors are engaged, but static, the system uses a total of 38.5 W of power.

With both motors moving their typical load, the total average power consumption is 51.2 W. The maximum power peak was measured at 64.9 W.

Overviews of the power consumption can be seen in Figure 49 and Table 4.

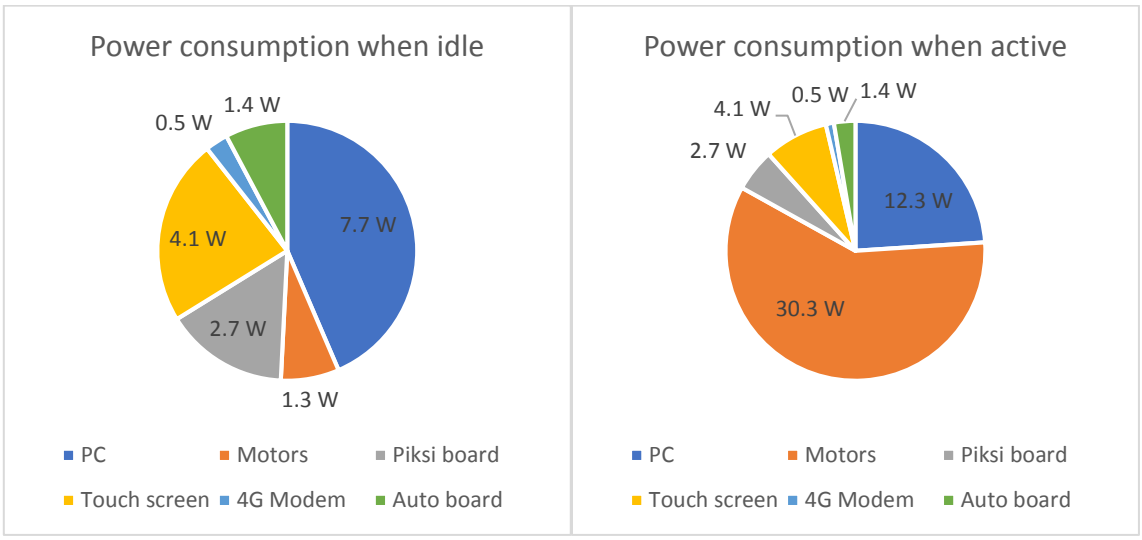


Figure 49: Pie charts of power consumption by device when system is idle or active

5 Future Work and Improvements

5.1 Known errors and possible solutions

5.1.1 Brake motor torque

The torque of the brake motor is sufficient for mild deceleration, but not strong enough for rapid action. The current motor can reach around 10 bar in hydraulic pressure before it starts to lose steps. For efficient braking, at least 30 bar should be obtainable. To reach this, either a higher gear ratio could be used, reducing the output speed but increasing the torque of the existing motor, or an all new motor could be found, fulfilling the criteria.

5.1.2 Control loop tuning

While the steering control loop works perfectly, the brake loop could still be optimized. For this, the system response, as well as the transmission delays from the pressure sensor to the computer, could be measured to better characterize the system. Based on this, the robustness and response of the control loop could be improved. Another improvement that could have a positive effect would be the increase of the CAN synchronization frequency. The CAN bus currently only updates at 10 Hz even though it should be capable of at least a 100 Hz update rate.

Furthermore, it could be beneficial to add a quick calibration setting, when the full calibration of the steering is not required.

5.1.3 LIDAR mount, enclosure, and connector

The connector to the LIDAR broke within short time on the London track. While the LIDAR has been returned to the manufacturer for repair, the mount would probably require some improvements to ensure that the cable is stress-free and to reduce the vibrations on the sensor. Furthermore, the exposed electronics of the LIDAR also require some protection, for example in the form of a 3D printed enclosure.

5.1.4 Socket communication robustness

The socket communication used to communicate between the apps and the computer work perfectly as long as a reliable connection exists. Under weak signal conditions or in the event of a loss of connection, the reconnection attempts often fail or require significant time to reestablish communication. For a critical component, such as the remote, but also for the time-critical RTK data, this should be improved to ensure robust operation similar to the USB communication used by the sensor nodes.

5.2 Improvements

5.2.1 Localization and Navigation

The sensor fusion for localization as well as the navigation has not been implemented yet. The data from the gyro, wheel sensor, and GPS should be combined to an estimate of the car's pose. Based on a predefined map and the live LIDAR data, a path should be planned. The necessary steering wheel position should then be set for the car to drive on the desired path.

The current envisioned architecture relies on odometry to provide the car's relative position and GPS to provide the global position. When a high accuracy GPS solution is available, it can be used to eliminate drift accumulated in the odometry sensors. The predefined map would contain a number of ordered coordinates through which the car would try to drive. Control over the speed of the car is reached by associating a maximum and minimum speed for each set of coordinates. If the current speed of the car is below the minimum speed, the car would accelerate until just below its maximum speed. If the maximum speed is exceeded, brake power would be applied. This follows the "coast and burn" strategy which increases fuel efficiency by driving the combustion engine at its optimal rotational speed (full throttle). The optimal path and speed can be simulated beforehand, as soon as the track data is available, similar to what the DTU Roadrunners team already does to prepare the driver for the race.

A lot of work on localization and navigation has already been performed by A. Bogdanowski [17] and A. Bellina [18] in their synthesis projects, but mostly in simulation and without real-world data. There is still a lot to be done, analyzing the collected data, and implementing a system in practice that is able to localize and navigate based on this data.

5.2.2 Stability and testing under various conditions

G. Dimitrakopoulos ([19]) defines "full autonomy" as the ability to maintain good performance under significant uncertainties in the environment for extended periods of time and the ability to compensate for system failures without external intervention. To reach this goal and increase system stability, a lot of testing needs to be done to ensure the system can gracefully handle different situations, environments, and failure causes. Analyzing the many variations in race conditions that can occur, and making sure the system is prepared for them, is also a big project.

5.2.3 Live feedback and visualization

Both for the driver and the spectators, as well as for processing the data later, good visualizations can be very useful. Currently, some data is streamed to the web server and visualized there, but there is much more data from the sensors and the operation of the system that can be displayed live. This includes what the system can see, but also what it is planning to do. The speaker also has potential, being able to provide audio-visual feedback in combination with a screen.

In addition to this, there may be a part of the competition which focuses on good visualization for the spectators of the event, making this improvement even more important.

5.2.4 GPS and LIDAR Lite mounts

In their current implementations, the GPS antenna and side LIDAR mounts are functional, but lack in practicality. The side LIDARs should be integrated into the chassis with the optical sensor being flush to the body of the car. The GPS antenna is currently mounted on the front part of the shell as well as the monocoque, making it difficult to disassemble the car quickly. A mount on the existing hole in the rear part of the shell would make assembly and disassembly of the car easier.

6 Summary and Conclusion[2]

As a result of this thesis, DTU Dynamo has been successfully expanded with autonomous capabilities.

Based on the requirements relevant to the participation of DTU Dynamo in the Autonomous UrbanConcept category of the Shell Eco-Marathon, devices were chosen and installed. Through the interaction of hardware design, embedded software, Android software and computer software, the sensors and actuators were integrated into a coherent system. This system achieves simple autonomy, following a predefined path as demonstrated to HRH Prince Joachim and the press, and with further work on navigation, will be able to run fully autonomously on the race track.

All the requirements in focus from Section 1.2 have been accomplished as part of this thesis:

- The system is able to reliably control the position of the steering wheels through the steering motor using a closed-loop system with an encoder and potentiometer.
- The system is able to reliably control the braking power through the brake motor using a closed-loop system with an encoder and pressure transmitters.
- The GPS, gyroscope, wheel sensor, laser scanner, side LIDARs and IMU form the sensory basis for localization and obstacle detection.
- Data was collected from these sensors both before and during the Shell Eco-Marathon 2017.

Furthermore, the following accomplishments going beyond the focus of the problem statement can also be highlighted:

- The remote-control app allows the car to be controlled remotely using any Android powered smartphone.
- The Piksi base station app overcomes the limitations of 2.4 GHz radio and allows the Piksi Multi to send RTK correction data over the internet.
- The car was able to steer itself and retrace the path of a prerecorded lap.
- Analyzing the collected data proved that the sensors provided useful data.
- Participation in the Shell workshop for the upcoming Autonomous UrbanConcept class allowed an insight into the shape and form of the competition.
- Following successful application submission and presentation, the project was mentioned at the Shell Eco-Marathon as the runner-up for the Technical Innovation award.

As specified in the problem statement, this project focused on some aspects of autonomous control. Taking the project further, "full autonomy" is desired, where the car can perform reliably even under significant uncertainties in the environment and can compensate for system failures without external intervention. This thesis has contributed with significant advancements towards this goal, bringing the DTU Dynamo closer to full autonomy and creating a platform for future projects.

[2] The self-evaluation can be found in a separate document as Appendix F, as per the DTU master's thesis rules.

7 Bibliography

7.1 References

[1] Shell, *Shell Eco-Marathon [Online]*, Available: http://www.shell.com/energy-and-innovation/shell-ecomarathon/about.html

[2] Tesla, *All Tesla Cars Being Produced Now Have Full Self-Driving Hardware [Online]*, Available: https://www.tesla.com/blog/all-tesla-cars-being-produced-now-have-full-self-driving-hardware

[3] R. Siegwart et al., "*Introduction to Autonomous Mobile Robots*", The MIT Press, 2004

[4] SAE International, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, J3016, 2016

[5] Oxford Robotics Institute, *Make the Future London [Online]*, Available: http://ori.ox.ac.uk/projects/sem/

[6] Phidget, *Libraries and Drivers – Linux Libraries (Source Code) [Online]*, Available: https://www.phidgets.com/docs21/Language_-_C/C%2B%2B#Libraries_and_Drivers

[7] Sigrok, *libserialport [Online]*, Available: http://sigrok.org/api/libserialport/0.1.1/

[8] Swift Navigation, *Specification and Bindings for Swift Binary Protocol [Online]*, Available: https://github.com/swift-nav/libsbp

[9] JsonCpp Authors, *JsonCpp – A C++ library for interacting with JSON [Online]*, Available: https://github.com/open-source-parsers/jsoncpp

[10] Madratman, *ROS driver for Lightware SF-40/C [Online]*, Available: https://github.com/madratman/lightware_sf40c_ros_driver

[11] Paul Bouchier, *swiftnav_piksi [Online]*, Available : https://github.com/Paul-Bouchier/swiftnav_piksi

[12] Controlwear, *Virtual Joystick Android [Online],* Available: https://github.com/control-wear/virtual-joystick-android

[13] MAX, *SlideView [Online]*, Available: https://github.com/MAXDeliveryNG/slideview

[14] Bolder Flight Systems, *MPU9250 [Online]*, Available: https://github.com/bolder-flight/MPU9250

[15] B. Blanchon, *ArduinoJson [Online]*, Available: https://github.com/bblanchon/ArduinoJson

[16] Garmin, *LIDAR-Lite v3 Arduino Library [Online]*, Available: https://github.com/garmin/LIDAR-Lite_v3_Arduino_Library

[17] A. Bogdanowski, "*Fitting Shell Eco-Marathon car with autonomous functions*", Synthesis report, DTU Automation and Control, 2017

[18] A. Bellina, "*Fitting Shell Eco-Marathon car with autonomous functions*", Synthesis report, DTU Automation and Control, 2017

[19] G. Dimitrakopoulos, "*Current Technologies in Vehicular Communication*", Springer, 2017

7.2 Table of Figures

7.3 Table of Tables

Appendix A. Source files for software and hardware designs [D]

The source files for the ROS software, Android software, embedded software, EAGLE designs, 3D prints and website can be found in a ZIP folder marked as Appendix A as part of the digital appendices.

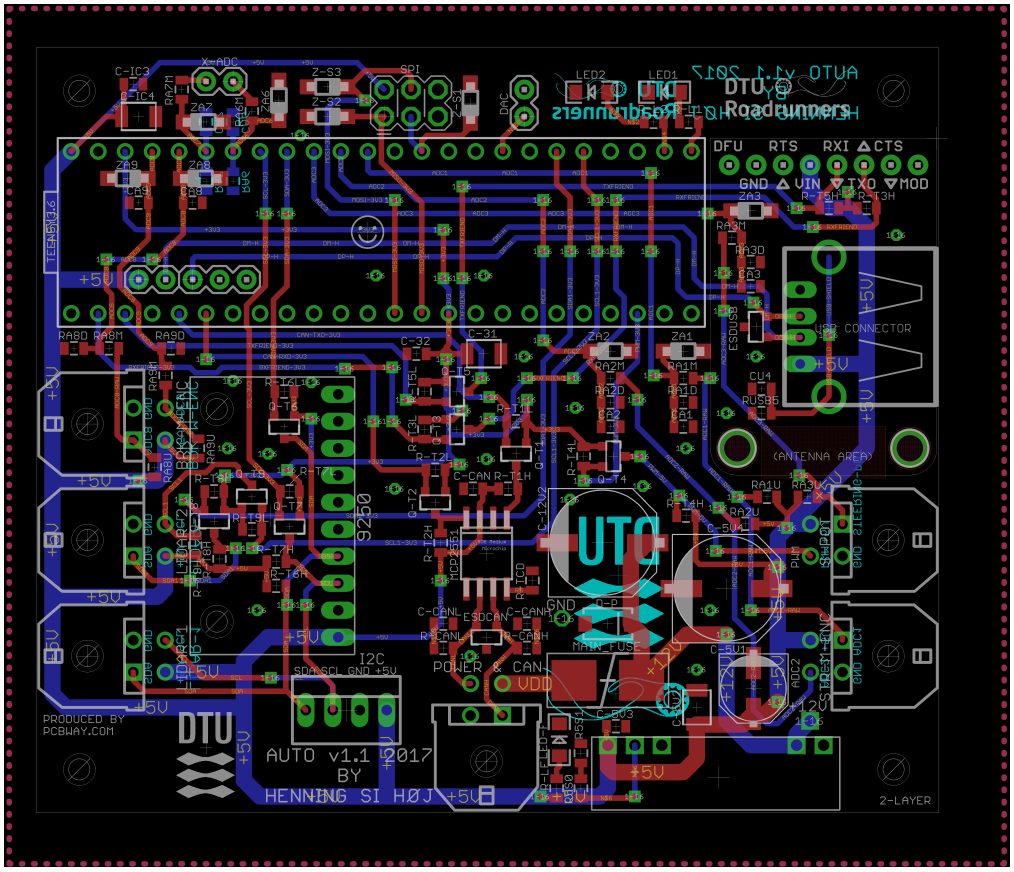
Appendix B. Design and Implementation of Electronics in the DTU Ecocars [D]

This report describing the electrical and electronics system in DTU Dynamo 12.0 can be found as part of the digital appendices.

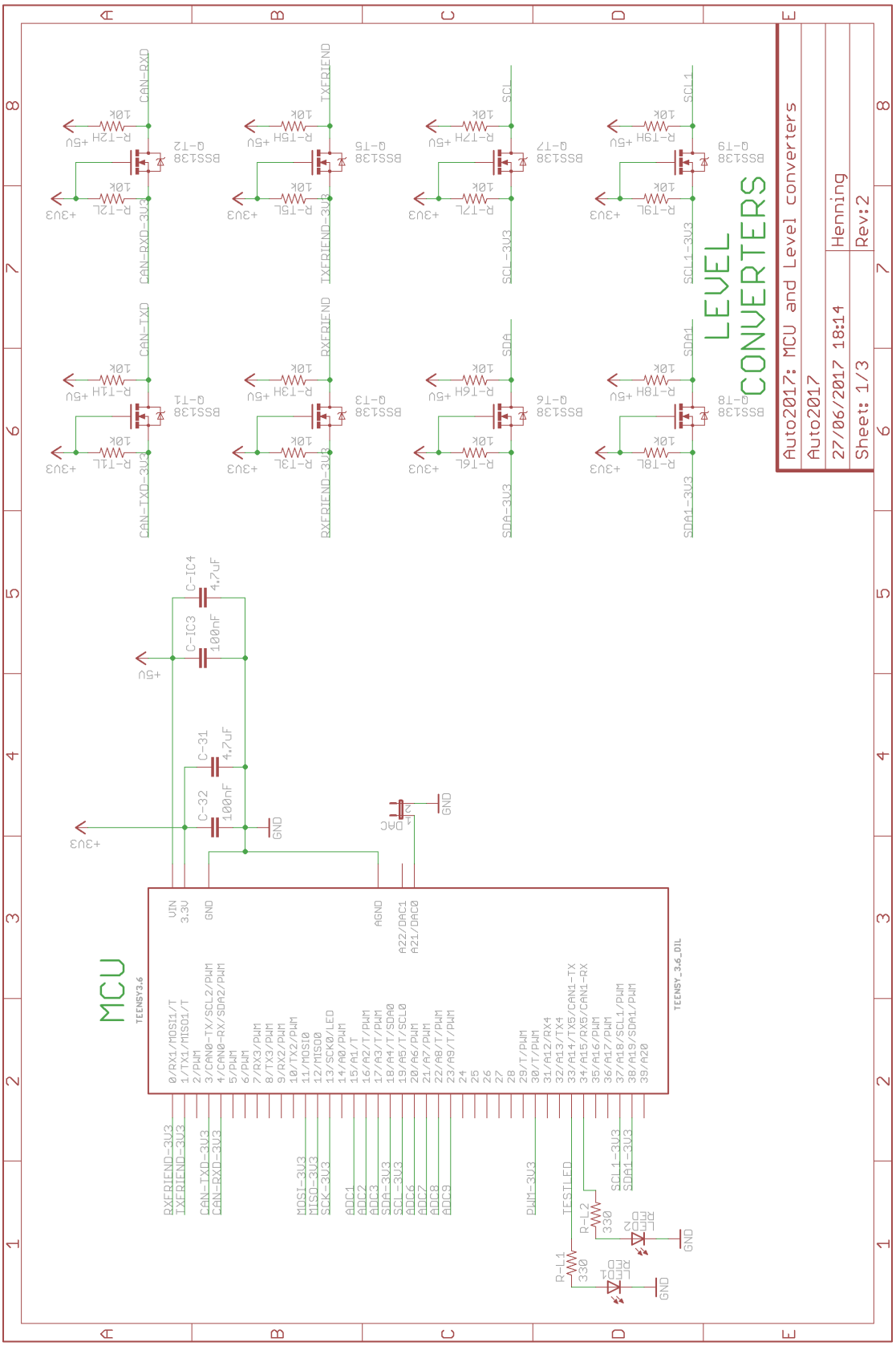
Appendix C. Application for Shell Eco-Marathon 2017 Technical Innovation Off-Track Award [D]

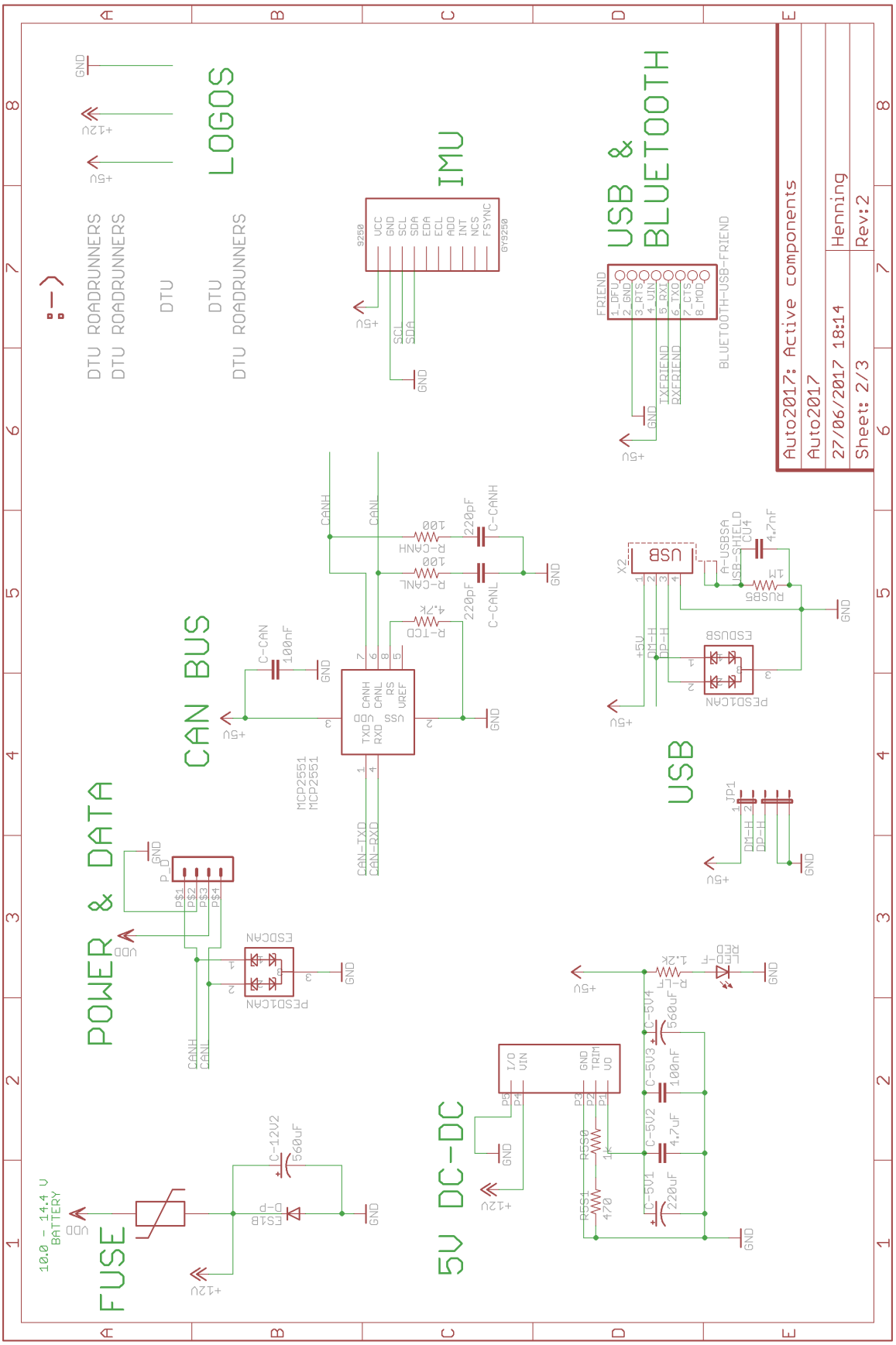
This application for the Shell Eco-Marathon 2017 Technical Innovation Award gives a short summary of the status of the work before the competition and is part of the digital appendices.

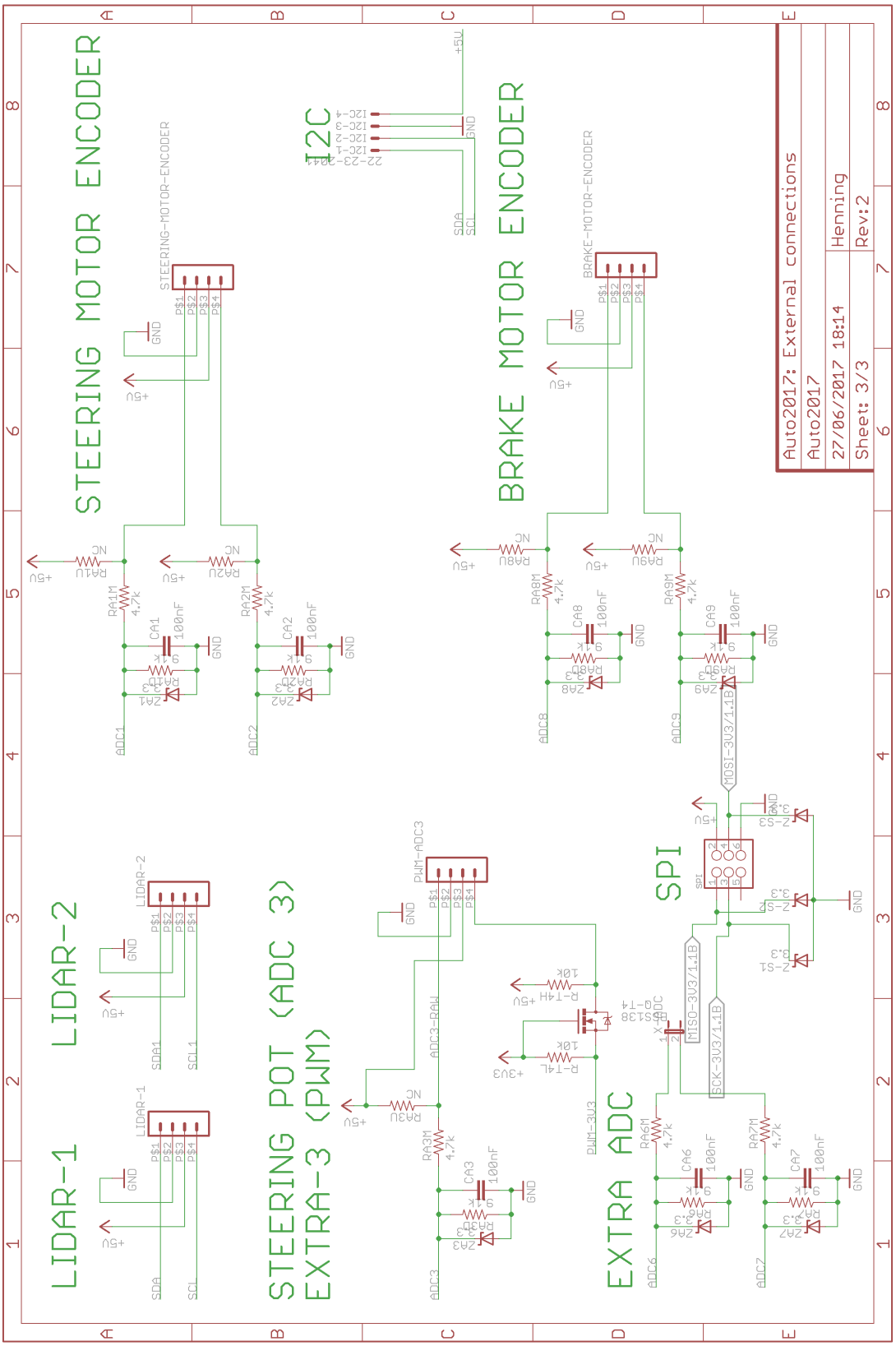
Appendix D. Auto 2017 1.1 Board



Appendix E. Auto 2017 1.1: Schematics




DTU ROADRUNNERS
DTU ROADRUNNERS
DTU
DTU ROADRUNNERS
LOGOS
IMU
USB & BLUETOOTH
CAN BUS
POWER & DATA
USB
FUSE
5V DC-DC
Auto2017: Active components
Auto2017
27/06/2017 18:14
Henning
Sheet: 2/3
Rev:2



Appendix F. Project plan and self-evaluation

The project plan and self-evaluation appendix is enclosed in a separate document according to the master's thesis rules.