Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Project Course in Master of Science in Engineering - Dependable Systems

PROJECT AUTOBIKE

Viktor Aronsson Karlsson
vkn17002@student.mdh.se

Love Bridén
lbn17012@student.mdh.se

Maya Zawawi
mzi17001@student.mdh.se

Gabriel Sherif
gsf17001@student.mdh.se

Examiner: Ivan Tomasic, Nikola Petrovic, Luciana Provenzano
Mälardalen University, Västerås, Sweden

Supervisor(s): Henrik Falk
Mälardalen University, Västerås, Sweden

Company Supervisor(s): Jonas Sjöberg
Chalmers University, Gothenburg, Sweden

09/01/2022

Acknowledgements

We would like to highlight and give our thanks to the sponsors from the industry, for making this project possible: Volvo Cars, Veoneer, Autoliv, Cycle Europe and Asta Zero.

We would like to give our thanks and gratitude to our supervisor Henrik Falk for supporting us throughout the project.

We would like to thank the project owner Jonas Sjöberg for trusting and guiding us throughout the project.

We would like to thank Yixiao Wang for providing his support and advice.

We would like to thank our fellow students working at C2, for providing their support and knowledge.

We would like to thank our Examiners Ivan Tomasic, Nikola Petrovic and Luciana Provenzano for providing feedback and advise throughout the project.

Lastly we want to thank all the previous members of the Autobike project from both Chalmers and Mälardalen, for all their work and knowledge that made the continuation of the this project possible.

Abstract

Author: Maya Zawawi

This AutoBike project is a result of cooperation between different parties since 2017 with Mälardalens university, in order to achieve an autonomous bicycle platform that will be used in the future to test the Volvo's cars safety system in order to reduce the collisions against the cyclists on the roads. This iteration of the AutoBike is a continuous phase of a previous project in Chalmers University. The main goal was to minimise the hardware size to make it easier to be portable from one bicycle to another, and to make it more dependable, moreover to modify the software to be fully controllable from the New software. The project succeeded to reduce the size of the hardware in a way that saved much room for any extra functionality needed in the future, and reduced the disturbance produced from some sub-circuits and components, as well a new LabVIEW code has been created to read the signals from the sensors and to control the actuators of the AutoBike. Of course there are some modifications have to be done in the future to developed both hardware and software to achieve a fully autonomous AutoBike that meets the stakeholder's requirements.

Contents

List of Figures

List of Tables

1. Introduction

Author: Viktor Aronsson Karlsson

This section of the report will give an overview of the Autobike project as a whole. Firstly, an introduction into the Autobike project will be given, followed by the project goals. Lastly, a short description of the content in this report.

1.1. The Autobike Project

The Autobike project has been a multi-year project between Mälardalens University and Chalmers University, with the explicit purpose of developing an autonomous bicycle that will be used during the validation process for Volvo's autonomous car avoidance algorithms. The concept of fully self-driving vehicles are becoming more of a reality for every passing day, this means that the future avoidance algorithms need to be as accurate as possible. Cyclists are one of the most unpredictable and vulnerable objects in the traffic scenario, with their often lacklustre protection gear and their unpredictable movements. Due to the unpredictability, the need for a more accurate depiction of a cyclist were required when testing autonomous vehicles. The Autobike projects end goal is to create a test platform that acts less like a computer driven vehicle and more like a real cyclist were operating it.

The previous iterations of the Autobike project explored the different areas needed for the creation of an autonomous bicycle. This includes, but not limited to, various control algorithms for both the balancing control and steering control, different operating platforms such as the National Instruments(NI) RoboRIO, Raspberry Pi 3 and BeagleBone Black/Blue.

1.2. Project Goals

The goals of this iteration of the Autobike project was the creation and implementation of a new hardware and software designs. Majority of the hardware components were replaced with new ones and the new software implementation was created in LabVIEW. In the list below are the desired goals that were set for this iteration of the Autobike project.

- Replacement and assembly of the new hardware components
- Reorganisation and redesign of hardware placements
- Creation of the LabVIEW architecture
- Implementation of the Python Control codes
- Validation of the accuracy for the gathering of sensor values
- Validation of the execution time of the LabVIEW code, to be under 100Hz
- Safety analysis of the Autobike system
- Construction of requirements for the project

1.3. Content of the report

The report describes the tasks and work performed by students during the project course at Mälardalens Högskola/Universitet in the year 2021. The report starts of by introducing the project and presenting areas related to the project. Followed by the work process throughout the project, the work process is divided into six categories: Hardware, Software, Verification & Validation, Requirements, Safety and General. The sections that follows covers the work performed during the project, divided into the previous mentioned categories excluding general. Lastly the results from the project is presented and discussed.

2. Background

Author: Maya Zawawi

According to the National Society for Road Safety (NTF) is a nongovernmental organisation that works to improve road safety. In the year 2019, 2000 cyclists out of 3850 were injured, and 222 cyclists were seriously injured, which means that they needed a kind of medical care depending on their injuries. Which in some cases can be very serious that can affect their lives for a really long period of time.[1]
The Autobike project is a cooperative project between Chalmers and Mälardalens Universities since 2017, in order to develop an automated bicycle, that will be tested by Volvo cars on the field to reduce road accidents against cyclists. The latest iteration of the AutoBike project that has achieved by Mälardalens University was on 2019 where the students succeeded to balance the bicycle with dummy mounted on it, the bicycle was designed to be able to has a steering angle and forward velocity only , and the students could hide the electronic in a basket that fixed on the front frame.[2]

3. Related Works

Author: Maya Zawawi

Fundamentally, the bicycle is a nonlinear system that can be affected easily by the surrounding environment and factors, which created a considerable challenge to the researchers in order to find the best methodologies to achieve the balanced and controllable unmanned bicycle at different velocities and paths. In the following section different approaches that have been carried through other engineers and researchers will be discussed to highlight different solutions that can maintain the balance and steering the autonomous bicycle.

3.1. Gyroscopic Balancer Controlled by FSMC and AFSMC

Hung Chi. C & Jen Chou.J succeeded to modify a riderless bicycle that can balance an unmoved bicycle that can resist the surrounding disturbances, as well as for moving one in both forward motions and turning paths, The researchers could prove that using a gyroscopic balancer that is controlled by a Fuzzy Sliding Mode Controller (FSMC) and an Adaptive Fuzzy Sliding Mode Controller (AFSMC) has better performance when it comes to steer a nonlinear system compare with other balancers such as PID, because it provides a faster response of the system, reduces the ratio of mass, and has a quite larger moment. As a result of using the gyroscopic balancer specifically that is controlled by FSMC and AFSMC respectively the bicycle could preserve the lean angle at zero degree while the steering angle is 37 degree even when the riderless bicycle has been tested and validated against, uneven territory, external disturbances and jolts.[3]

3.2. Design of Mechanism of Self-Driving Bikes

This paper was published by students from Chalmers University in the Spring of 2021, the purpose of this paper was to improve the existing hardware of self-driving bicycle with a new design of the hardware. The thesis focused on developing the hardware only, especially the steering, using an encoder and a new forward motor that can be simulated in SIMULINK. The paper pointed to re-design a new circuit that has both high and low voltage components and this circuit will be implemented in one box, this requirement is similar to MDU project when it comes to redesigning the hardware and implementing it in one smaller box.[4]

4. Method

4.1. General

Author: Love Bridén

The work within the group was split among the members of the group. Two persons were part of the hardware team and two persons were part of the software team. In addition, one of the four roles requested by the course (requirements manager, safety manager, verification and validation manager and quality assurance manager) were given to each of the members of the team. At the beginning of the iteration of the project the group worked together to reach a consensus of what was supposed to be done. After the initial weeks the work was split according to the roles. Weekly meetings were held to give the group as a whole an understanding of what has been done, what was supposed to be done, eventual problems that arose and help with those eventual problems were given if possible. Apart from those meetings the work was mainly split according to the roles as mentioned, of course help was received outside of those meetings when requested. When questions arose they were addressed by the group to reach design choices that fit the goals of all the roles. In conclusion the work was conducted as a group and individually while consolidating the other group members for the solutions to fit the project.

4.2. Hardware

Author: Love Bridén

The work related to the hardware of the Autobike during this iteration of the project began with a study of what has previously been done in the project. Later the old electronic hardware was removed entirely from the bicycle. Following, the electronic architecture was initially designed by studying the documentation of the previous iteration of the project. The documentation of the previous iteration were lacking, hence examination of some components and sub circuits were necessary. These examination efforts aided the design choices for those sub circuits of the architecture. When, finally, the desired functionality was reached while prototyping during the examinations the resulting circuit and other information were documented in appendices C and B. With a functional design the optimisation began. The optimisation was conducted by swapping the components (e.g. microcontroller and forward motor controller) the stakeholder explicitly requested. To be able to use those new components some modifications had to be done e.g. the communication protocol between the microcontroller and the forward motor controller was changed to Universal Asynchronous Receiver-Transmitter (UART). Later, redundant or faulty components were identified (e.g. the old circuit breakers) and alternatives were located from online vendors. The search for components was difficult due to the shortage of components worldwide due to the pandemic. Researchers from Chalmers University of Technology identified problems within the PCB received at the beginning of this iteration. This sparked examination of the PCB (see appendix E) where the DC-DC (Direct Current) conversion capabilities were accidentally damaged. To compensate for this accident a new PDB had to be designed and manufactured (see appendices D H for detailed information). Later a hardware user guide (appendix A) was created to aid future iterations of this project. When the architecture was informally verified faults related to the so called "finder circuit" were identified and design choices were made to tackle these problems (see appendix G). These final modifications were the last activity related to hardware except for the documentation in this report and the showcase during the day of presentations of the projects at Mälardalen University this fall.

4.3. Software

Author: Viktor Aronsson Karlsson

When software is mentioned throughout the report in the context development, it is referencing the constructed LabVIEW code. All software throughout the project were developed on a windows computer that was connected via USB to the MyRIO. The development of the software started of with an examination of the previous years solutions, that were provided by Chalmers. The outcome from the examination with the help of the stakeholder goals were used to design an LabVIEW architecture, which consisted of multiple modules that will work in parallel. The modules within the architecture were developed independent from each other and as the modules were finalised, they were brought together for implementation testing.

4.4. Verification & Validation

Author: Maya Zawawi

The standard V-shaped model has been conducted in order to perform all tests and data analysis, to control if all stakeholders requirements have been met and modived, starting from low-level design, through implementation & coding phase, ending with high-level design.
All tests took place in Malardalens facilities , and carried off by using all needed instruments and equipment available in the Malardalens laboratory and workshop such as the function generator, stationary computer, multimeter, and oscilloscope. Discussions and observations between the group members have been conducted to reach the most accurate result of tests. See appendixR

4.5. Requirements

Author: Love Bridén

The method of eliciting requirements for this iteration of the Autobike project is described in detail in appendices O and P. To recap, meetings with the main stakeholder of the project laid the foundation of the work. The goals of the stakeholder were derived into stakeholder requirements with the aid of models. These requirements were later derived further into system, subsystem and component requirements using different models. Lastly the requirements elicited underwent a review process where eventual flaws in the requirements were corrected.

4.6. Safety

Author: Gabriel Sherif

The methods that have been used for this project were Hazop, Hazard and Operability Study. The controllability did not exist in the AutoBike project, so I adopted another method that was more adaptable to our project. I read a book that uses a standard hazard analysis, which can be used in any domain or industry. And a standard from a book written by Nicholas Bahr. For safety assessment, this book introduced various techniques and methods focusing on standardized safety processes - The different methods were easy to adapt to the AutoBike projects. The standardized format or concepts that have been used in this project for the safety development process are found in Hazard analysis (appendix T). They are the cause, effect, probability and severity index, and hazard risk index (HRI). All concepts have been modified and adapted to situations and the purpose of the AutoBike project. From previous experience, the Safety Manager has looked at the Tree Tree Analysis (FTA) and Waterfall Model. FTA was not beneficial enough for this project to look at complexity and redundancy in systems. Also, the Waterfall model was is next similar to the standard found in the book (ref to the book). But the chosen method was helpful enough to fulfill its purpose. The methods used to develop the Safety process part for this project have come from a book [5].

5. Hardware Components

Author: Love Bridén

This section describes what hardware is related to the development during this iteration of the Autobike project. Other hardware components such as the bicycle frame and wheels are included in the hardware of the Autobike but since they are unrelated to the development during this iteration of the project they are not mentioned further.

5.1. MyRIO-1900

Author: Maya Zawawi

In this iteration of the project Autobike uses the MyRIO-1900 that manufactured by National Instruments. It is a real-time embedded board, That develops applications that can take advantage of its on-board FPGA and microprocessor. The maximum current is 32mA, and the power supply voltage range is 6-16VDC. The window of NI myRIO-1900 must be facing away from the mounting surface to ensure the optimal operation environment ambient temperature near the device(IEC 60068-2-1, IEC 600682-2) 0 to 40°C, and storage temperature(IEC 60068-2-1, IEC 600682-2)-20 to 70°C. In order to connect and test myRIO the following must be considered, the USB maximum length is 2m, the maximum signal wires length is 30cm, and the keyholes of myRIO are sensitive to Electrostatic Discharge (ESD). The MyRIO-1900 has compact embedded device ten analog input (AI) and six output (AO), 40 digital I/O lines, on-board accelerometer, a Xilinx FPGA, and a dual-core ARM Cortex-A9 processor. [6]

5.2. IMU

Author: Love Bridén

In the Autobike an Inertial Measurement Unit (IMU) is included, the specific IMU included is the Pmod NAV designed by Digilent. The device has a supply voltage of 3.3V. The device features sensors that measure linear acceleration in three axes, angular rate in three axes, magnetic flux density and air pressure. The resolution for the pressure data is 24 bits, the resolution for the other sensor data is 16 bits. The linear acceleration sensor has a range of ± 16g, the angular rate range is ± 2000°/s, the magnetic flux density range is ± 16G and the pressure sensor data range is between 260 and 1260hPa. The device features communication through Serial Peripheral Interface (SPI) or Inter-Integrated Circuit (I^2C). [7]

5.3. GPS Antenna

Author: Love Bridén

The Autobike features a Global Positioning System (GPS) antenna, more precisely the GPS_WP/TRK/SMA_3.0 designed and manufactured by CTI. The antenna has a centre frequency of 1575.42MHz and a bandwidth of 5MHz. The antenna has a low-noise amplifier gain of 27dB at 5V. The antenna has an impedance of 50Ω and an operating temperature range of -40 to 85°C. The antenna has a voltage standing wave ratio of 1.5 to 1. The antenna is waterproof and has a magnetic mount. The antenna is accessible through a RG174 coaxial cable. [8]

5.4. GPS Module

Author: Love Bridén

In order to obtain GPS data from the GPS antenna of the Autobike a GPS module is used. The GPS module used in the Autobike is the ZED-F9P GNSS module designed by u-blox. The GPS module can operate in velocities up to 500m/s, at altitudes up to 50km above sea level and while under the influence of forces up to 4g. The GPS module can operate in temperatures between -40 and 85°C. The time pulse signal of the GPS module is configurable between 0.25Hz and 10MHz. The accuracy of the velocity data reading is 0.005m/s and the accuracy of the dynamic heading is 0.3°. The GPS module has a power supply voltage of 3.3V and consumes up to 130mA. The module features a backup battery that can supply the module with typical currents of 80μA and voltages between 1.65 and 3.6V. The module is capable of communicating with other components through UART, SPI and Universal Serial Bus (USB). [9]

5.5. Teltonika RUT950

Author: Love Bridén

In the Autobike system a router is included. The specific router included is the RUT950 designed by Teltonika. The router has two Subscriber Identification Module (SIM) card slots as well as two mobile and two WiFi antenna connectors. The router has one Wide Area Network (WAN) and three Local Area Network (LAN) Ethernet ports. The router also features LEDs for indicating the status of the different functionalities of the router. The router has a data-transfer rate of up to 150 Mbps. The router features WiFi security in the form of different WiFi Protected Access (WPA) versions and can support up 100 simultaneous users. Virtual Private Networks (VPN) is also a feature supported by the router. Hardware wise the router has an Atheros Wasp processing unit with a clock speed of 550MHz, a 128MB Double Data Rate (DDR2) Random Access Memory (RAM) and a 16MB flash memory. The router has a supply voltage of 9 to 30VDC, it consumes up to 7W and the device has an operating temperature range of -40 to 75°C. [10]

5.6. Hall Sensor

Author: Love Bridén

The Autobike features a speed sensor in the form of a reed switch. The specific reed switch included is Assemtech Europe's MPPSA 240/100 reed switch. The reed switch has a maximum switching voltage of 400VAC and a minimum breakdown voltage of 600VDC. The switch has a maximum carry current of 1A and a maximum switching current of 0.5A. The switch has a maximum contact resistance of 150mΩ and a minimum switching distance of 10mm. The switch is covered by a case consisting of blue nylon 66. The reed switch has an operating temperature range of -20 to 85°C. [11]

5.7. Encoder

Author: Love Bridén

The Autobike features a steering angle encoder by using Avagos HEDS-5540-A11 optical incremental encoder. The component has a supply voltage of 5V with a maximum ripple of 100mV peak to peak and an operating temperature range of -40 to 100°C. The encoder can operate in rotational speeds up to 30000 rotations per minute (rpm) and rotational accelerations up to 250000rad/s^2. The encoder has a maximum count frequency of 100kHz. The encoder has a maximum supply current of 85mA. The digital output channels of the encoder are under 0.4V for digital low and above 2.4V for digital high signals. [12]

5.8. Escon Motor Controller

Author: Love Bridén

The Autobike features a steering motor and steering motor controller. The steering motor controller in the Autobike is ESCON 50/5 designed by Maxon. The motor controller has a supply voltage between 10 and 50VDC. The maximum output voltage of the motor controller is 49VDC, its maximum continuous output current is 5A and its maximum pulse (under 20s) current is 15A. The motor controller features Pulse Width Modulation (PWM) at a frequency of 53.6kHz. The motor controller has a maximum efficiency of 90%. The motor controller features analog inputs with a range of -10 to 10V and a resolution of 12 bits. Analog outputs are also included in the motor controller, they have a range of -4 to 4V and a resolution of 12 bits. Furthermore the motor controller features digital inputs and outputs. The motor controller also features hall sensor signal inputs and 5VDC outputs that could power peripherals such as encoders and hall sensors. The motor features Light Emitting Diodes (LED) as a user interface. The operating temperature range of the motor controller is between -30 to 45°C, however, the range can be extended towards 85°C if the device is derated properly. The device can operate in altitudes up to 10km above sea level. The motor controller will switch-off if it receives voltages under 7.2V, above 58V, currents above 22.5A or operate in temperatures above 100°C. [13]

5.9. Vesc 6 mkV motor controller

Author: Love Bridén

The Autobike features a so called "forward motor" responsible for accelerating the Autobike, that motor is controlled by a motor controller referred to as the "forward motor controller". The forward motor controller is the Vedder Electronic Speed Control (VESC) 6 MK V. The motor controller can control three phase DC and brushless DC motors. The motor controller is powered by a DC source of between 6 and 60V and consumes continuous currents up to 100A (burst currents up to 120A). The motor controller can communicate via UART, USB and Controller Area Network (CAN). The motor controller can also be controlled using an analog signal or a Pulse-Position Modulation (PPM). The motor controller supports sensor inputs such as HALL sensors or other magnetic position sensors. The VESC can output 5V with currents up to 1A to supply power to low power consuming peripherals. The VESC also has a 3.3V output that can output 0.5A. The motor controller is configurable using the "VESC Tool", some of the configurable features are regenerative breaking, current output limits, high/low input voltage protection, current output limits, the Rotations Per Minute (RPM) of the motor, acceleration rates and protection against too high temperatures. The VESC features LEDs that indicate its status to a user. [14]

5.10. Batteries

Author: Love Bridén

A battery is included in the Autobike. Upon receiving the Autobike Turnigy's Graphene Professional 12000mAh 6S15C LiPo Pack was included. As the name implies it is a battery with six lithium polymer cells connected in series. The nominal voltage is 22.2V and the capacity of the battery is 12Ah. The discharge rate of the battery is 15C, during short periods of time the discharge rate is up to 30C. [15]

Purchased during this iteration of the Autobike was GensTattu's Tattu 12000mAh 22.2V 15C 6S1P Lipo Smart Battery Pack. This battery, just as the previously mentioned battery consists of six lithium polymer cells connected in series resulting in a nominal voltage of 22.2V for the battery. The battery has a capacity of 12Ah and a discharge rate of 15C (30C for short bursts). This battery features a Battery Management System (BMS) that indicates the status of the battery, features storage and self discharge functions, alerts if the battery reaches too high or too low voltages, alerts if the battery reaches outside of its operational temperature range and if the cells are unbalanced. [16]

5.11. Finder

Author: Love Bridén

Information about the "finder" can be seen in appendix C, further information is provided in this subsection.

The finder is a relay with two poles. The relay is rated for 20A continuous, it can tolerate currents up to 30A for brief periods of time. The relay has a breaking capacity of 20A when a difference of 30V is applied to either of the poles. The coil of the relay has a minimum switching load of 1W and is consumes 1.25W when operating in DC. The coil switches the state of the finder in 15ms and can return to the original state in 8ms. The finder is able to operate in a temperature range of -40 to 40°C. When 24V is applied to the coil it consumes 52.2mA when operating in DC. [17]

5.12. LED

Author: Love Bridén

The LED is briefly explained in appendix C, further information is provided in this subsection. Included in the Autobike is a ZBVB3 LED designed by Schneider Electric. The LED illuminates green light when a voltage of 24V is applied across its terminals and its current consumption is 18mA. The LED has an operating temperature range of -40 to 70°C. [18]

5.13. Reset button

Author: Love Bridén

In appendix C brief information about the reset button is presented, further information about the reset button is presented in this subsection. In the Autobike a reset button is featured. The button is Schneider Electrics ZBE101. The button allows for currents up to 10A to flow through it. The button is capable of insulating potential differences up to 600V applied across the two contacts. The button has an operating temperature range of -40 to 70°C. [19]

5.14. Emergency Stop Button

Author: Love Bridén

Information about the emergency stop button featured in the Autobike can be seen in appendix C. Further information is presented in this subsection. The button featured in the Autobike is Omrons A22-01. The button is rated for up to 380VDC. At 24VDC the button tolerates currents up to 10A to flow through it. The button can operate in ambient temperatures between -20 and 55°C. When pressed, the two poles of the button are disconnected from each other. [20]

5.15. Power Switch

Author: Love Bridén

In appendix C the power switch is briefly explained, further explanation of the component is presented in this subsection. In the Autobike a power switch is included, the specific switch is Arcoelectrics CC1350XB BLK/GRN. The switch has four 6.3mm blade terminals that in the default state are not connected. If the button of the switch is pressed the switch enters a second state where two of the four terminals are connected to each other forming a switch with two poles. The button has a green sliver that is visible when the switch is in its second state. [21]

5.16. Circuit Breaker

Author: Love Bridén

In the Autobike two circuit breakers are present. They are Kraus' and Naimers CG4 A220-600FS2. The breakers has four terminals where one is not connected and one is connected to one of the remaining two terminals dependant on the state of the circuit breaker. The state is changed by rotating the knob of the circuit breaker. The circuit breaker has a rated insulation voltage of 440V and a rated operational current of 10A. [22]

5.17. Steering Motor

Author: Love Bridén

The steering motor in the Autobike is Maxongroup's ECX TORQUE 22 L. It is a brushless DC motor with a nominal voltage of 24V. When the motor rotates without a load applied it reaches a rotational speed of 10300rpm and consumes 129mA. The motor has a nominal torque of 48.1mNm and a nominal current of 2.13A. The motor has a stall torque of 684mNm and a stall current of 31.1A. The DC motor has a maximum efficiency of 87.7%. The torque and speed constants of the motor are 22mNm/A and 434rpm/V respectively. The motor can operate in ambient temperatures between -40 and 100°C. The motor has a maximum speed of 11000rpm and a maximum axial load of 4N. [23]

5.18. Forward Motor

Author: Love Bridén

The so called forward motor, the motor accelerating the Autobike, is Shimano steps DU-E6010. The motor is a brushless DC motor with a voltage rating of 36V. The motor can provide a torque of up to 50 Nm. The motor is rated for continuous powers up to 250W and a burst power consumption of up to 500W. The motor weighs approximately 3.05kg. [24]

5.19. DC-DC Converter

Author: Love Bridén

In appendix D information about the DC-DC converter used in the Autobike is presented.

5.20. MISC

Author: Love Bridén

In appendix D information about the Power Distribution Board (PDB) can be found describing some of the miscellaneous hardware of the Autobike. Apart from these previously mentioned hardware components the Autobike includes connectors such as XT90, AS150, 6.3mm blade terminals and bootlace ferrules in different sizes. The system also features stranded wires in the sizes 12, 18 and 21 American Wire Gauge that connect components of the Autobike. To be able to connect the GPS antenna to the GPS module an SMA (SubMiniature version A) adapter (SMA socket to RP (reverse polarity) SMA plug) is used as the socket of the GPS antenna is not compatible with the wire connecting the GPS module with the GPS antenna. Breadboard jumper wires are also used for low power signals in the Autobike. In addition a PCB received by Chalmers University of Technology is used as an input/output for the MyRIO device, information about this PCB is found in appendices B and E.

6. Software

The following chapter will describe the software aspects of the project, the module that were developed, the areas surrounding the module and the behaviour of the software architecture.

6.1. LabVIEW

Author: Viktor Aronsson Karlsson

LabVIEW is an graphical programming environment tool, developed by National Instruments[25]. The tool was used for developing and execution the code. The LabVIEW code will be executing on both computing cores of the MyRIO-1900, which are a real-time and a FPGA core. For the develop of the LabVIEW code the environment was of the 32-bit variant and these three modules was used: myRIO Toolkit, FPGA and Real-Time.

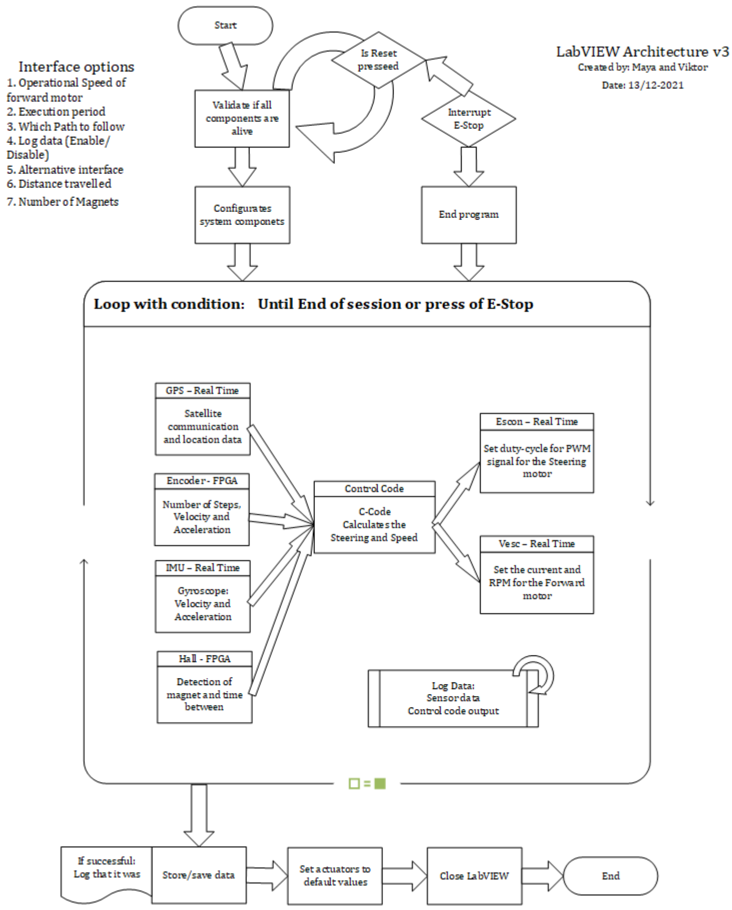


Figure 1: LabVIEW Architecture v1

6.2. The code architecture

Author: Maya Zawawi

The architecture in the figure 1 gives an overview of the whole LabVIEW code.
In this structure of LabVIEW, we will have a new executive order, semi-parallel execution. The code starts with validating if all components work as they are supposed to, then the code will calibrate the system parameters, if something goes wrong within the system, we will have an emergency stop to re-calibrate the parameters again. Then the LabVIEW code can run further into the main control loop. There we read all signals sent by sensors of the system through C code

that is integrated into the LabVIEW code. After that commands will be sent to the actuators and so on until the test is completed. Then the code will run out of the loop to save data set actuators to default values and stop LabVIEW from running.

6.2..1 IMU

Author: Viktor Aronsson Karlsson

The task of the IMU is to track the orientating of the Autobike, this is achieved by using a gyroscope to know the angular velocity and a accelerometer to know the angular acceleration, both being three axis sensors. When all six axis and the position of the IMU on the bicycle are known, the current and future orientation can be calculated.

The LabVIEW code runs on the real-time side of the myRIO, and utilises the myRIO toolkit low-level I2C VIs to acquirer the data from the IMU. Before the correct data can be acquired form the IMU, the IMU need to be configured, the configuration can be seen in the two first rows in table.1. When the IMU is correctly configured the code will enter the main acquirer loop, where LabVIEW will send a specific message to the IMU telling it which register data to send back, this is repeated for each of the six axis. The messages that are sent back to the myRIO are represented as a 16bit word in a two's compliment.

Register	**Description**	**Hex**
Accelerometer Configuration	Output Data Rate: 952Hz, Full-Scale: 4g	0x20 | 0xD0
Gyroscope Configuration	Output Data Rate: 952Hz, Full-Scale: 245dps	0x10 | 0xD0
Gyroscope X:	Angular Pitch rate	0x18
Gyroscope Y:	Angular Roll rate	0x1A
Gyroscope Z:	Angular Yaw rate	0x1C
Accelerometer X:	Acceleration X-axis	0x28
Accelerometer Y:	Acceleration Y-axis	0x2A
Accelerometer Z:	Acceleration Z-axis	0x2C

Table 1: IMU register

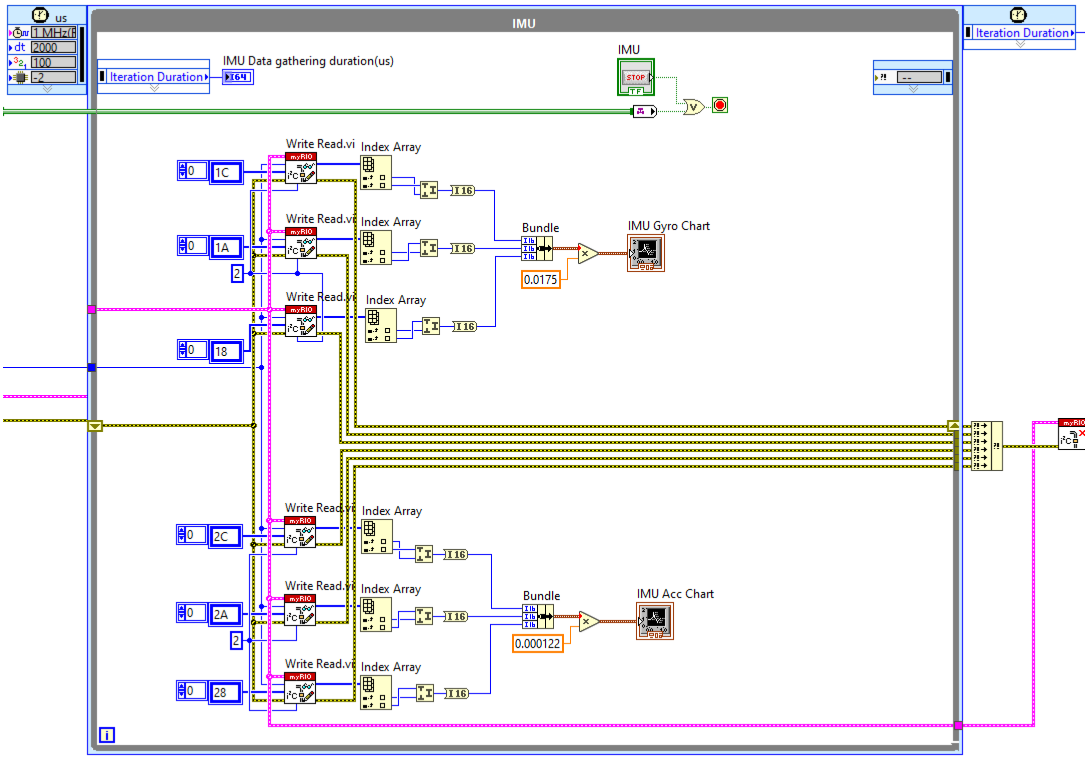


Figure 2: LabVIEW Real-time code for the IMU

6.2..2 Hall Sensor

Author: Viktor Aronsson Karlsson

The hall sensor code operates on the FPGA side of the myRIO. The FPGA reads the digital input and depending if it's digital high or low, it will act differently. When it's digital high it will be in a idle state. When state switches to a digital low, a few things happens. First thing that happen is the calculation of the elapsed time between magnet pulses, it does this by taking the current time and subtracting the time of the last detected pulse. Afterward it stores the current time in a shift register to be used for the next calculation. To prevent the elapsed time calculation to be executed more then ones a case structure is used, which switches to false after the calculation preventing future the operation to activate more then ones.

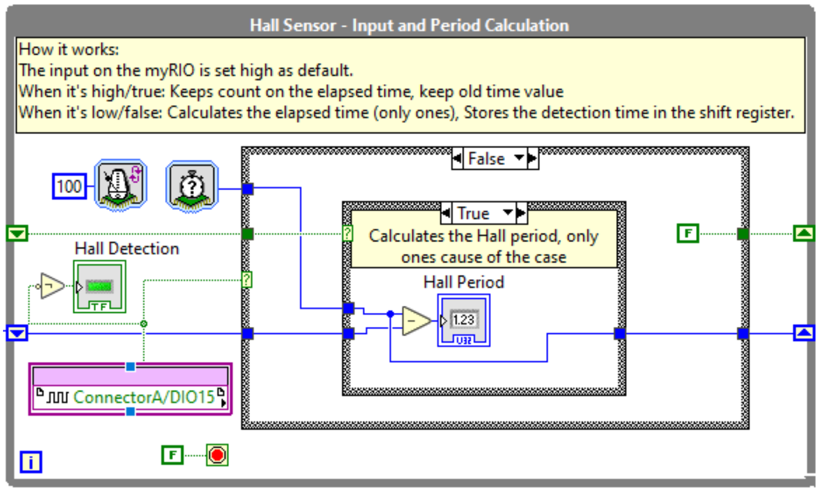


Figure 3: LabVIEW FPGA code for the Hall Sensor

6.2..3 Encoder

Author: Viktor Aronsson Karlsson

The Steering Encoder operates fully on the FPGA side of the myRIO. The encoder code utilises two digital inputs where both inputs reads boolean signals that are sent from the encoder unit to the myRIO. When processing the boolean signals with a chain of logical calculations, the rotational position of the steering motor can be calculated. With the help of the tick count both the velocity and acceleration can be calculated.

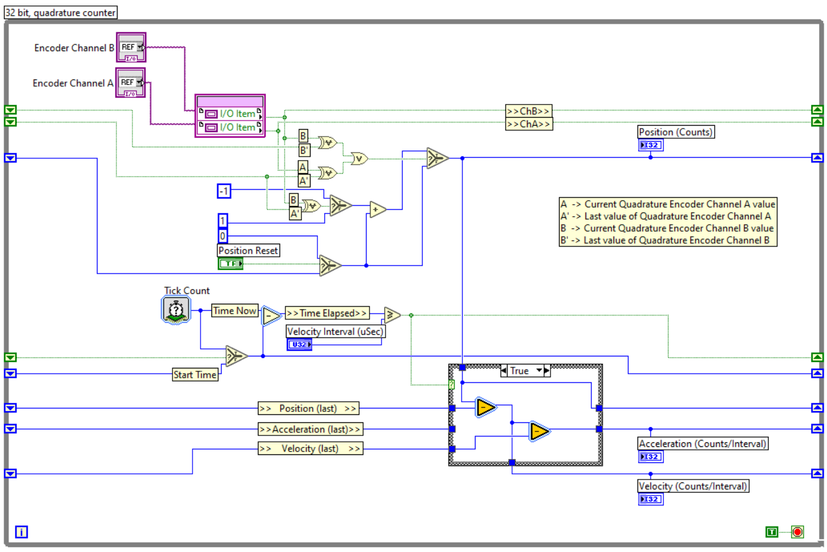


Figure 4: LabVIEW FPGA code for the Encode

6.2..4 E-Stop

Author: Viktor Aronsson Karlsson

The E-Stop reads the register on the FPGA that stores the value of the digital input from the Finder component, this signal can also be generated within the program with the help of a stop button. When the value of the signal becomes digital high, it will send a stop signal to all other functions, ending the run time of the system and stopping the LabVIEW code from running.

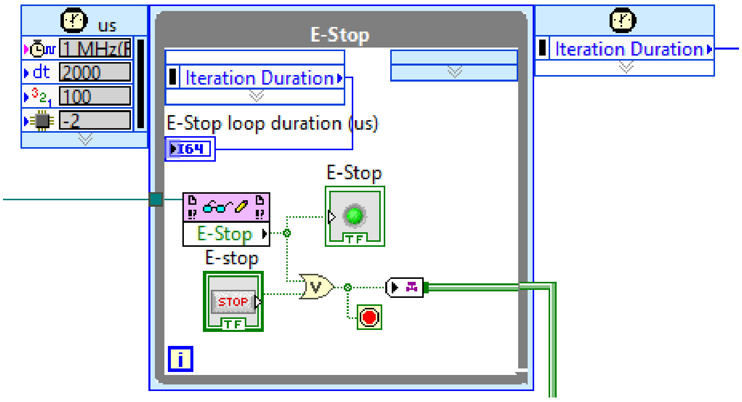


Figure 5: LabVIEW code for the E-Stop

6.2..5 GPS

Author: Viktor Aronsson Karlsson

The LabVIEW code for the GPS is simple. The code starts of with initialising the UART bus on the real-time side of the myRIO. All that is needed afterwards is to repeatably read the input buffer on a static interval and categorise the incoming data correctly.

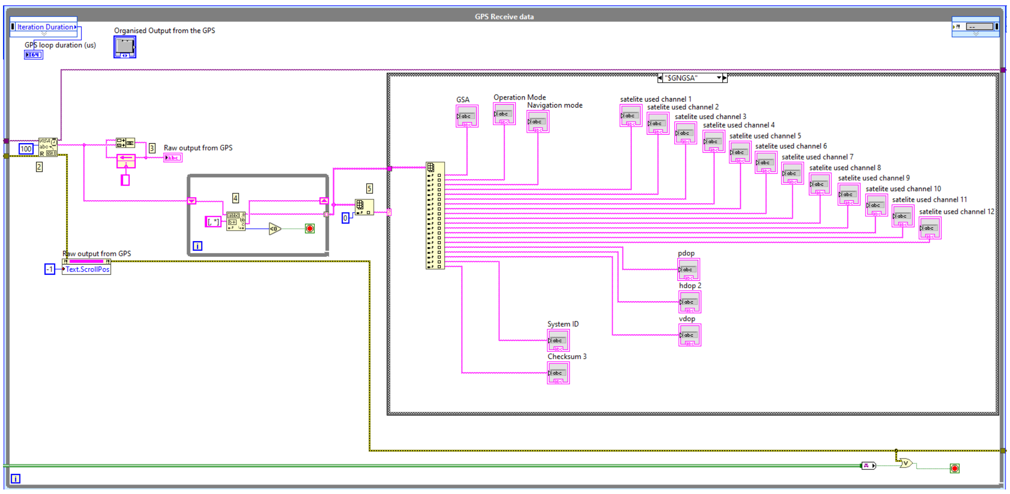


Figure 6: LabVIEW code for the GPS

6.2..6 Steering

Author: Viktor Aronsson Karlsson

The location of the control code for the Steering motor is on the Real-time side of the myRIO. The control functionality is controlled by having a enable pin that needs to be digital Low before starting the motor controller. When the program is ready to send signals to the motor controller, the pin will switch to a digital high and enable the ability to control the motor. The type of signal that is used for controlling both the direction and speed is a PWM signal, the neutral state of the controller is when the PWM signal has a duty cycle of 50%. One limitation that the motor controller has is related to the PWM signal, the limitation is that the duty cycle has to be within the margins of 10% to 90%.

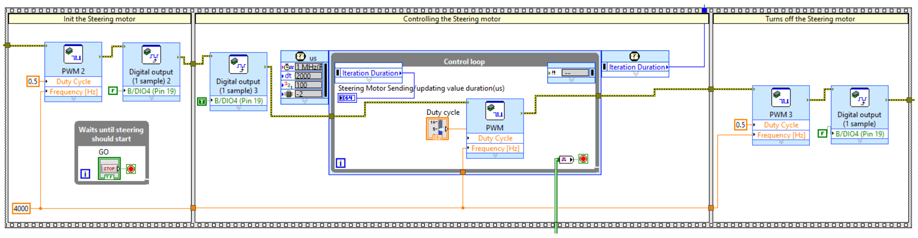


Figure 7: LabVIEW code for the Steering control

6.3. Control algorithms

Author: Viktor Aronsson Karlsson

There exist two control algorithms, one for the forward speed control and another for the balancing control. The algorithms are written in Python and divided in two main parts, an inner loop and an outer loop. The creation of the algorithms have already been made by an earlier iteration of the project and the objective for this iteration is the implementation of the algorithms in LabVIEW. The algorithms are meant to be placed in the LabVIEW architecture between the acquisition of data and distribution of commands, as can be seen in fig.1 represented by the block named Control Code.

6.3..1 Python Nodes

Author: Viktor Aronsson Karlsson

Python Node is a function in LabVIEW, which allows python code to be executed within LabVIEW. The nodes function by calling a specific function in the desired python code file. All nodes that are being utilised are connected within a session, which allows the nodes to share variables without going outside of the python code. The aspect of sharing resources between nodes simplifies the LabVIEW code by decreasing the amount of interaction between LabVIEW and Python. Even though with all the benefits that python nodes brings, it comes with a major flaw which is that lack of support for python nodes on Real-time and FPGA targets, this means that the current developed control algorithms can not be deployed on the myRIO or any National Instruments real-time platform. Possible solutions to the flaw is described in the appendix I.

6.3..2 Executing C code in LabVIEW

Author: Viktor Aronsson Karlsson

One of the options for the new implementation of the control code is to convert the code into C code which LabVIEW real-time can execute. To execute the C code in LabVIEW it first need to be converted into a .so file, how this is performed is described in appendix J , which then can be called with the function 'call library function'. Instead of recreating the control code by hand an automated way was examined. The tool that was examined was Cython[26] which have the functionality to translate python code into C code, the examination can be seen in Appendix K.

6.4. Communication within LabVIEW

Author: Viktor Aronsson Karlsson

The LabVIEW code utilises two methods for its internal communicating, one between the cores and the other between the functions. For the communication between the two cores a FPGA bit file is used, which have stored the FPGA configuration. The FPGA can be configured to store specific variables, which then can then be accessed by the real-time core via FPGA bit file. The communication that is used in between the functions are channel wires. Channel wires purpose is to relay data from the data gathering loops to the control loops, its also used to halt all operation when the E-stop is pressed.

6.5. Loggin of Data

Author: Viktor Aronsson Karlsson

The logging of the data values are stored in a TDMS file. The TDMS file format is a format created by NI with the goal of storing test data in a clearer format. The TDMS formats the data into groups and channels, where both contains properties to correctly identify the data. The desired function stores the data within the function loop using shift-registers to avoid undesired delays in the execution, when the program have finished the stored data will enter a new loop where it will be deposited into the TDMS file.

7. Requirements

Author: Love Bridén

All work done regarding the requirements of the project is described in detail in appendices O, P and Q.

The requirements management plan (appendix O) is what governs the whole requirement process. It contains information about the project in general, the stakeholders of the project and the main stakeholders for this specific iteration of the project. Further the requirements management plan contains information about the goals of the stakeholders of the project and how requirements were elicited from them. The requirements management plan also contains information about the structure of the requirements and the structure of the document hosting them. The requirements management plan also contains information about the method of the review process, how changes in the requirements are dealt with and how changes in the management plan are dealt with. Lastly, information about traceability among the requirements and how it is achieved is found in the requirements management plan.

The requirements specification document (appendix P) contains information about the stakeholder goals, the stakeholder requirements, the system requirements, the subsystem requirements and the component requirements that were elicited and controlled the development during this iteration of the Autobike project. The requirements specification document contains specific information about the work done in order to elicit the requirements, the models used for eliciting requirements, equations used to elicit the requirements and the requirements review process. The resulting requirements are presented in appendix Q.

8. Verification & Validation

Author: Maya Zawawi

The following section will include an appendix that has a full description of the verification and validation process, and how the tests have been conducted according to the standard V model to control if all improvements and modifications of the product(AutoBike) have been accomplished according to the stakeholder's requirements and specifications to fulfil the intended purpose. More information about the Verification & Validation plan and tasks have been described in the appendix "Verification and Validation Management Plan for project AutoBike 2021" See appendix R. Furthermore all tables that include test cases and information about the result of validation on unit, integration and system level will be found under the ninth section of the report with name Results. The main purpose of the V&V procedure is to check the latest modifications of the final product and to control if the specifications and requirements have been met , plus to highlight missing adjustments and improvements of the final product (AutoBike).

9. Safety

Author: Gabriel Sherif

As previously mentioned, an autonomous bicycle has been created in a collaboration between Mälardalen University and Chalmers University. In previous years of Autobike projects, the self-driving aspect of the project has been achieved. The purpose and goal of this project are to improve system reliability and increase performance by transferring the existing python code to a LabVIEW interface. The end product will be used in research and at Volvo Cars - in the automotive industry for self-driving cars, the autonomous bike will be used to validate crash prevention algorithms.
The AutoBike project has the self-driving aspect that creates the need for the end product to have a safety evaluation aspect. The self-driving part immediately says that there will be no driver for AutoBike. This means that controllability and controllability will be Autonomous. In this case, Autonomous implies that the system must control itself and make decisions in certain situations, other objects, and with the environment. With these aspects, safety must be taken into account - this AutoBike will be used to test new self-driving cars. There are also components and critical parts that require a safety development process. Safety The development process is necessary to ensure that the system is more reliable and minimize the danger to the project staff involved. Even from the component side, it should be sufficiently secure in service life and not damage any way in the system.
In connection with the project, a safety manager is needed for general safety analysis of system safety- the Safety manager was given responsibility for writing all necessary Safety -related requirements that would affect the project and was also given responsibility for ensuring that Safety -related requirements have been complied with.
Writing safety-related requirements requires a fundamental basis for guidelines for this project, a set of concepts and steps are needed that must be followed, and correct standards adopted. The system safety manager developed these primary bases for this project and found them in the system safety plan (appendix S). With the help of others in the AutoBike project, the safety manager takes into account hazards or things that can go wrong to ensure creating a reliable system for AutoBike that can be used for various assignments in the research and automotive industry.
This safety process expects to achieve some basics - a set hazard (appendix T) in the form of a table that illustrates the potential hazards that may arise in a particular situation during the project. A list of the safety chief proposed safety measures to control the high-level risk. And check for opportunities to implement these safety measures. A safety requirement with general proposals for all requirements from the safety evaluation process. (appendix U)

The figure below generally shows how the system is structured regarding the safety analysis. All descriptions can be found in the appendix T. **Note.** The Risk Analysis Schematic is adapted to the projects.

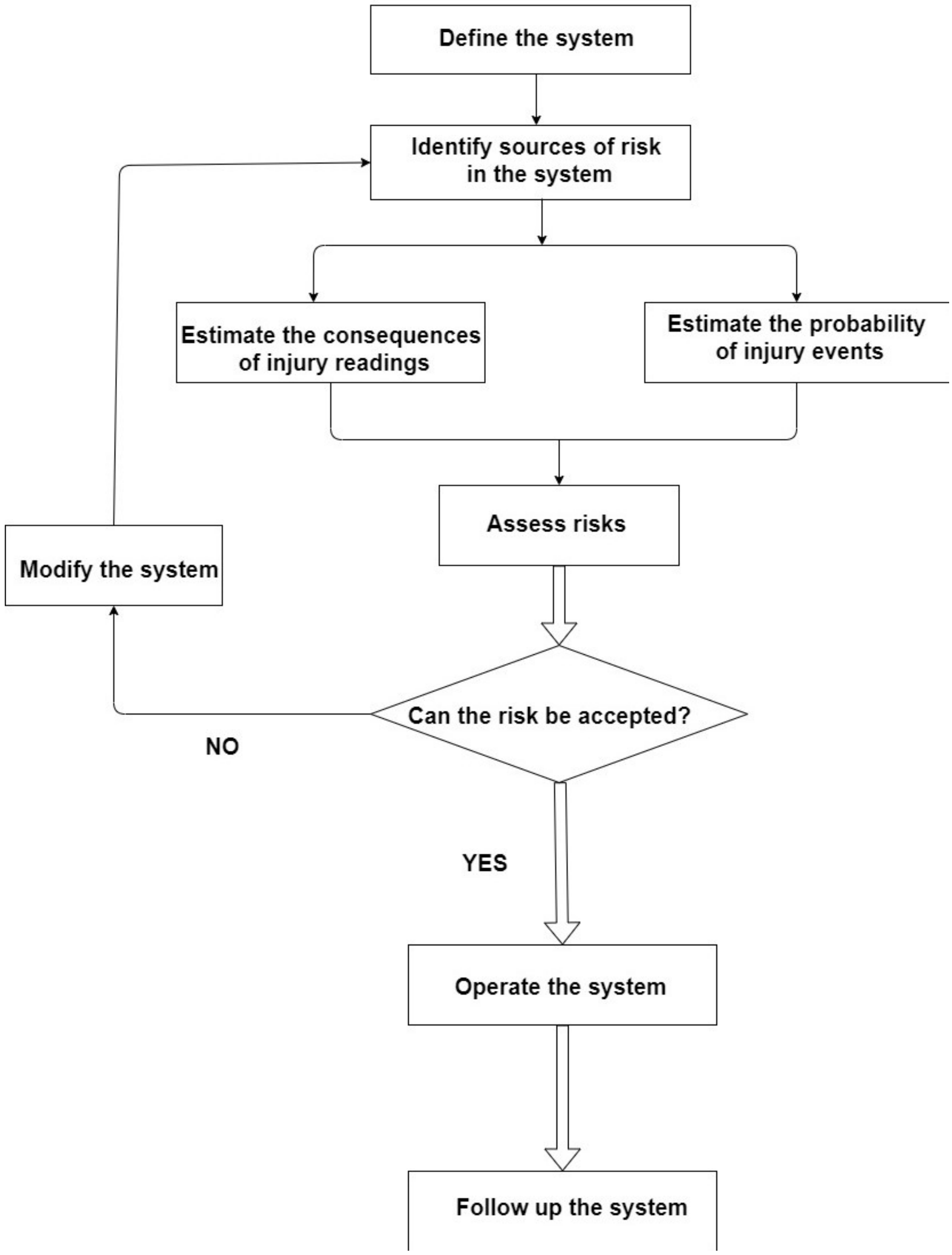


Figure 8: Risk Analysis Schematic

10. Results

Author: Maya Zawawi

In this section an illustration of the results will be explained, which can show that the system became more dependable and reliable by make the data received from the sensors more accurate and the calculations faster. As well the hardware received from Chalmers has been modified, additionally a new software has been created in LabVIEW real-Time which allow the system to be more reliable and perform faster than the old one, which met the stakeholder's goals. Plus all tables that conducted due to V-Model, and safety analyse are available under this section.

10.1. The execution time

Author: Maya Zawawi

The following table shows the result of running the program for about 5 minutes, plus the number of samples that have been conducted under the testing time, where the code has logged all the data timing, for each loop,then the periods were averaged.

Component	NO.SAMPLES	AVERAGE	WORST CASE
IMU	77.5K	1137	1300
HALL	103K	210	250
ENCODER	103K	111	200
GPS	1K	17	30
E-STOP	109K	112	150
STEERING	104K	208	250

A clearer view about the new software performance can be shown in the figure 10 which reflects the results of the previous table graphically. The results show a faster execution time. and more reliable sensor data collection. The code could collect data during a maximum time equal to 5 ms and approximately 2ms for execution.

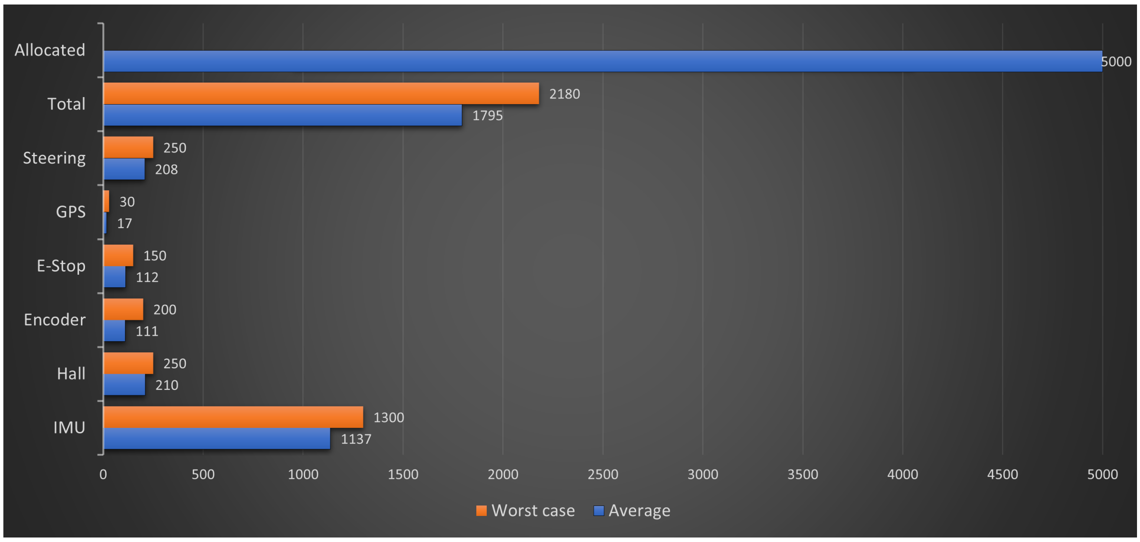


Figure 9: The execution time for the Lab VIEW code

10.2. Accuracy of Sensors data

Author: Maya Zawawi

The following figures illustrate the accuracy of the sensors, which show that the noise received from the sensors is very small about 0.1-0.2, and acceptable, which means that the accuracy of the system is convince.

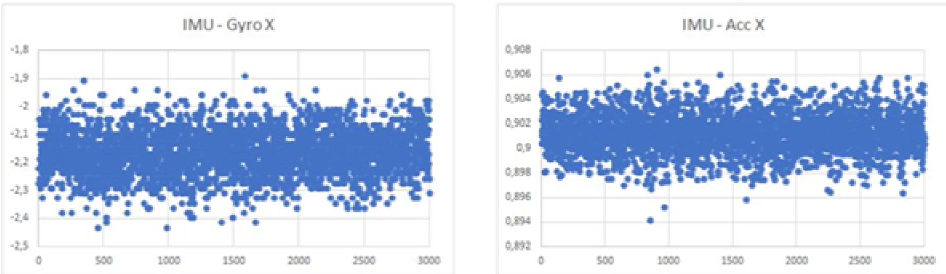


Figure 10: The accuracy of IMU in x-axis

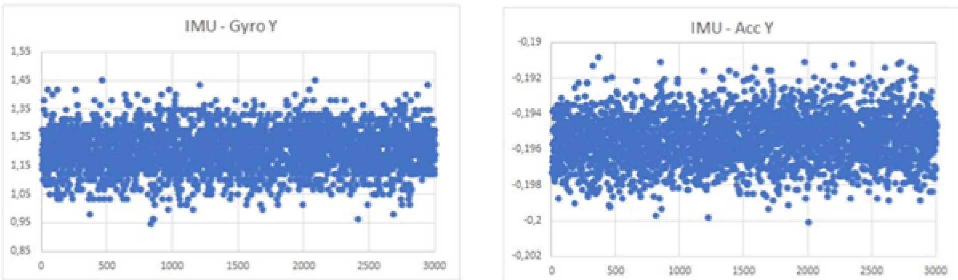


Figure 11: The accuracy of IMU in y-axis

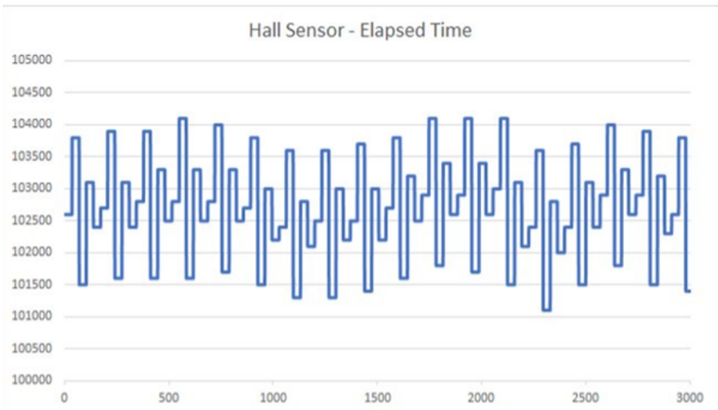


Figure 12: The Hall sensor- Elapsed time

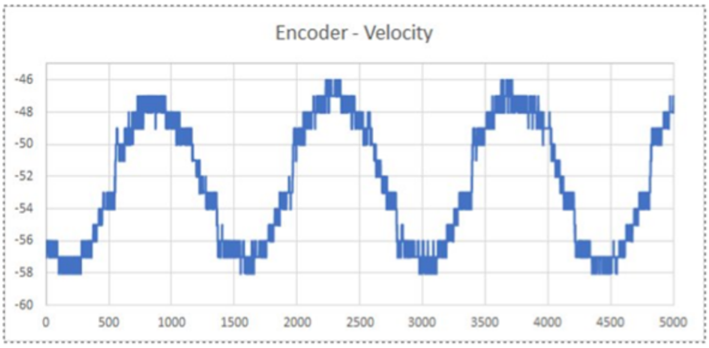


Figure 13: The Encoder velocity graph

10.3. Hardware

Author: Love Bridén

The hardware of the end product used the same box received from Chalmers university because the battery could not fit in the smaller box that was received from Chalmers. Some components present in the old version of the Autobike (see figure 14) are reused, e.g. the steering motor controller and the battery are reused. Some components used in the old version of the Autobike are swapped for other components, e.g. the beagle bone black board is swapped for the MyRIO device and the old forward motor controller is swapped for the VESC as the main stakeholder requested. The components used in the new version of the Autobike are presented in section 5.. The new hardware in the so called "electronics box" is pictured in figure 15.

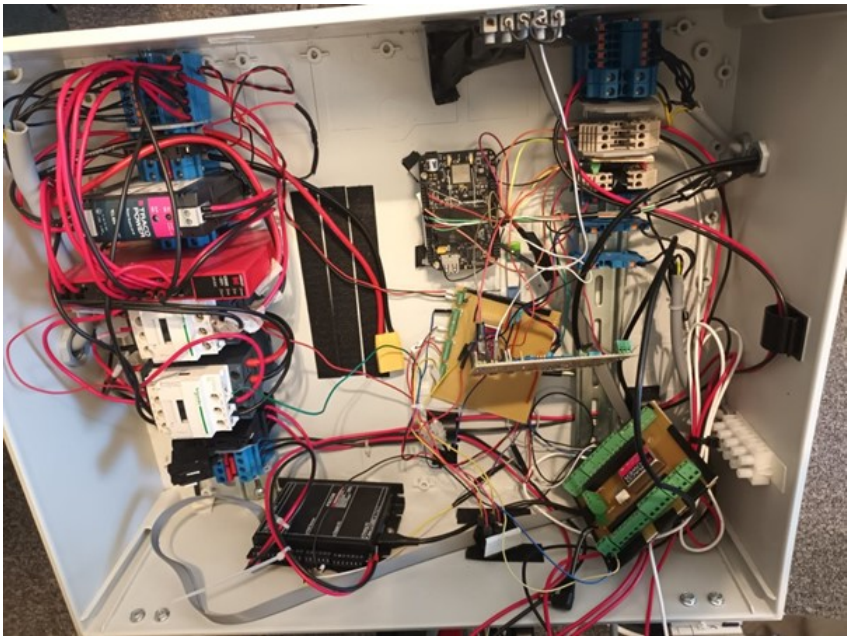


Figure 14: The hardware in the "electronics box" before the modifications. The battery is excluded from this picture.

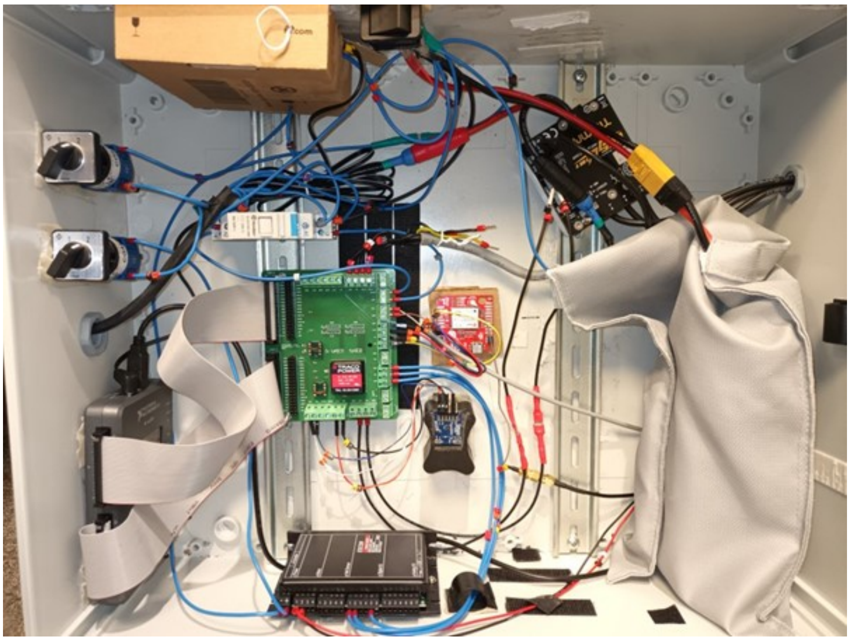


Figure 15: The hardware in the "electronics box" after the modifications.

A comparison between figure 14 and 15 shows that a reduction in the number of components in the electronics box has been achieved. The total size of the components has also been reduced. The updated version of the Autobike also has less wiring in the electronics box compared to previously. In figure 15 on the green circuit board a DC-DC converter is located, this is the DC-DC converter that was accidentally damaged during this iteration of the project (see appendix E). Instead of swapping the DC-DC converter on the PCB (green circuit board) a Power Distribution Board (PDB) was designed, the new PDB is located in the cardboard box in the top of figure 15. Information about the PDB is presented in appendix D. The schematic of the PDB is presented in figure 16 and the layout is presented in figure 17. The idea is that the PDB converts the voltage of the battery to a voltage level acceptable to power the MyRIO that in turn will power the rest of the low power consuming peripherals of the Autobike. Included in the PDB are robustness features such as Transient Voltage Suppression (TVS) by using TVS diodes (D3 and D4), reverse polarity protection by using diodes (D1 and D2), overcurrent protection by using a fuse (F1). There also exists capacitors (C1, C2, C3 and C4) and an inductor (L1) that are used to reduce the noise of the system.

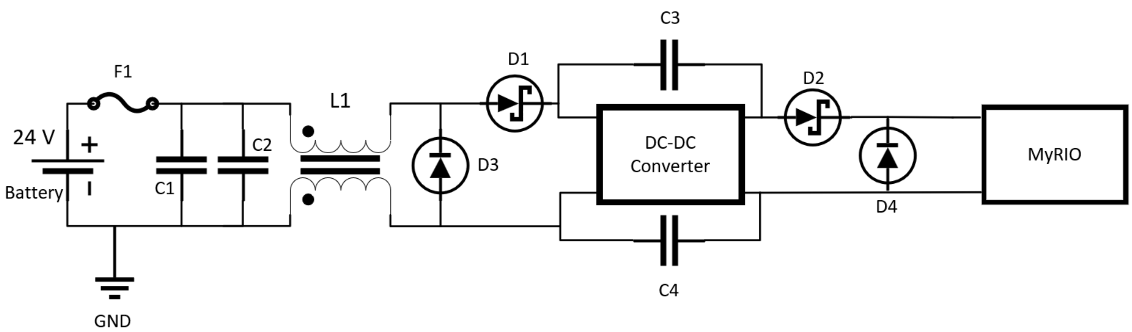


Figure 16: Schematic of the new PDB. The battery, ground and the MyRIO device are connected to the terminal of the circuit board, thus they are not located on the board. This figure is presented in appendix D.

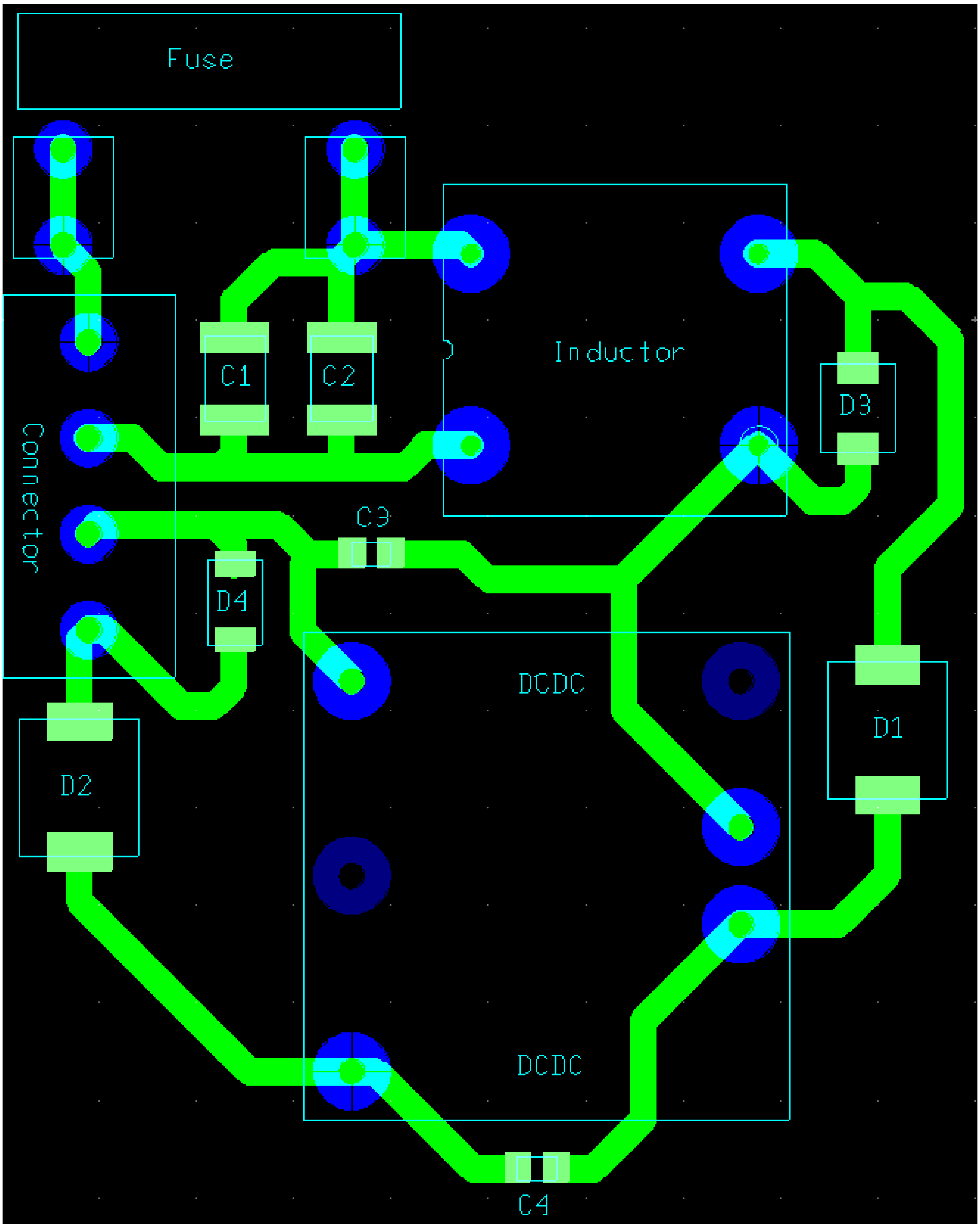


Figure 17: Component and trace layout for new PDB. The fuse is mounted on the fuse connectors, in the layout the fuse component is placed besides the fuse connectors for illustration purposes. The light blue squares and texts represents the components of the PDB. The darker blue rings represents drill holes and pads for through hole mounted components. The green represents different nets consisting of traces and pads. This figure is presented in appendix D.

In appendix C information about the so called finder circuit is presented. The schematic of the finder circuit is presented in figure 18, the schematic is part of the architecture presented in figure 19.

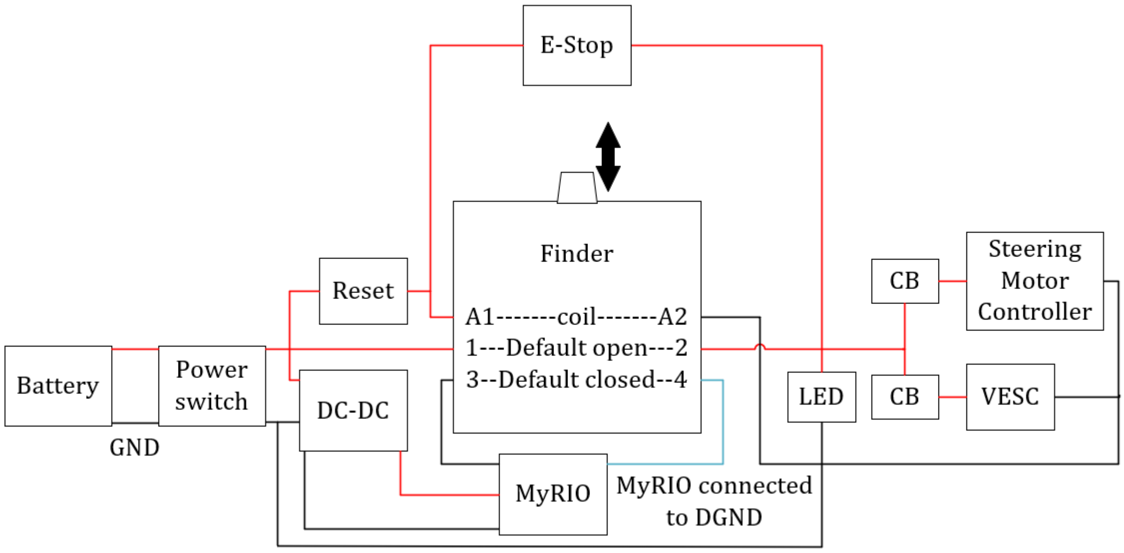


Figure 18: Schematic of "Finder circuit" which is part of the architecture of the Autobike. This picture is presented in appendix C.

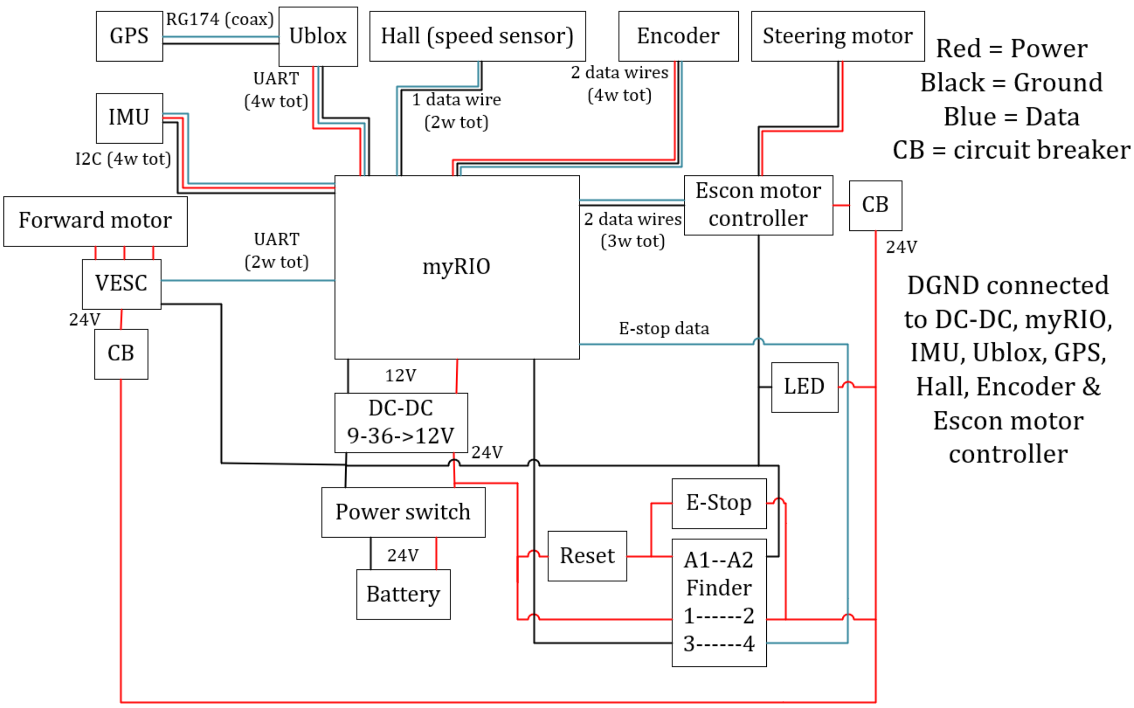


Figure 19: Schematic of the electrical/electronical architecture of the Autobike after the modifications. This figure is presented in appendices A and .

The components of the architecture that are located outside of the electronics box are the GPS antenna, located on top of the box, the Hall speed sensor which is located at the rear wheel, the encoder which is located on the steering motor, the steering motor which is located on the handlebars of the Autobike, the LED, E-stop button and reset button that are located on top of the electronics box and lastly the forward motor located by the petals of the Autobike. Appendix A describes the interfaces between the human using the Autobike and the hardware of the Autobike, the information presented is meant to aid a user in using the Autobike. All components of the system are powered by the battery. The MyRIO device collects data from the sensors of the Autobike and it sends actuator data to the actuators of the Autobike. The finder circuit is responsible for making the reset, LED and E-stop functionality to function properly. In addition a document describing the setup of the hardware was produced (see appendix B) to aid any user to setup the system, locate faults in the case of failures and to easier modify the software of the Autobike.

10.4. LabVIEW

Author: Viktor Aronsson Karlsson

The part of the LabVIEW architecture that was the focus of this project was the core components, the handling of external inputs and outputs. The fully finished area of the architecture that the construction was all the data gathering components, which includes the IMU, Hall sensor, Encoder, E-stop and GPS. The control for the steering motor was constructed, while the control for the forward motor was not constructed. The Python control algorithms were concluded to be unusable in their current state. Alternative solutions were brought forth as can be seen in Appendix I, where the most promising solutions for the problem were the reconstruction of the control code in C code instead of Python, instructions on how to run C code in LabVIEW is described in appendix J.

10.5. Requirements

Author: Love Bridén

The results from the work done related to the requirements are the requirements presented in appendix Q.

10.6. V&V Results

Author: Maya Zawawi

All tables of conducted Verification & Validation on unit, integration and system level are available in the following figures (8,9,10,11), where the initial requirements of the project and stakeholder's needs have been verified and validated:

Id	Test Scenario	Test Case	Pre-condition	Expected Result	Actual Result	Pass/Fail	Author
un001	Measure the voltage difference of hall sensor	MYRIO	sensor near the magnets that placed on the rim and spin the wheel of the bike	Change in the voltage value	sor react to the	pass	MZ
un002	verification the E-stop digital signal to the MyRIO	Multimeter	Connecting MyRIO to the PCB & PDB boards with 24v power supply	Push the reset button will allow the finder to forward the current through its coil to the motor	The current flowed correctrly throug the finder	Pass	MZ

Figure 20: The table of Unit Verification & Validation

Id	Test Scenario	Test Case	Pre-condition	Expected Result	Actual Result	Pass/Fail	Author
int001	Implement pthyon node in LabVIEW	LabVIEW	Insatll python code from Chalmers into LabVIEW	Using Python nodes to controll the LabVIEW code	The labview could not identify the python code	Fail	MZ
int002	Test the new PCB	Oscilloscope	Testing the power distribution for the MyRIO	Motor rotation	Motor rotated	Pass	MZ
int003	Convert python code into C code	Jupyter	Install Cython to convert python into C code	Run C code into LabVIEW	Cython could not read Python could from Chalmers	Fail	MZ
int004	Trasfer Python C code into LabVIEW	LabVIEW	-	-	The Cython changes the functions names	Fail	MZ
int005	Test if LabVIEW code can controll the motor	LabVIEW	-	The LabVIEW code will control the rotation of the motor	The LabVIEW could controll the motor´s rotation velocity due to the duty cycle	Pass	MZ
int006	Test if LabVIEW code can controll the IMU	LabVIEW	-	The IMU will be sensitive to any movement	The graphs of IMU Gyro and IMU acceleration are sensitive to any movement	Pass	MZ

Figure 21: The table of integration Verification & Validation

Id	Test Scenario	Test Case	Pre-condition	Expected Result	Actual Result	Pass/Fail	Author
sys001	The hardware architecture will minimize and has a better organization and understanding.	Review	Access to one of the designing environment will be available	An hard copy of the new architecture will be available	New architecture of the hardware is achieved	Pass	MZ
sys002	A manual of the hardware will be produced	-	-	Manual will be produced at the end of project	Tables of new hardware has been produced	Pass	MZ
sys003	Data received from all sensors (excluding GPS) with frequency 100 HZ	The LabVIEW environment will be used to measure the sampling	BeagleBone has been swapped to MyRIO	The sampling frequency shall remain 100HZ after using MyRIO	100 HZ	Pass	MZ
sys004	Data received from all sensors within 10ms	The LabVIEW environment will be used to measure the sampling time	BeagleBone has been swapped to MyRIO	The sampling frequency shall remain 100HZ after using MyRIO	7ms	Pass	MZ
sys005	Data send to the actuators with frequency 100 HZ	The LabVIEW environment will be used to measure the frequency	BeagleBone has been swapped to MyRIO	The sampling frequency shall remain 100HZ after using MyRIO	100HZ	Pass	MZ
sys006	Data send to actuators within 10ms	The LabVIEW environment will be used to measure the sending time	BeagleBone has been swapped to MyRIO	The sampling frequency shall remain 100HZ after using MyRIO	7ms	Pass	MZ

Figure 22: The table of system Verification & Validation

sys007	Calculating sampling/sending data using MyRIO	LabVIEW	BeagleBone has been swapped to MyRIO	The calculation shall be conduct within 10ms	All calculations conducted under 10ms using LAbVIEW-Rio	Pass	MZ
sys008	The size of the electronics box shall be reduce	Comparsion between the old and the new	-	Smaller box	The battery can not fit into the smaller box	Fail	MZ
sys009	Reduce the distribution of the electronic components	Comparsion between the old and the new distribution	-	less wires	Alot less wires	Pass	MZ
sys010	The autobike shall power the forward motor using a VESC 6Mkv motor	Function generator	Install new motor controller	Control the the forward motor	Success to control the motor	Pass	MZ
sys011	The autobike will receive GPS signal with frequency oh 10 Hz	LabVIEW	BeagleBone has been swapped to MyRIO	Signal receiving from GPS is 10Hz	-	Pass	MZ
sys012	The autobike shall fulfill its complete functionality using less wiring	Comparsion between the old and the new wiring	-	Less amount of wiring	-	Pass	MZ

Figure 23: The table of system Verification & Validation- Continuous

10.7. Safety

Author: Gabriel Sherif

The Safety process consists of two parts (appendix S): the analysis part and the second control part.

Analysis part

- System description
- Hazard identification
- Hazard analysis
- Risk analysis

Each of the above steps has its description contained in SSMP (appendix S). All steps have been adapted to this project, taking into account various causal factors.
As described in SSMP, the Safety process consists of two different parts. Consequently, it divides the safety results into two parts: the analysis and effects, probability, severity, and Hazard Risk Index (HRI). This was done for all three parts of the system, before and under, and after operation. And has divided each risk level for itself (appendix T). To be clear and avoid unnecessary confusion, each row in the table (Refer to Hazard analysis) belongs in the same hazard. Each column in the table shows the results of a specific step in the safety process. And to be even clearer, each risk level has its table.
After the safety analysis process, a set of safety requirements was derived with general proposals for all safety requirements, all requirements have been reviewed and approved. (appendix U)

10.7..1 Implementation Results from safety

Author: Gabriel Sherif

The high-level risks were extracted from the previous tables (table below), i.e., level 3. Level3 are the hazards that are particularly dangerous and need safety measures. And after discussion and review with the management, project manager, safety measures, it was decided (appendix T) to implement measures - by buying a new battery. All hazards that ended up in Level 3 were connected to the battery with the basic factor that the battery is too old, and there is a risk of overcharging "battery fire or explosion" (appendix T)

ID	**Hazard**	**Causes**	**Effect**	**Probability**	**Severity**	**HRI**	**Recommendation**	**Effect of recommendation**
5.0	Loss of power	The battery discharges	Missing the power and the bike cannot move	Remote	Critical	D2	Voltage regulations needs	It will change the HRI from D2 to D3 (Remote, Minor)
5.1	Battery start fire	The battery is too old	Missing the power and the bike cannot move	Remote	Critical	D2	Change the battery to new one	It will change the HRI from D2 to D3 (Remote, Minor)
5.2	Battery start explosion	The battery has damages	Missing the power and the bike cannot move	Remote	Critical	D2	Change the battery to new one	It will change the HRI from D2 to D3 (Remote, Minor)

11. Discussion

Author: Gabriel Sherif

The AutoBike project has involved several challenges with limitations around the project area. In the beginning, the projects were not well defined, which means there was some uncertainty about what would be done. The projects revolved around autonomous and reliability, which the project members had experienced earlier. Which made it more encouraging to plan and discuss. The results showed that the project team has succeeded well in addressing these challenges that have been described in the section's methods. Everyone in the group has learned more about project constraints and project scheduling work. Last but not least, it was instructive and challenging to work on a larger project with few group members.

11.1. Hardware

Author: Love Bridén

The new hardware features less wiring compared to previous iterations. This will theoretically reduce the level of noise in the system. This is because the strength of an electric field is inversely proportional to the square of the distance between the source of the electric field and the conductor in the field. Having less wiring means that there, generally, is a longer distance between emitters and susceptible objects. The average distance between the components in the electronic box is larger compared to previous iterations of the project, this would theoretically reduce the amount of noise in the system. Also the length of the average wire is shortened. This will also theoretically reduce the amount of noise in the system. The new noise levels were not measured, hence no definite answer to those problems can be provided. This is an activity for future work.

The creation of a new PDB with proper filtering and robustness features is also a way to reduce the noise of the system, for future work this has to be measured. Also, the robustness features of the PDB shall be tested in the future.

Concerning heating issues the components in the electronics box were placed with quite a distance between them. This would theoretically increase the ability of the surrounding air to transport heat away from the components. This has to be tested in the future. The Autobike is meant to be used for short periods of time, hence, heat buildup in the system should not be a big issue.

Some of the components are rated for quite small currents for an electronic vehicle (e.g. the finder and the circuit breakers). The currents drawn at operation should be measured as a future work. The measurements should be used to guide any design choices in the future to determine if any components should be swapped for components rated for larger currents. Also, the wiring of the Autobike is quite thin. This is because some of the components had small terminals that required small enough bootlace ferrules to be used and in turn, thin enough cables for those ferrules had to be used. If the current measurements indicate that large currents flow through the system, these cables should be swapped for thicker cables. To allow this, some components may have to change, or at least the terminals of the components has to change.

The stakeholder requested that the forward motor controller should be examined to evaluate if it could send speed data to the microcontroller of the Autobike. The forward motor used does not provide position data that could be translated into speed data by the forward motor controller. Instead, as a future work, after control of the forward motor controller has been successfully implemented perhaps some sort of regression or plotting could be used to determine the speed of the inputs to the forward motor controller. With this, an approximation of the speed could be calculated and if successfully validated the speed sensor could be removed reducing the complexity of the Autobike further.

The Teltonika described in the hardware section of this report was not implemented as it was deemed a low priority task and due to the lack of resources. This is left for future work.

The new battery purchased during this iteration of the project features a battery management system. Using this battery would increase the safety of the Autobike significantly as unbalanced cells are a major factor for LiPo battery failures. In order to use it, the connectors of the battery has to be changed into a female XT90 connector.

The forward motor controller features an internal IMU. As a future work (after the forward motor controller has been successfully implemented software wise) the accuracy of this sensor should be evaluated. If this sensor is deemed accurate enough for the purpose of the Autobike perhaps the IMU currently used could be removed further reducing the complexity of the Autobike.

Lastly, hardware wise, the placement of the IMU should be evaluated as a future work. If it, for some reason, has to change an optimal location should be identified. As of now, the IMU is located in the middle of electronics box.

Author: Maya Zawawi

As future work, The hardware needs to be implemented in a smaller box, with good isolation to be protected from external environmental factors, the box needs to be shock resistant in case of any collision under the outdoor test.

11.2. Software

Author: Viktor Aronsson Karlsson & Maya Zawawi

The software implementation of the Autobike was fully recreated from scratch in LabVIEW, next to none of the pre-existing LabVIEW code that was provided at the beginning of the project could be used. For all the external interfaces to the sensors and actuators, an examination of their data sheets and varies online resources were needed to be performed in order to understand how to communicate and implement them in an LabVIEW environment. Resources with valuable information was far and few between, the resources that existed were often, either an old implementation that doesn't work anymore or the implementation was for another environment, meaning the implementation could be completely different e.g. implemented in C++ and using its library functions and not describing how they work.

The control algorithm code that was given to us from Chalmers, was given in the format of python code. Though the problem with the current implementation is that the control code is scattered in multiple parts of the code and the code spans multiple files, the number of files that were provided was over 200+. Not all files were used, some had not been used for years, but the problem was to locate the important files when there was no documentation within or outside the code about what function the code file served. The Python code received from Chalmers University will be rewritten in C language in order to be used in LabVIEW itself and to enable full control from LabVIEW code, this however was left for future work.

The code was constructed piece by piece, where each component was implemented one at a time. When the implementation for the component was finalised, it was combined with the other components that were already finished for the purpose of knowing that they function together. All components was implemented except the UART communication for the forward motor controller, the command used for controlling the forward motor controller were not found, so the implementation for the controller was left for future work. The creation of more Built-in-tests, to be conducted after changing the Python code will help to validate the reliability and dependability of the system much more.

11.3. V&V

Author: Maya Zawawi

With the aim to achieve the optimal performance of both software and hardware of the AutoBike, the verification and validation procedure activities needed to be conducted, with the satisfaction of standard requirements that have been created by the requirements engineers of the team. A series of tasks and test cases have been performed in MDU facilities to complete the V&V activities, which was a more feasible method to control the quality of achievement of all improvements at all project levels and attributes. moreover to analyse the weak points that needed to be covered in future phases of the project for example to use C code instead of Python code to be able to install the code nodes in LabVIEW. In another word, the V&V that has been conducted gave a reference to the next development process.

11.4. Requirements

Author: Love Bridén

Regarding the requirements, the work had to be rushed as there was a lack of resources throughout the project. The plan was to complete the requirements at an early stage for the requirements to guide all development during this iteration of the course. Instead, the requirements were not completed when the design choices took place, this resulted in that the work with the requirements had to be done informally while developing to guarantee that the design choices fit the goals of the project. At a later stage the work with the requirements were done formally. This, to say the least, is not an optimal workflow for the project.

11.5. Safety

Author: Gabriel Sherif

As previously mentioned, an autonomous cycle has been created in a collaboration between Mälardalen University and Chalmers University. In previous years with Autobike projects, the self-driving aspect of the project has been achieved. The purpose and goal of this project are to improve system reliability and increase performance by transferring the existing python code to a LabVIEW interface. The end product will be used in research and at Volvo Cars - in the automotive industry for self-driving cars, the autonomous bike will be used to validate crash prevention algorithms. Concerning the results that took care of level3 with the high-risk level - then we have achieved the desired safety results process, but with a specific part. These level 3 had the highest risk level - which we have tried to minimise the risk level by implementing risk measures.
When it comes to Safety in a system, you can not say that it is only a part - it is a whole system. We have reduced the risk level by implementing risk measures; it is a specific subsystem, not the entire system. Then, we can not say that the whole system is safe and reliable enough. Each risk must be controlled and remedied if necessary. When it comes to Safety , we can say that the system is more secure but not 100 percent. Dangers that ended up in Level 2, such as a bicycle colliding, fall to the ground. Theoretically, we must take care of it to avoid and minimise the level of risk. But because this project is a big project that leads to future researchers and the automotive industry will benefit from the bike. Then it is this kind of danger that ends up in level 2 is accepted danger and can happen to someone in the meantime. But to be more secure and reliable enough, one can take measures that can lead to risk reduction - these can be a potential implementation of action Safety . Within the framework of this project, it is accepted that the bicycle falls to the ground, national level at level 2. All levels have their description (appendix S). Every danger that ends up in Level 2 has an accepted effect and was not considered threatening enough to lead to something serious. But, of course, all risks must be addressed for the system to be safe - the probability and degree of difficulty must be carefully examined because it affects the effect. If the effect is not dangerous, then the system is more secure.

Last but not least, all components have a built-in safety system that protects against specific hazards and reduces the risk of them occurring.

11.6. Planning and Organising

Author: Viktor Aronsson Karlsson

At the start of the project the main goal was to implement Chalmers software solution in the new hardware components, with the main part of the project being dedicated to analysing the Autobike, as can be seen in the appendices M and N, which where both created at the beginning of the project. Parts of the scope, such as the parts about validation of old solutions and compassion between the difference version of the Autobike, had to be drooped due to the given hardware, software and documentation where lacking, which lead to the need of reexamination of the systems. Due to the lack of personnel in the project, the role of Quality Manager was decided to not be assigned and the roles responsibility was left unfulfilled, though before this decision was made a quality management plan was created but no one was there to maintain it, the plan can be seen in appendix V.

11.7. Implications with the roles

Author: Love Bridén

The roles I received were hardware manager and requirements manager. The work with requirements, usually, is conducted at an early stage of a project as the results from those activities dictate the design choices throughout a project. This means that the work with requirements has a high priority to be conducted at an early stage. Regarding the hardware, the pandemic currently raging in the world has led to a shortage in components and especially semiconductors. This decreases the availability of components at online vendors and it increases transportation times. To tackle this the hardware architecture should be completed as early as possible to still have time to manufacture, setup and test the resulting system, especially in a project like this iteration of the Autobike project where a limited amount of time is given. This means that I received two roles with high priority during the beginning of the project. I decided to focus on the hardware since those results were of higher interest to the main stakeholder. This resulted in a workflow that was not optimal as described in subsection 11.4..

Author: Maya Zawawi

In the AutoBike project my roles were a member of the software team, and a verification & validation manager, where I prepared the V&V procedure. The main tasks were to verify and validate the software performance, review the requirements, control the execution and tests, and finally analysing the results. As a member of the software team I participated in creating the architecture of LabVIEW code, and took four online courses about the Real-time and LabVIEW, moreover I did a research about the Cython and how to solve the problem of implementation Python nodded in LabVIEW code and find other alternatives.

Author: Viktor Aronsson Karlsson

In the project Autobike, the main role that I performed was that of Project Manager, my secondary role was that of Software Developer. The role of Quality Manager was also assigned to me but due to the already massive workload, the role of quality manager was discontinued. The main tasks were to organise the group and the project, making sure everything worked as it should and the resources that were needed were available. In the latter half of the project, the tasks of software development started to become more prominent and needed to take a higher priority than project management, this was not a major problem since the main workload for a project manager is at the beginning and end of a project.

Author: Gabriel Sherif

The roles I received were hardware member and system safety manager. For a system to be safe, it is important to detect and analyse all hazards - to see how each hazard will affect the system. Therefore a System Safety Management Plan (SSMP) (appendix S)and then a risk analysis(appendix T). As Safety Manager, I was responsible for creating all system Safety documents. As a Hardware member, I had the role of analysing Hardware for related tasks.

12. Conclusions

Author: Love Bridén

During this iteration of the Autobike project some of the goals were reached and some were not due to different reasons. The stakeholder requested that some hardware components should be replaced by other hardware components providing the same functionalities. This goal was partly met. The forward motor controller was swapped and integrated hardware wise. However the communication between the microcontroller and the forward motor controller was not completed due to lack of resources. The microcontroller was also requested to be swapped by the stakeholder. This goal, hardware wise, was completed. Although software wise the goal was not met due to the crucial problem presented in appendix I, this problem was not tackled due to a lack of resources during this iteration of the project. Although, possible solutions were proposed. The goal regarding the placement of hardware components was not met as requested by the stakeholder. This is because the battery was too large to fit in the smaller boxes received, instead the old, larger, box was used. The placement of components inside the box was done with respect to EMC and heat issues, although a lack of resources required these design choices to be done in a haste resulting in a solution that most likely is not optimal. Regarding the goals related to software of the Autobike they were partly met. The LabVIEW-Python wrap-around requested was not reached due to the problems presented earlier in this section. The design of BIT:s were also not entirely completed, although, software components created can easily be reused in a BIT. The goal related to the sampling-calculating-sending frequency was met. Data about the accuracy of the sensors of the Autobike is presented in the results of this report, in other words, this goal was met. Likewise were the goals related to requirements, safety and verification and validation.

Author: Gabriel Sherif

When it comes to safety, it seems that the security of a system will never be 100 percent. But one can reduce the risks of occurring within measures. If some risks have not been addressed at a level that would need action, then that does not mean they exist. It may be that human factors have influenced these assessments or are due to a mutual understanding. But results on projects have shown that it could make the system reliable and dependable. As we have mentioned in the discussion section, the bike falls to the ground, and by implementing safety measures, you can avoid this incident - it may be future work.

References

[1] National Society for Road Safety, *Cyklister*. [Online]. Available: https://ntf.se/ntf-anser/cyklister/.

[2] H. Baaz, D. Bengtsson and M. Östgren, 'Project autobike autonomous bicycle platform,' Unpublished, 2019.

[3] C.-H. Chi and J.-J. Chou, 'Riderless bicycle with gyroscopic balancer controlled by fsmc and afsmc,' in *2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, IEEE, 2015, pp. 150–157.

[4] A. Siwerstam and N. Jeansson, 'Design of mechanism of self-driving bikes,' Unpublished, 2021.

[5] N. J. Bahr, *system Safety Engineering and Risk Assessment*, T. bibinitperiod F. Group, Ed. Lund : Studentlitteratur, 2014.

[6] National Instruments, *Myrio-1900 user guide and specifications*, 2013. [Online]. Available: https://www.ni.com/pdf/manuals/376047c.pdf.

[7] Digilent, *Pmod nav reference manual*, 2017. [Online]. Available: https://docs.rs-online.com/8dd8/0900766b815970f5.pdf.

[8] AAMP Global, $Gps_wp/trkseries - gosantenna$, Year NA. [Online]. Available: https://docs.rs-online.com/e3b5/0900766b80d1ee46.pdf.

[9] u-blox, *Zed-f9p*, 2018. [Online]. Available: https://www.mouser.se/pdfDocs/ZED-F9P_DataSheet_UBX-17051259.pdf.

[10] Teltonika Networks, *Rut950*, 2021. [Online]. Available: https://teltonika-networks.com/downloads/en/rut950/RUT950-Datasheet.pdf.

[11] Assemtech Europe, *Mmpsa 240/100*, 2008. [Online]. Available: https://www.farnell.com/datasheets/1673516.pdf.

[12] Avago technologies, *Hedm-55xx/560x & heds-55xx/56xx*, 2012. [Online]. Available: http://www.avagotech.com/docs/AV02-1046EN.

[13] Maxon motor, *Escon 50/5 servo controller p/n 409510 hardware reference*, 2015. [Online]. Available: https://www.elfa.se/Web/Downloads/_t/ds/409510_2_eng_tds.pdf.

[14] Trampa, *Vedder esc for dc and bldc motors*, 2020. [Online]. Available: https://trampaboards.com/1-x-vesc-6-mkv--p-27529.html#manuals.

[15] Hobbyking, *Turnigy graphene professional 12000mah 6s15c lipo pack w/ xt90turnigy graphene professional 12000mah 6s15c lipo pack w/ xt90*, 2021. [Online]. Available: https://hobbyking.com/en_us/turnigy-graphene-professional-12000mah-6s-15c-lipo-pack-w-xt90.html?___store=en_us.

[16] GensTattu, *Tattu 12000mah 22.2v 15c 6s1p lipo smart battery pack with as150 + xt150 plug (new version)*, 2021. [Online]. Available: https://www.genstattu.com/tattu-plus2-15c-12000mah-6s1p-as150-lipo-battery.html.

[17] Finder, *Modular monostable relays 20 a*, 2017. [Online]. Available: https://www.farnell.com/datasheets/2626616.pdf.

[18] Schneider Electric, *Zbvb3 green light block for head ø22 integral led 24v screw clamp terminals*, 2019. [Online]. Available: https://docs.rs-online.com/65cd/0900766b816e964f.pdf.

[19] ——, *Zbe101 single contact block for head ø22 1no silver alloy screw clamp terminal*, 2019. [Online]. Available: https://docs.rs-online.com/2ef4/0900766b816e9638.pdf.

[20] Omron, *Pushbutton switch (cylindrical 22/25-dia.) a22*, 2016. [Online]. Available: https://www.farnell.com/datasheets/2340500.pdf.

[21] Bulgin, *1550 standard and 1350 high inrush*, 2020. [Online]. Available: https://www.farnell.com/datasheets/2634477.pdf.

[22] Kraus and Naimer, *Catalog 120 cg, ch, chr switches 10 a-25 a*, 2003. [Online]. Available: https://www.elfa.se/Web/Downloads/rc/g4/frCG4.pdf.

[23] Maxongroup, *Ecx torque 22 l*, 2021. [Online]. Available: https://www.maxongroup.com/medias/sys_master/root/8882191040542/EN-21-226.pdf.

[24] Shimano, *Shimano steps drive unit for coaster brake*, 2021. [Online]. Available: https://bike.shimano.com/en-EU/product/component/citytrek-ebike-e6000/DU-E6010.html.

[25] National Instruments, *What is labview?* [Online]. Available: https://www.ni.com/sv-se/shop/labview.html.

[26] Cython, *Cython c-extensions for python.* [Online]. Available: https://cython.org/.

[27] Y. Wang, 'Issues in imu and suggestions,' Unpublished, 2021.

[28] Traco Power. 'Traco power dc/dc converter thl 15wi series emi consideration.' (2020), [Online]. Available: https://www.tracopower.com/int/media/3673/download.

[29] ——, 'Traco power dc/dc converter thl 15wi series, 15 watt.' (2021), [Online]. Available: https://www.farnell.com/datasheets/2865002.pdf.

[30] Stelvio Kontek. 'Terminal blocks and connectors for p.c.b.' (2014), [Online]. Available: https://www.elfa.se/Web/Downloads/74/07/04837407.pdf.

[31] Eska. 'G-sicherungseinsätze 5 x 20 mm.' (2003), [Online]. Available: https://www.elfa.se/Web/Downloads/_t/ds/522-500serie_ger-eng_tds.pdf.

[32] Osterrath. 'Sicherungshalter.' (1991), [Online]. Available: https://www.elfa.se/Web/Downloads/_t/ds/82164411_eng_ger_tds.pdf.

[33] Kemet. 'Surface mount multilayer ceramic chip capacitors (smd mlccs).' (2019), [Online]. Available: https://api.kemet.com/component-edge/download/specsheet/C1812C475K5RACTU.pdf.

[34] ——, 'Surface mount multilayer ceramic chip capacitors (smd mlccs).' (2016), [Online]. Available: https://api.kemet.com/component-edge/download/specsheet/C0805C331J1GACTU.pdf.

[35] TDK. 'Power line chokes.' (2015), [Online]. Available: https://www.elfa.se/Web/Downloads/_t/ds/B82721A_J_K_eng_tds.pdf.

[36] ST. 'Sm6t.' (2008), [Online]. Available: https://www.farnell.com/datasheets/1761324.pdf.

[37] Bourns. 'Sma6j-q transient voltage suppressor diode series.' (2015), [Online]. Available: https://www.elfadistrelec.fi/Web/Downloads/_t/ds/Bourns_SMA6J-Q_eng_tds.pdf.

[38] Onsemi. 'Schottky rectifier ss32 - s310.' (2021), [Online]. Available: https://www.farnell.com/datasheets/2303855.pdf.

[39] JKSH (Members of LabVIEW Advanced Virtual Architects community), *Python and labview combination work*, 2021. [Online]. Available: https://lavag.org/topic/22072-python-and-labview-combination-work/?do=findComment&comment=136219.

[40] U. ERDİNÇ, 'Modelling, validation and control of an autonomous bicycle,' M.S. thesis, Chalmers University of Technology, Gothenburg, Sweden, 2019.

[41] National Instruments, *Integrating c code with labview on ni linux real-time targets.* [Online]. Available: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000YGvGCAW&l=sv-SE.

[42] ——, *Creating shared library for labview real-time or veristand on ni linux rt target.* [Online]. Available: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000YGNdCAO&l=sv-SE.

[43] Cython, *Cython - an overview.* [Online]. Available: https://cython.readthedocs.io/en/latest/src/quickstart/overview.html.

[44] Python Wiki, *Windowscompilers.* [Online]. Available: https://wiki.python.org/moin/WindowsCompilers.

[45] akbar E-z, *techseries8 - section 03 - pip install cython opencv-python.* [Online]. Available: https://www.youtube.com/watch?v=eBJhUKou1lw.

[46] M. Zawawi and A. Fdhil, *Requirement management plan*, 2020.

[47] L. Provenzano, 'Reviewing requirements [powerpoint slides],' Unpublished, 2020.

A User Guide for Autobike 2021

Author: Love Bridén

1.1. Introduction

This document describes how to operate the hardware of the Autobike properly. There are some functions that a user can utilise in the hardware, those are; the circuit breakers, the reset button, the LED, the E-stop button, the power switch and a fuse. This document also describes how to use the cable index sheet in order to set up the system properly.

1.2. User interfaces

1.2..1 Circuit Breakers

The two circuit breakers included in the architecture (figure 24.) has a knob each that has two states; state 1 and state 2. If the circuit breakers are set up properly (as described in appendix B) state 2 disables current to flow to the motor controllers (excluding current from communication links) which in turn disables the motors of the Autobike. Each motor controller is connected to a circuit breaker enabling the isolation of none, one, the other or both of the motor controllers. If the circuit breakers are set to state 1 (and properly set up) they allow current to flow to the motor controllers of the Autobike and, in turn, enable the motors to be used. The circuit breakers shall remain on state 2 until a user decides to use the motors of the Autobike. See figure 24.

1.2..2 Reset

In appendix C the functionality of the reset button is described. For simplicity's sake some of the information is repeated in this document. When a user wants to enable the usage of the motors of the Autobike the user shall push the reset button for a slight moment. If this is done while the circuit is properly set up and the power switch "on" it will allow current to flow to the motors of the Autobike. If the reset button is pushed and held it will allow current to flow to the motors of the Autobike regardless of the state of the E-stop button, this can result in an unsafe override of the E-stop button. Therefore the reset button shall not be pressed for any longer than one second at a given time. See appendix C and figure 24.

1.2..3 LED

The LED included in the architecture is connected to on end of the coil in the finder and to ground. When current flows through the coil of the finder (resulting in current being able to flow to the motors of the Autobike) the current will also flow through the LED resulting in it illuminating. In other words, when the LED is illuminated current can flow to the motors of the Autobike, when the LED is not illuminated current can not flow to the motors of the Autobike (if the circuit is properly set up). If a user wants to use the functionality of the reset button the results of the action are provided by the LED. If it illuminates when the reset button is pressed, it was done successfully. The LED also provides results from pressing the E-stop button, if the LED is not illuminated after pressing the E-stop button, it was done successfully. See appendix C and figure 24.

1.2..4 E-Stop

The E-stop button in the architecture can be pressed and thereby disallowing current to flow to the motors of the Autobike. If it is pressed and a user wants to reset it the user shall reset it by rotating it in the direction of the arrows on top of it. This will allow it to spring up to its original position. The E-stop button is meant to be pressed in a potentially harmful situation to minimise the risks of any harm. It will result in no more acceleration by the motors of the Autobike and it thereby falling and stopping. See appendix C and figure 24.

1.2..5 Power Switch

The power switch in the circuit has two states, either "on" or "off". In the "off" state the four pins of the power switch are not connected and the button of the switch is black. In the "on" state two sets of pins are connected to each other, in this state the button of the switch has a green sliver. See appendix C and figure 24.

1.2..6 Fuse

The PDB featured in the Autobike has a fuse that burns when 3.15 A is flowing through it. The fuse is placed on the same side of the PCB as the connectors. The fuse features a thin wire in the glass casing, if this wire is not whole the fuse shall be replaced. See appendix D.

1.2..7 Cable Indexing

In order to properly set up the system and to evaluate if the system is properly set up a document was created. This document features information of what cable should be connected to what component and to what pin. The header row of the document features number of the wire, between what components the cable is connected to, colour of the cable, the pins of the components, comments, colour of the indexes and abbreviations. For example, wire number 18 has the interface entry Steering Motor - ESCON, meaning that the wire connects these two components. The colour of the cable is black, providing information of how to find it in an eventual cluster of cables. The pins are Steering Motor (Black wire) and ESCON Motor 2. The steering motor has no pins, rather, it has two wires where one of the two ends of the wires are connected internally to the motor, the wires are coloured differently, thereby the black wire is identifiable. The ESCON has a group of pins labelled "Motor" and pin 2 is identifiable among those pins. The comment is vacant on this wire as it was deemed unnecessary to provide one. The number index consists of two digits, 1 and 8. These two digits are coloured differently. The colour of the first digit of the index is brown and the colour of the second digit of the index is grey. To conclude, the black wire from the steering motor which is labelled with the number 18 where the one is brown and the eight is grey should be connected to pin Motor 2 of the ESCON motor controller. See appendix B for the full cable indexing document.

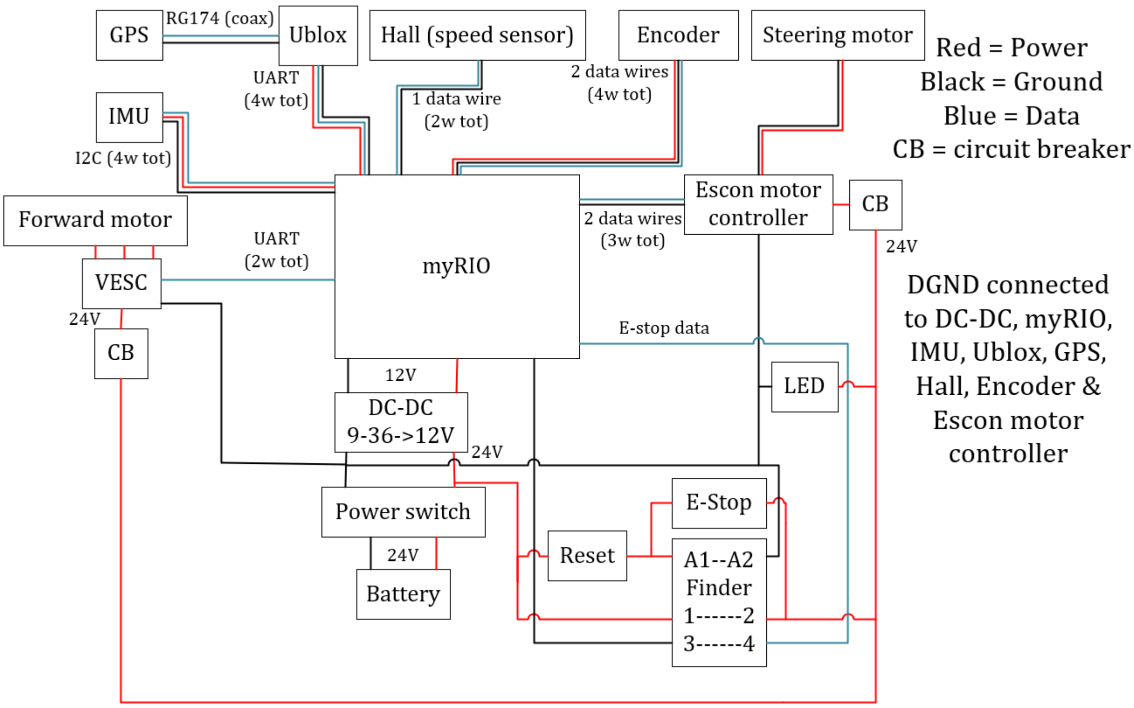


Figure 24: Picture of the electrical/electronical architecture of the Autobike

B Cable index

Author: Love Bridén

Content of appendix:

1. MyRIO MPX connections index
2. Cable index, colour coordination index
3. MyRIO I/O PCB Connection index

2.1. MyRIO MPX connections

Pin number and functionality	MyRIO MXP A Pinout:	MyRIO MXP B Pinout:
1: +5V		
2: AO0		
3: AI0		
4: AO1		
5: AI1		
6: GND		
7: AI2		
8: GND		
9: AI3		
10: UART.RX	UART.RX (Ublox)	UART.RX (VESC)
11: DIO0		
12: GND		
13: DIO1		
14: UART.TX	UART.TX (Ublox)	UART.TX (VESC)
15: DIO2		
16: GND		
17: DIO3		
18: DIO11/ENC.A	DIO11: Digital read (ENC.A)	
19: DIO4		DIO4: Digital write (ESCON enable)
20: GND		
21: DIO5/SPI.CLK		
22: DIO12/ENC.B	DIO12: Digital read (ENC.B)	
25: DIO7/SPI.MOSI		
26: DIO13	DIO 13: Digital read (ESTOP)	
27: DIO8/PWM0		PWM0 (ESCON rotate)
28: GND		
29: DIO9/PWM1		
30: GND		
31: DIO10/PWM2		
32: DIO14/I2C.SCL		I2C.SCL
33: +3.3V		
34: DIO15/I2C.SDA	DIO15: Digital read (HALL)	I2C.SDA

Table 2: Connection with MyRIO MXP connectors. Pins without any text are either unused or they are passive (in the sense that they provide the requested functionality without the use of code produced by a user).

Number	Interface	Colour of cable	Pins	Comment	Colour of index	Colour two of index
0	Battery - Power Switch	Black	Battery - and Power Switch 2A	Index is located on the thinner of the two wires	Black	NA
1	Battery - Power Switch	Red	Battery + and Power Switch 5B	Index is located on the thinner of the two wires	Brown	NA
2	Power Switch - PDB	Blue	Power Switch 4B and PDB 4		Red	NA
3	Power Switch - PDB	Blue	Power Switch 1A and PDB 3		Orange	NA
4	PDB - MyRIO	Black	PDB 2 and MyRIO power input (-)		Yellow	NA
5	PDB - MyRIO	Black	PDB 1 and MyRIO power input (+)		Green	NA
6	Power Switch - Finder	Black	Power Switch 4B and Finder 1		Blue	NA
7	Finder - Reset	Blue	Finder 1 and Reset 3	Index located on both ends of cable	Purple	NA
8	Finder - MyRIO	Blue	Finder 3 and PCB GND	The PCB is used as an interface	Grey	NA
9	Finder - MyRIO	Blue	Finder 4 and PCB ESTOP	GND located besides ESTOP	White	NA
10	Finder - Reset - E-stop	Blue	Finder A1, E-stop 1 and Reset 4	Index located on both ends of cable	Brown	Black
12	LED - Power Switch	Blue	Power Switch 1A and LED 1	Index located on both ends of cable	Brown	Red
13	Finder - E-stop - Circuit Breaker - Circuit Breaker - LED	Blue	Finder 2, E-stop 2, Circuit Breaker 2, Circuit Breaker 2 and LED 2		Brown	Orange
14	Power Switch - VESC	Blue	Power Switch 1A and VESC IN -		Brown	Yellow
15	Power Switch - ESCON	Blue	Power Switch 1A and ESCON IN -		Brown	Green
16	Circuit Breaker - ESCON	Blue	ESCON IN + and Circuit Breaker 1		Brown	Blue
17	Circuit Breaker - VESC	Blue	Circuit Breaker 1 and VESC IN +		Brown	Purple
18	Steering Motor - ESCON	Black	Steering Motor (black wire) and ESCON Motor 2		Brown	Grey
19	Steering Motor - ESCON	Red	Steering Motor (red wire) and ESCON Motor 1		Brown	White
20	VESC - MyRIO	Black	VESC COMM RX and PCB TX	Located below Traco Power DC-DC	Red	Black
21	VESC - MyRIO	Black	VESC COMM TX and PCB RX	Located below Traco Power DC-DC	Red	Brown
22	ESCON - MyRIO	Blue	ESCON Digital 1 and PCB PWM		Red	Red
23	ESCON - MyRIO	Blue	ESCON Digital 2 and PCB EN		Red	Orange
24	ESCON - MyRIO	Blue	ESCON Digital 5 and PCB GND	Located besides EN	Red	Yellow
25	Encoder - MyRIO	Black	Encoder 1 and PCB GND	Green wire to Encoder 1 (pin at index)	Red	Green
27	Encoder - MyRIO	Black	Encoder 3 and PCB ENC.A	Orange wire to Encoder 3	Red	Purple
28	Encoder - MyRIO	Black	Encoder 4 and PCB 5V	Red wire to Encoder 4	Red	Grey
29	Encoder - MyRIO	Black	Encoder 5 and PCB ENC.B	Brown wire to Encoder 5	Red	White
30	Hall sensor - MyRIO	White	Hall sensor (wire 30) and PCB Hall		Orange	Black
31	Hall sensor - MyRIO	Purple	Hall sensor (wire 31) and PCB GND	GND located besides HALL	Orange	Brown
32	Ublox - MyRIO	Black	Ublox GND and PCB GND	GND located on opposite side as MXP connectors	Orange	Red
33	Ublox - MyRIO	Red	Ublox 3V3 and PCB 3.3V	3.3V located on opposite side as MXP connectors	Orange	Orange
34	Ublox - MyRIO	Blue	Ublox RX/MOSI and PCB TX	TX located on opposite side as MXP connectors	Orange	Yellow
35	Ublox - MyRIO	Yellow	Ublox TX/MISO and PCB RX	RX located on opposite side as MXP connectors	Orange	Green
38	IMU - MyRIO	Blue	IMU 2 and PCB SDA	IMU I2C SDA	Orange	Grey
39	IMU - MyRIO	White	IMU 4 and PCB SCL	IMU I2C Clock	Orange	White
40	IMU - MyRIO	Black	IMU 5 and PCB RX	RX located on 6-port terminal below Traco Power DC-DC	Yellow	Black
41	IMU - MyRIO	Red	IMU 6 and PCB 3.3V	IMU VCC 3.3/5V	Yellow	Brown
42	GPS Antenna - Ublox	Grey	GPS and Ublox Active Antenna L1/L2		Yellow	Red

Table 3: Cable index, colour coordination

2.2. MyRIO I/O PCB (clockwise order)

IN	OUT (MXP connectors)
SPI GND	GND
SPI 3.3V	3.3V
SPI MOSI	A 25: DIO7/SPI.MOSI
SPI MISO	A 23: DIO6/SPI.MISO
SPI CLK	A 21: DIO5/SPI.CLK
SPI CS	A 19: DIO4
GND	GND
5V	+5V
ENC.B	A 22: DIO12/ENC.B
ENC.A	A 18: DIO11/ENC.A
ENC B	A 22: DIO12/ENC.B
ENC A	A 18: DIO11/ENC.A
ESTOP	A 26: DIO13
GND	GND
HALL	A 34: DIO15/I2C.SDA
GND	GND
UART GND	GND
UART 3.3V	+3.3V
UART TX	A 14: UART.TX
UART RX	A 10: UART.RX
GND	GND
AIN	A 3: AI0
GND	GND
EN	B 19: DIO4
PWM	B 27: DIO8/PWM0
myRIO -	NC
myRIO +	NC
Battery -	NC
Battery +	NC
UART GND	GND
UART RX	B 10: UART.RX
UART TX	B 14: UART.TX
UART VCC	NC
RX	GND
3.3V	+3.3V
SHTN2	B 22: DIO12/ENC.B
SHTN1	B26: DIO13
SDA	B 34: DIO15/I2C.SDA
SCL	B 32: DIO14/I2C.SCL

Table 4: MyRIO I/O PCB Connection index

C Description of Finder Circuit (Project Autobike)

Author: Love Bridén

3.1. Introduction

This document describes the functionality and schematic of the "finder circuit" present in the Autobike. The picture found in the very end of this document is meant for illustration purposes and to complement the picture of the architecture (not found in this document). Section 3.2. describes the basic and relevant, for this purpose, information about the components in the schematic. Section 3.3. describes the functionalities of some of the components of the circuit. The last section, section 3.4., provides information about the user inputs and how to operate the circuit properly.

3.2. Component descriptions

The components whose names are seen in the subsections of this section are described according to relevant, for this documents' purpose, information. Other components are not described any further (other than stating what identifier they have in the picture at the vary end of this document). The "VESC" is the forward motor controller. "CB" stands for "Circuit-Breaker". If "CB" is followed by a number, it represents the current (in ampere) for which the circuit breaks, if no number is present it is a manual switch.

3.2..1 Finder

The finder has six ports and a button. Port "1" to "2" are by default open, meaning that they are not connected. While the button is pressed the ports close and connect meaning that they conduct electricity towards one another for the duration that the button is pressed. The ports "3" to "4" work oppositely to what is described in the sentences above meaning that they are by default closed and open while the button is pressed. The ports "A1" to "A2" are connected through a coil. When current beyond a threshold flows through the coil it pushes and holds the button, i.e. when a current flows through "A1" to "A2" the button presses.

3.2..2 Power Switch

The power switch has four pins and two states. In one state none of the pins are connected i.e. they are open. When in the other state two pairs of two pins are connected to each other resulting in a two pole switch.

3.2..3 Reset

The reset button has two pins and two states. In the default state the pins are not connected (open circuit). While pressed the pins connect (closed circuit). The button does not toggle (meaning that it stays pushed after an initial push until pushed again), rather it is sprung back to the default state immediately when the user stops pushing the button.

3.2..4 E-Stop

The emergency stop button has two pins and two states. When in the default state, the two pins are connected (closed circuit). When the button is pressed the pins disconnect. This state remains until the button is reset to its initial state by rotating the button allowing it to spring back up.

3.2..5 LED

The LED in the circuit is a traditional Light Emitting Diode. When current flows through it, it illuminates. When no current flows through it, it does not illuminate.

3.3. Circuit Description

The circuit is described according to the user interfaces presented in the following subsections. Not shown in the picture is the power supply of the MyRIO device. The device is connected to a DC-DC converter connected to the nets directly after the Power switch (i.e. not the same side of the power switch as the battery is connected to), this results in a circuit that can disconnect any motors without disconnecting the MyRIO.

3.3..1 Power switch

In the circuit, the power switch is used to disconnect the battery from the rest of the circuit. This disconnects the energy source of the (rest of the) circuit and thereby allows a user to safely inspect, alter, measure etc. the rest of the circuit.

3.3..2 Reset

The reset button works as a "play" button in the circuit. When the user wants to allow current to flow towards the motor controllers the user can press the reset button allowing current to flow through the coil of the finder. This pushes the button and thereby closes the connection between pin "1" and "2" of the finder. This will allow current to flow from pin "2" through the E-Stop button to pin "A1" and thereby continuing to allow the flow of current through the coil of the finder. This setup will allow the following: when the reset button is pushed (and released) the button on the finder is pushed and held pushed allowing the reset button to actually work as a reset button as opposed to a "dead man's switch".

3.3..3 E-Stop

The emergency stop button connects pin "2" and "A1" of the finder allowing current to flow through the coil of the finder even after the reset button is released. On the contrary, pushing the E-Stop button disconnects these previously mentioned pins. This will release the button of the finder that in turn will disconnect the energy source of the circuit from the motor controllers. This will remove power from the motors of the Autobike allowing any unsafe behaviour to be disrupted.

3.3..4 LED

The LED illuminates when current flows through it. Pushing the button of the finder allows the motor controllers to consume power allowing the motors to operate. Current flowing through the coil of the finder pushes the button of the finder. The LED illuminates when currents flows through it and current can reach the LED if current is flowing through the coil of the finder. This makes the LED an indicator for when the button of the finder is pushed (via a current through the coil of the finder). When the button of the finder is pressed the LED is illuminated.

3.3..5 MyRIO

The MyRIO is connected to pin "3" and "4" of the finder. These pins are connected by default and disconnected when the button of the finder is pushed. If the MyRIO sends a digital high signal to one of those finder pins and a digital input signal is acquired from the other pin of the finder, data about the state of the button of the finder can be obtained. The digital signal will be high from the moment that the E-Stop button is pressed until it is released and the reset button is pressed. This digital input will be used in the code of the Autobike to determine proper actions.

3.4. User Guide

When a user wants to operate the Autobike the user has to push the Power switch, connecting the battery to the rest of the circuit. After this the E-Stop button has to be reset if it was previously pushed, otherwise skip this step. When the user is ready for the Autobike to operate the user shall push the reset button (this will push the button of the finder allowing power to flow towards the motor controllers and thereby the motors of the Autobike). If this is done correctly (and the circuit is connected correctly) the LED will illuminate indicating that the Autobike is ready for operation. When in a state requiring the Autobike to stop its actuators from operating the E-Stop button shall be pushed, this will darken the LED indicating that no power is directed to the motor controllers and thereby motors of the Autobike. When the user is ready to operate the Autobike again the E-Stop button shall be reset, the power switch shall remain on and the reset button shall be pushed. If this is done correctly this will re-illuminate the LED.

3.4..1 Forbidden User Inputs

The circuit has one major drawback. If the reset button is pressed and held it will override the E-Stop button removing the functionality of it. If the E-Stop button is pressed the motors will receive no power. If the reset button is pressed while the E-Stop button remains pressed it will allow power to reach the motors for the duration that the reset button is pressed. This is dealt with by never holding the reset button pressed for longer than the duration of a tap (> 1s) and never pushing it when the E-stop button is pressed due to a hazardous event.

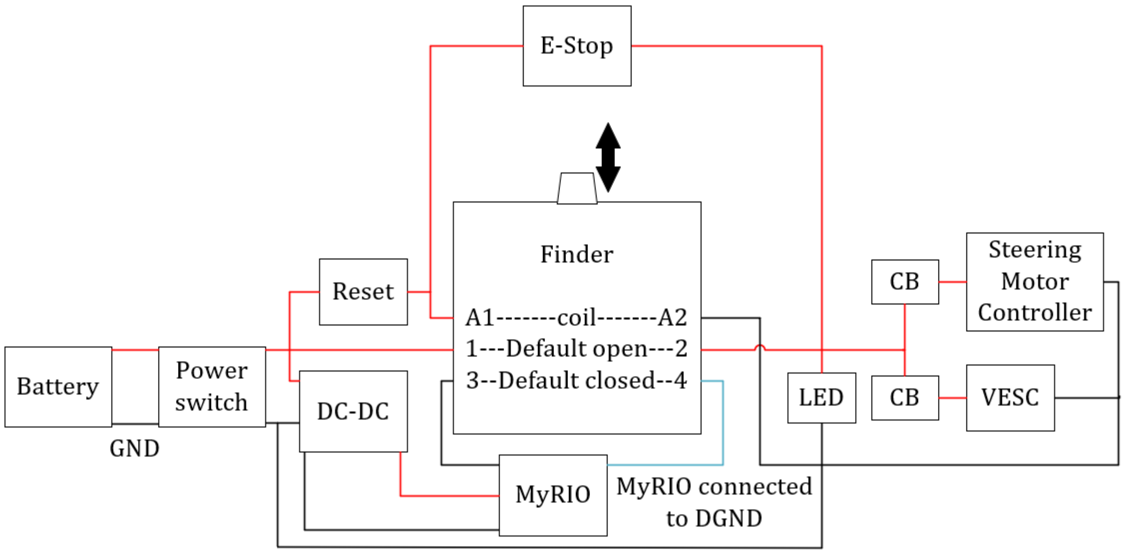


Figure 25: Schematic of "Finder circuit"

D Power Distribution Board for Autobike 2021

Author: Love Bridén

4.1. Introduction

This document describes the process of designing and manufacturing the new Power Distribution Board (PDB) for the Autobike. While starting this project a PDB designed by students from Chalmers University of Technology (CTH) was received (by the students of MDH). While studies at CTH continued they realised that signals from the Inertial Measurement Unit (IMU) underwent disturbances [27]. This was thought, by Henrik Falk, to be caused by the PDB which also features inputs and outputs to and from the myRIO device (including the IMU signals). To try to figure out what was going on with the faulty sensor values testing of the PDB were conducted (see appendix E). During the testing a major component of the PDB accidentally were damaged. To fix this the processes of attaining a new PDB with proper robustness features began.

4.2. Method

First off, documentation of the DC-DC (Direct Current) converter made by Traco Power suggests that capacitors and inductors should be used where some of them form an LC filter to mitigate effects noise may have on the circuit [28]. In addition, Transient Voltage Suppression- (TVS), reverse polarity protection diodes and a fuse were included for even more protection. When the circuit was finalised appropriate components were chosen from online vendors. With the circuit finalised and components chosen the circuit board design could begin using NI Ultiboard. In the program the footprints of the components were created using the information provided in each components data-sheet. As previously mentioned a fuse was included in the circuit, that fuse, with corresponding fuse holders as well as the circuit board terminal and the actual board the circuit were to be printed on, was found at the Collaborative Center (C^2). The footprints of the fuse holders were created by measuring the relevant measures for this purpose using calipers and creating components with footprints featuring those measures. When all the footprints were created the placement of the components on the circuit board could begin. With the placement complete the routing of traces began. Since the circuit only features a few components only one side of it were used. With the design completed the relevant files were exported and converted using Carbide Copper into G-code. The G-code was inserted into a Wegstr CNC machine and the circuit board and appropriate milling tool (V-cut 45°) were installed. When the milling was completed a slight connectivity test was conducted. After the test had been successful the drilling could commence, this was to enable the few through hole mounted components to be mounted on the circuit board. The drilling was done using the Wegstr machine using a 0.8 mm drilling tool. Some of the components have slightly thicker pins, to counter this another drilling effort was done, this time by using a handheld drilling machine and the thinnest drill bit available at C^2. These new larger holes allowed the through hole mounted components to be mounted on the circuit board. To properly mount the components they were soldered in place using a handheld soldering machine.

4.3. Results

The following components were used on the circuit board.

4.3..1 DC-DC Converter

The DC-DC converter used in the circuit is Traco Powers THL 15-2412WI. It has a broad input of 9-36 V and an output of 12 V. The output voltage has a maximum ripple of 100 mV peak to peak. The maximum power the converter can output is 15 W and the maximum output current is 1.25 A. The converter has a typical efficiency of 88% and has a switching frequency of 330 kHz.

In the circuit viewed in figure 26 the component is depicted as "DC-DC Converter". [29].

4.3..2 Terminal block

The terminal block used to access the circuit board is the 691136711104 terminal block created by Würth Elektronik. It is a four pole through hole mounted terminal block. The terminal block is rated for up to 24 A and 250 V. The terminal block is compatible for cables with sizes between 14 and 30 American Wire Gauge (AWG). The terminal block is able to lock cables in place providing a mechanically robust interface for the circuit board. In figure 26 the terminal block will connect the circuit board to the "Battery" and the "MyRIO". [30]

4.3..3 Fuse

The fuse included on the circuit board is 522.522 made by Eska. It is a slow burn fuse which burns at 3.15 A. It is rated for 250 V and has a breaking capacity of 35 A. The fuse has a power dissipation of 1.6 W and a maximum voltage drop of 100 mV. In figure 26 the fuse is depicted as F1. [31]

4.3..4 Fuse Connectors

The fuse connectors used on the printed circuit board is the 82-1644-11 connectors made by Osterrath. They are made of a copper alloy, more specifically, CuZn37 F45. In figure 26 the fuse connectors are located on each side of F1. [32]

4.3..5 Capacitor type 1

The type 1 capacitor used in the circuit is Kemets C1812C475K5RACTU. It is a multi layer ceramic capacitor with a capacitance of 4.7 μF and a rated voltage of 50 V (DC). The capacitor has a tolerance of 10% and has an operating temperature range between -55° and 125°C. In figure 26 C1 and C2 are type 1 capacitors (this type). [33]

4.3..6 Capacitor type 2

Type 2 capacitor in the circuit is Kemets C0805C331J1GACTU. The capacitor has a capacitance of 330 pF within 5 % and is of multi layer ceramic type. The capacitor has an operating temperature range of -55° to 125°C. The capacitor is rated for DC voltages up to 100 V. In figure 26 the capacitor is labelled as C3 and C4. [34]

4.3..7 Inductor

The inductor featured in the circuit is Epcos B82721A2362N001. The inductor has an inductance of 400 μH and a tolerance of 30%. The inductance is rated for 250 V and 3.6 A. The inductance features a resistance of 35 mΩ. The inductance consists of two inductors wound on the same ferrite core. In figure 26 the inductor is labelled as L1. [35]

4.3..8 Zener diode type 1

The type 1 zener diode used in the circuit is ST:s SM6T30CA. The diode features a peak pulse current of 75 A and power dissipation of 600 W. The diode has a maximum breakdown voltage of 31.5 V and a clamping voltage of 53.5 V. It has a maximum reverse leakage current of 200 nA and an operating temperature range between -55° and 150°C. In figure 26 type 1 zener diode is labelled as D3. [36]

4.3..9 Zener diode type 2

Zener diode type 2 used in the circuit is Bourns SMA6J14A-Q. The diode operates in temperatures between -55° and 150°C. The component has a peak pulse power dissipation of 600 W and current of 40 A. The diode has a maximum breakdown voltage of 17.2 V and a reverse leakage current of 1 uA. In figure 26 type 2 zener diode is labelled as D4. [37]

4.3..10 Schottky diode

The schottky diodes included in the circuit, labelled as D1 and D2 in figure 26, are Onsemis SS34. The diode has a power dissipation of 2.27 W and a forward voltage drop of 500 mV. The diode has a peak average forward current of 3 A and a maximum reverse repetitive voltage of 40 V. The diode can operate in temperatures between -55° and 150°C. The diode has a maximum reverse current of 500 uA. [38]

4.3..11 Actual board

The board on which the circuit is printed and components soldered on is Scankemis 110051. The board consists of FR4 epoxy resin and a 35 um thick lamination layer of copper. The measures of the board is a length of 160 mm, a width of 100 mm and a thickness of 1 mm. The board is single-sided meaning that there exists only one copper lamination.

4.3..12 Circuit

The schematic of the circuit can be seen in figure 26, it is found in section 4.3..14. D1 and D2, the schottky diodes, are meant to protect sensitive components in the circuit from reverse polarity inputs from all poles of the connectors. D3 and D4, the zener diodes, are meant to protect sensitive components in the circuit from transient voltages. F1, the fuse, is meant to protect the DC-DC converter from high input currents. The capacitors and inductor are meant (L1, C1, C2, C3 and C4) to mitigate impacts of electromagnetic compatibility. The DC-DC converter is meant to convert the DC voltage from the battery to a lower level within the range accepted by the MyRIO. The MyRIO is the microcontroller featured in the Autobike responsible for among other receiving data from the sensors of the Autobike, calculate proper actions and send said data to the actuators of the Autobike.

4.3..13 Layout

The layout of the components and traces on the circuit board can be seen in figure 27. The fuse is held in place by the fuse connectors (the two unlabeled components located just below the fuse), the fuse is placed above the fuse connectors to get the spacing between them right. All through hole mounted components (the connector, fuse connectors, inductor and DC-DC converter) and the fuse are mounted on the bottom side of the circuit board. This is to make it possible to solder the components to the correct net of the circuit board. The connector, fuse connectors and inductors feature such a symmetry that allows them to be mounted on the bottom of the circuit board even if the Ultiboard design has them placed on the top of the circuit board. The DC-DC converter was accidentally mirrored in the design and it does not feature such a symmetry as the previously mentioned components. Having it (the mirrored DC-DC converter) placed on the top of the circuit board in the Ultiboard design allows for it to be placed on the bottom of the physical circuit board and thus enabling for it to be soldered on to the correct nets of the circuit board. The circuit board features a connector for in- and output purposes, the four different poles of the connector are numbered "1" through "4". The following numbers correspond to the following nets of the circuit board; 1 - Vout (12 V), 2 - Digital Ground, 3 - Analog Ground (battery -) and 4 - VCC (battery +).

4.3..14 Figures

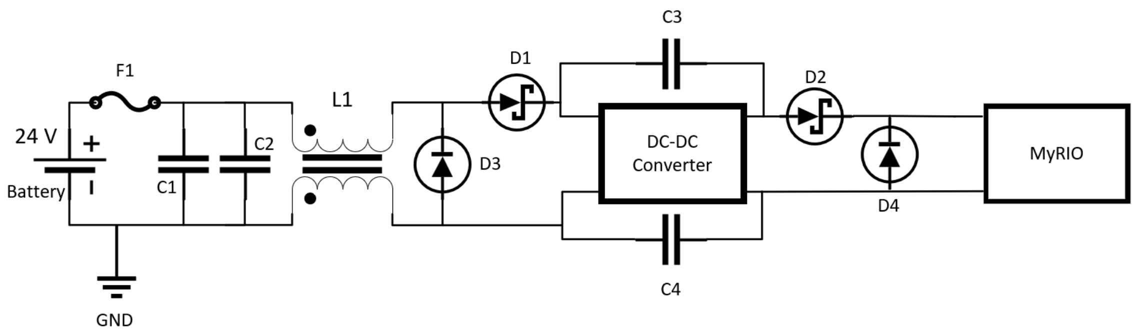


Figure 26: Schematic of the new PDB. The battery, ground and the MyRIO device are connected to the terminal of the circuit board, thus they are not located on the board.

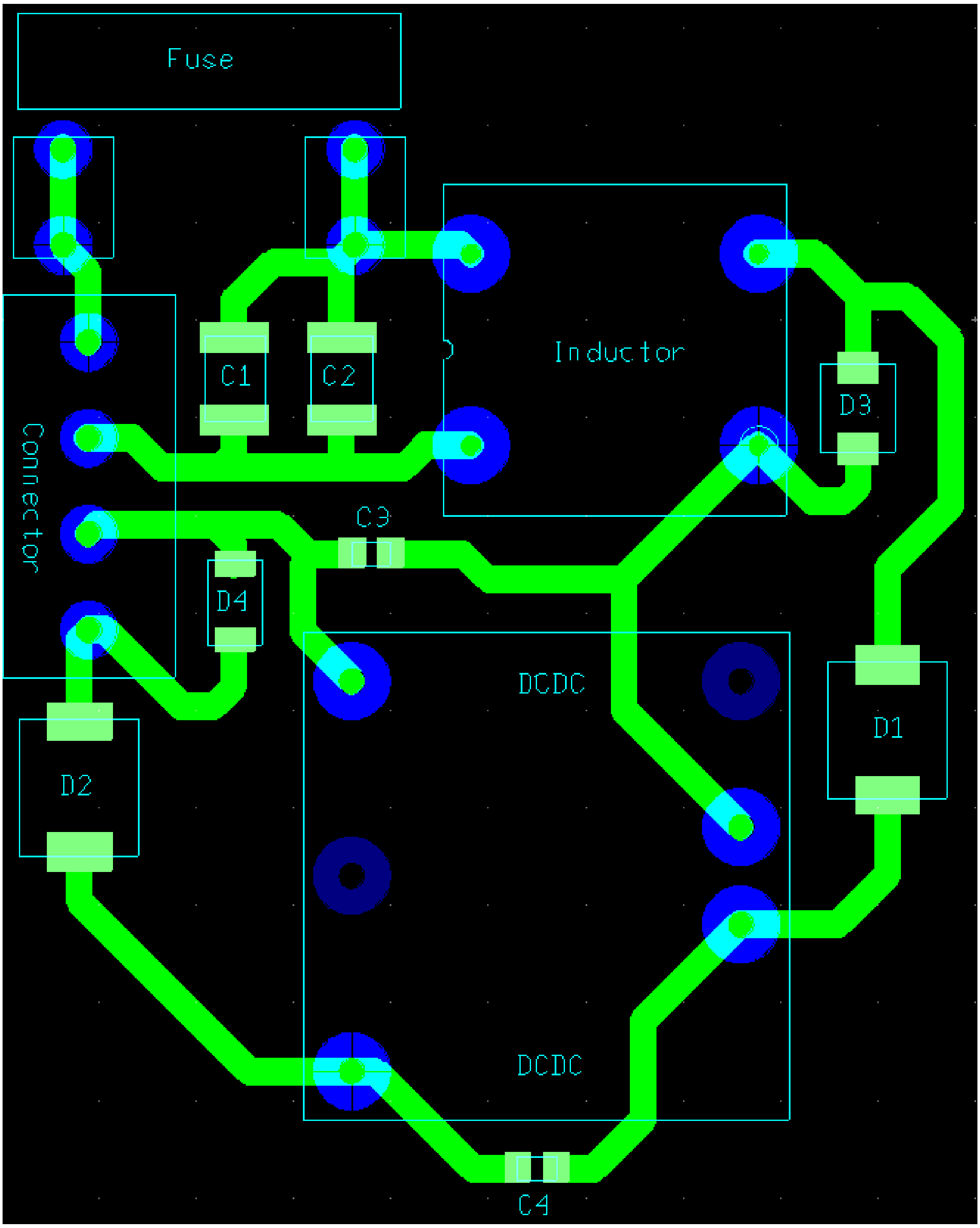


Figure 27: Component and trace layout for new PDB. The fuse is mounted on the fuse connectors, in the layout the fuse component is placed besides the fuse connectors for illustration purposes.

E Test of PCB designed and manufactured by Chalmers University of Technology

Author: Love Bridén

5.1. Motivation

The testing concerns the Power distribution and input-output board for the MyRIO device produced by Chalmers University of Technology. A teacher at MDH recommended that the PCB (Printed Circuit Board) should be tested to figure out what caused the IMU (Inertial Measurement Unit) signal to be disturbed.

5.2. Method and Outcomes

The testing was conducted by applying a 24-volt potential difference between the "Battery +" and "Battery -" ports of the PCB using a power supply and connecting a DC (Direct Current) motor to the "myRIO +" and "myRIO -" ports. The DC-DC converter on the board converted the 24-volt potential to a 12-volt potential towards the DC motor. The DC motor is rated at 12-volt and spun accordingly. Using an oscilloscope measuring the voltage difference between the two separate grounds (analog and digital) and later the difference between the digital ground and the input ports (one after the other) where the IMU output data signal (SPI MISO) should be connected during runtime. For all these measurements points the voltage difference measured by the oscilloscope was around 100-mV. When the DC motor was slightly obstructed by gripping the rotational shaft and thereby applying a mechanical force to it, the DC-motor received a higher amount of current from the converter and the measured potential (of the oscilloscope) rose to around 200mV for all ports tested.

Later the DC motor was changed for a 24-volt rated DC-motor. When the motor received the 12-volts potential from the PCB it started moving but could not rotate fully, the potential was too small. When the motor was helped in the rotation (by mechanical inputs to the rotational shaft) it suddenly began spinning full rotations. When the motor was obstructed (by the same principle as previously mentioned) the current drawn from the motor was even greater than the previous DC-motor and the potential measured by the oscilloscope rose to a peak value of around 500mV, this was when a relatively small amount of current (compared to what is drawn during a full test of the Autobike) was drawn by the motor. Further testing was conducted, and the motor was helped again using mechanical inputs, this time the motor stopped and could not rotate even a little with mechanical inputs. This indicated that the poles of the motor were short circuited by some fault in the PCB. Later investigation using a connectivity testing voltmeter indicated that the DC-DC converter was destroyed by the excessive voltage received at its outputs (generated by the mechanical inputs to the motor) and in the process short circuiting the two output poles of the DC-DC converter. This motivated the design and manufacturing of a separate PDB (power distribution board) to begin with proper protection for the DC-DC converter featured to spare it from potential damage. Transient voltage suppression diodes and a fuse, among other measures, were included in the design.

F Testing of the Hall Sensor

6.1. Component Info

Author: Viktor Aronsson Karlsson

Part Number: MMPSA 240/100, proximity switch, reed switch based, rectangular, normally open contacts. [11]

6.2. Testing documentation

Author: Love Bridén
Testers: Love Bridén and Viktor Aronsson Karlsson

11/11: The Hall sensor was tested using the MyRIO device by measuring the voltage difference between the two poles of the sensor using the analog read function and the analog input ports of the MyRIO device. The sensor was placed near the back wheel of the bicycle (where the magnets are placed along the rim of the wheel) and the wheel was spun, this resulted in a changing magnetic field where the sensor was placed. The results were unsatisfactory because no significant voltage difference corresponding to the rotational speed of the wheel was visible.

Testers: Love Bridén, Maya Zawawi and Viktor Aronsson Karlsson

12/11: The hall sensor tested again using an oscilloscope measuring the voltage difference between the two poles of the sensor. The signal was noisy and when the sensor was moved across magnets the noise dropped, indicating that a magnet was near the sensor. The sensor was later tested using a MyRIO device where one pole of the sensor was receiving a digital high signal (approximately 3.3 V) from a DIO pin of the MyRIO device, the other pole was also put in a DIO pin of the MyRIO device measuring a digital input. The sensor was placed by the back wheel of the bicycle and the wheel was spun allowing the magnets placed on the rim of the wheel to pass by the sensor. When the magnets were close to the sensor the input was read to digital high, in other words the testing was successful. When the wheel was spun too fast the input did not turn high, this might be due to a sampling issue either with the MyRIO device (due to setup), the monitor used or the testers eyes. One theory to why this might have happened is that the setup used did not use the full capacity of the MyRIO device as the test was only meant to show if the sensor reacted to the magnets passing by. A more thorough test will be conducted at a later stage where the velocity will be measured.

Author: Viktor Aronsson Karlsson
Tester: Viktor Aronsson Karlsson

12/11: Created LabVIEW FPGA code that was able to read and write digital signals. The LabVIEW code read the digital signal and it became a digital High when the Hall sensor was next to a magnet. The code was then recreated, into two VIs where the FPGA read the signals and the host VI displayed the results in a wave form graph and counted the amount of detected digital highs.

A major discovery was made when going to the next step, which was to implement the Python code. The python code has worked in LabVIEW when executed on the computer environment, but when trying to implement the python nodes into the myRIO environment it was discovered that the FPGA or real time environments do not support python nodes. This problem is discussed in [39].

29/11: When implementing the sensor input in Labview FPGA, it was discovered that the DIO pins of both A and B have pull-up resistors, so they are always in a high state. This means that in order to read the Hall sensor value the second pin need to be set to low.

G Reset button troubleshooting

Author: Love Bridén

This process began with the verification of the E-stop digital signal to the MyRIO device. The setup necessary had already been set up and the MyRIO device was connected to the Printed Circuit Board (PCB) (referring to the circuit board hosting input and output connections to the MyRIO device) and the PDB. The E-stop related finder and PCB pins were connected properly. The setup was connected to a power supply set to 24V powering the circuit. The PDB did in fact power the myRIO device properly indicating further that it was properly designed and manufactured. When the reset button was pressed the LED illuminated for the duration that the button was pressed, when the button was released, the LED darken indicating that something was wrong within the setup. The myRIO did not correlate its Boolean E-stop to the hardware, it never turned high. The reset button was intended to be pushed and released that in turn pushes and latches the button on the finder, allowing it to forward current to the motor controllers of the circuit. When the button of the finder was pressed it latched and stayed in that state until the E-stop button was pressed. To tackle this a diode (an identical Schottky diode to what is used in the PDB) was used between the E-stop button and the A1 pin of the finder disabling the current to flow directly through the E-stop button without first flowing through the coil of the finder. This setup did not change the problem at hand, the finder button still did not latch by pressing the reset button. When this method did not solve the problem a multimeter measuring the current flowing through the coil of the finder was installed, it measured currents up to 11mA. Research about the finder was done and information of when the finder should switch was found. According to the datasheet of the finder [17] it needs 1W to flow through the coil for it to latch the button. With 24V as input voltage around 42mA was required to flow through the coil for it to latch the button. To try to reach this higher current a second LED was connected in parallel to the original LED, however, the problem remained. To increase the current flowing through the coil of the finder the LED was removed, and the output of the coil (finder A2) was connected straight to ground. This solved the problem by increasing the current flowing through the coil of the finder enough. The multimeter was used again measuring currents up to 49mA. With the setup thought to be correct the myRIO device still did not gather data about the state of the E-stop button. After extensive efforts to verify that the setup indeed was correct the last fault was identified. The wires connecting the finder (pin 3 and 4, the pins related to the E-stop signal) was connected to the E-stop related ports of the PCB. Those ports are connected to the MXP A output of the PCB. The myRIO was connected to the MXP B port of the PCB, when this was changed to MXP A the code worked as intended and the Boolean signal for E-stop correlated well with the state of the physical button.

H Testing of new PDB

Author: Love Bridén

The new PDB was finalised and tested, the method for testing is described in this document.

The test setup consisted of the PCB, an oscilloscope, a power supply and some wiring. One end of each of the four wires was connected to the connector of the PCB. The other ends of the two wires connected to the input ports of the connector were connected to the power supply (with the positive pole of the power supply connected to the Vcc port of the PCB and the negative pole to the AGND port of the PCB). The other two cables yet unused ends were connected to the oscilloscope (with the positive pole of the oscilloscope connected to the Vout pin and the negative pole connected to the DGND pin). The power supply was set to output 0 V and 0.1 A to begin with. Then, with time, the output voltage of the power supply was ramped up towards 30 V. At around 9 V (out from the power supply) the oscilloscope sprung from measuring 0 V to 12 V (accuracy was hard to determine). The output, for sure, remained within a 0.5 V margin of error from the 12 V signal but any more precise accuracy could not be determined with a high level of significance due to the oscilloscope not being able to scale into an optimal range for this purpose. However, the error was more likely 200 mV at most. Either way the output was within the 6-16 V range acceptable by the MyRIO device. The output voltage was seemingly identical when the input voltage was within the span of 9 to 30 V. This data hints towards the fact that the PCB was successfully designed and manufactured.

I Python in LabVIEW RT compatibility issues and proposed solutions

ID	Author	Changes	Date
1	Viktor Aronsson Karlsson	Creation of the document	15/11/2021
2	Viktor Aronsson Karlsson	Added the Section 'Conclusion'	21/12/2021

9.1. What is the problem

Author: Viktor Aronsson Karlsson
The problem is that Python nodes, which is the function that is used in LabVIEW to utilize the python code, doesn't work in Real-time or FPGA targetets [1]. The support for python nodes in cRIOs, which includes the myRIO, have been documented by other developers to be non-existent and python nodes only support the desktop version of LabVIEW [2].
[1] Python Node described by NI
[2] Python and Labview combination work

9.1..1 Problem with using Python for a Real-time system

Author: Viktor Aronsson Karlsson
Python for real-time application is not recommended by other software developers, this is because the compiling overhead that Python have over other languages. The compiling process of Python is that it first converts the code into byte code and then that byte code is executed in a python virtual machine, in comparison to for example C/C++ where the code is compiled into binary code and then executed directly on the CPU.
Is-Python-good-for-real-time-applications

Another reason that Python is discouraged to be used in a real-time environment is its memory handling process with the built-in feature 'Garbage Collector', where Python automatically frees up unused memory. The problem with the Garbage Collector is that when it wants to execute it will stop all other processes from executing. The Garbage Collectors execution frequency and duration is unknown, that in turn will cause unknown jitter throughout the whole system.
real-time-operating-via-python python-garbage-collection

9.2. Possible Solutions

9.2..1 Replacing the myRIO for a Single-Board RIO

Author: Love Bridén
Switching the myRIO for a CompactRIO does NOT solve the problem at hand. CompactRIO (an example of a single board RIO), just like the MyRIO, provides an FGPA and a Real Time (RT) target. LabVIEW RT or Linux can be run on the real time targets while the FPGAs only supports LabVIEW FPGA.
compactrio-single-board-controller

9.2..2 Standalone C/C++

Author: Maya Zawawi
In order to solve the problem that Python nodes doesn't work in a Real-time or FPGA targets. The official website of National Instrument (NI) suggested that the NI myRIO processor is totally programmable in C/C++ and gives the ability to program a processor running a real-Time.
Program in C/C++ to interact with the myRIO

9.2..3 C/C++ in LabVIEW RT

Author: Maya Zawawi
Calling a Dynamic Link Library (DLL) from LabVIEW allows the utilization of C/C++ code in LabVIEW Dynamic Link Library

The 'Call library function node' function in LabVIEW can be used to call C-code:
Call library function node

"Programmers can reuse C/C++ libraries from past projects by calling them from within a LabVIEW Real-Time application, which simplifies tasks like real-time thread scheduling" - NI in

Note that sometimes the DLL needs to be wrapped when an external DLL will be called in LabVIEW, but the parameters of the function do not easily map to LabVIEW data types. ''A wrapper is a piece of software that provides a compatibility layer to another piece of software. One is often necessary when developing LabVIEW applications because third-party DLLs are typically designed to be accessed from C (or similar low-level languages) and not LabVIEW. Such a DLL may, for example, return pointers or complex data structures which LabVIEW cannot easily handle."
What is a Wrapper DLL and When Do I Need One?

9.2..4 Simulink support in LabVIEW

Author: Gabriel Sherif
Simulink is focusing on dynamic simulation and LabVIEW in the measurement system and data analysis. To utilize these dynamic simulations in LabVIEW the models can be converted into a DLL file format which can then be used in LabVIEW RT.

To convert Simulink file to DLL, see Converting a Simulink Model into a DLL

9.2..5 Model Interface Toolkit

Author: Viktor Aronsson Karlsson
Model interface toolkit is a module for LabVIEW that allows LabVIEW to run models, e.g., Simulink model, when they have been converted to a DLL file format.

Info about Model Interface Toolkit

Examples on running Compiled Simulink Models in LabVIEW and in VeriStand
Short explanation on how Simulink and LabVIEW integrate
How to convert Simulink model to a model the 'Model Interface Toolkit' can run.

9.2..6 LabVIEW Control Algorithms

Author: Love Bridén
The control algorithms of the bicycle consist of (if the documentation is correct) a triple cascaded PID controller. It utilizes three full PID controllers. LabVIEW has built in PID controller functions allowing some of the functionalities of the Python code to be implemented purely in LabVIEW. Information was found in report from Chalmers [40].

PID control in LabVIEW

9.2..7 Formula nodes

Author: Viktor Aronsson Karlsson
Formula nodes is a function in LabVIEW RT which executes text based mathematical equations using a syntax resembling C language

9.2..8 System Exec VI

Author: Viktor Aronsson Karlsson
System Exec VI is a function in LabVIEW that act like the command prompt from Windows.

Its main usage is to open external programs, so running python files might not be the intended function of this feature, but it could work. Further analyses/studies has to be conducted in order to determine how it is compatible with LabVIEW RT

9.3. Conclusion

Author: Viktor Aronsson Karlsson
After much discussion within the group and project owners, it was decided to forgo Python Nodes and find an alternative solution. The new path that was deemed to be followed was to recreate the control code in C and implement it with 'Call Library Function' in LabVIEW. An alternative path that also will be explored is using Simulink code instead of C code.

J How to make *.so files from C code

Author: Viktor Aronsson Karlsson

10.1. Introduction

This document will describe how to create a .so file from C-code. The first two sections tell the reader what programs are needed and what need to be installed for everything to work. The section after that goes through step-by-step the process to convert the C-code into a .so file. The last section is the documentation by the writer of the process of testing different methods used to create a .so file.

10.2. Programs

- C & C++ Development Tools for NI Linux Real-Time 2017, Eclipse Edition
- FileZilla
- LabVIEW

10.3. Requirements

- LabVIEW Real-Time Module
- Java SE 6 – 32bit version (need an Oracle account to download this)

10.4. How to

Steps are adapted from [41] [42]

1. File->New->C Project
2. Shared Library->Cross GCC
3. Check: Debug and Release
4. Add Cross compiler prefix:
   (a) arm-nilrt-linux-gnueabi-
5. Add Cross compiler path or navigate to:
   (a) C:\build\17.0\arm\sysroots\i686-nilrtsdk-mingw32\usr\bin\arm-nilrt-linux-gnueabi
6. Create a Source folder if it doesn't appear
7. Create/import the C and H files
8. Right click the project in project Explorer select properties
9. C/C++ Build
   (a) Builder->Builder type
      i. Choose: Internal builder
10. C/C++ Build -> Setting
   (a) Tool settings
      i. Cross GCC Compiler->Miscellaneous->Other flags
      ii. Add a space then enter:

            A. --sysroot=C:\build\17.0\arm\sysroots\cortexa9-vfpv3-nilrt-linux-gnueabi
        iii. Cross GCC Linker -> Miscellaneous -> Linker flags
        iv. Enter:
            A. --sysroot=C:\build\17.0\arm\sysroots\cortexa9-vfpv3-nilrt-linux-gnueabi

11. C/C++ General -> Paths and Symbols -> Symbols
    (a) Add: kNIOSLinux
12. Click Apply
13. C/C++ Build -> Settings
    (a) Tool Settings
        i. Cross GCC Compiler -> Miscellaneous
            A. Enable "Position Independent Code"
            B. Click Apply
        ii. Cross GCC Compiler
            A. In the Command box, add a space and enter:
            B. -fvisibility=protected
            C. Click Apply
14. Error Parsers
    (a) Enable:
        i. GNU Assembler Error Parser
        ii. GNU gmake Error Parser 7.0
        iii. GNU Linker Error Parser
    (b) Disable everything else
    (c) Click Apply
15. Ok out of settings
16. Right click the project and select Build Project
17. Now you have a *.so file under the Tab "Debug"
18. Insert it into the myrio under the file path using FileZilla:
    (a) \usr\local\lib
19. Restart the myRIO for the myRIO to detect the file.

K Cython

11.1. Overview

Author: Maya Zawawi
Cython is a pragmatic programming language that extend python and enables python to use static types system of C /C++. Also it is compiler that translate the Cython source code into an optimized C/C++ source code, in its turn this source can be translated into python extension module or a standalone executable.

Note that since the Cython translate the code to C language we must have C compiler. The most important advantage is generating C code allows speed improvement because it changes the static typing into a dynamic language. [43]

11.2. Cython compilation pipeline

Author: Maya Zawawi
The compilation process in Cython is two steps process. In the first step the system code is converted into an equivalent optimized and platform independent C/C++ code, and from there on in the second step the C/C++source code is converted into shared library using C/C++ compiler. This shared library is platform depended, the shared library has a dot so extension, on Linux or mac osx and in windows they have pyd extension. The figure 1 shows the pipeline compilation of cython.

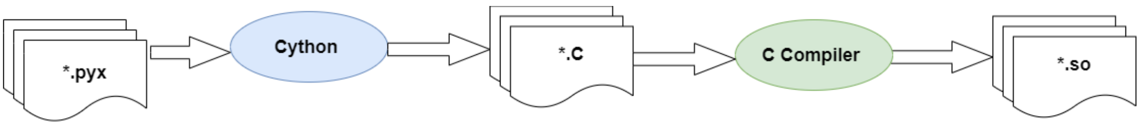


Figure 28: Cython compilation pipeline

11.3. Steps for installing Cython [44]

Author: Maya Zawawi

- Install Anaconda.
- A jupyter notebook will be installed.
- Using jupyter notebook for writing the python code.
- Install Microsoft visual studio (Community version 17)
- Install Microsoft visual studio C++ Build tools (C/C++ Compiler)

''Windows The CPython project recommends building extension modules (including Cython modules) with the same compiler that Python was built with. This is usually a specific version of Microsoft Visual C/C++ (MSVC) - see `https://wiki.python.org/moin/WindowsCompilers`. MSVC is the only compiler that Cython is currently tested with on Windows. A possible alternative is the open source MinGW (a Windows distribution of gcc). See the appendix for instructions for setting up MinGW manually. Enthought Canopy and Python(x,y) bundle MinGW, but some of the configuration steps in the appendix might still be necessary." [45]

- Go to the Windows (C) driver —> Program Files (x86) —-> Microsoft Visual studio 14.0 —-> VC —> CL (This is a compiler of C/C++)
- Go to cmd

1. Typ pip install Cython —> Successfully installed Cython-0.29.24

2. Maybe the user will need to upgrade the pip version 21.2.3, to, version 21.3.1 by using the following command 'C:3100.exe -m pip install –upgrade pip'
3. pip install opencv-python
4. If the user wants to check if he/she installed the right packages type 'pip list' the user should have (Cython, opencv-python, numpy, and pip)

11.4. Steps for converting Python code into C code through Cython

Author: Maya Zawawi

- Go to Anaconda prompt —> typ ' conda activate main ' This is the environment I choose to work in to avoid working in a root folder.
- ' jupyter notebook'
- The python program could not identify the saved files. That the user had created.
- The user could not run Chalmers's codes, the following messages appeared : " Chalmers's LabVIEWcode.xlsx is not UTF-8 encoded", the package "pyvisa " is missing.
- The pyvisa has been downloaded "Successfully installed pyvisa-1.11.3"
- After installing "pyvisa" package the code could not be opened at all.

11.5. Workflow

Author: Viktor Aronsson Karlsson

1. Took Mayas suggestion and didn't install jupyter, since it created problems, and used pycharm and anaconda instead
2. Followed this tutorial on youtube: `https://www.youtube.com/watch?v=Ic1oE6SEOBs`
3. Managed to convert the Python file to C, and the code executed correctly
4. Tried following the steps from "" to create a *.so file, that will be used in LabVIEW
5. This attempt ended in a failure
6. By looking in the created C file it could be concluded that the created C code were difficult for humans to understand
7. After some research online this tutorial was found: `https://www.youtube.com/watch?v=mXuEoqK4bEc`
8. In this video it could be observed that a .so file is created
9. The tutorial was performed in Linux, so Linux was installed in VirtualBox
10. The tutorial was followed and a .so file was created
11. Inserting the file into the myRIO and executing it in LabVIEW
12. This also ended in a failure
13. The conclusion for now is that Cython is not the solution.

11.6. Conclusion

Author: Viktor Aronsson Karlsson

After following two tutorials, one for windows and the other for Linux, where both ended in undesired results. Cython is an excellent tool when used to speed up the execution of python code. Unfortunately, due to the fact that the generated C code is constructed in a confusing manner and very hard for a human to understand, made the last step of transferring the code into LabVIEW near impossible. When Cython generates the C code it changes the names of the functions in the code, which must be known in order to use the in LabVIEW. In conclusion Cython is not useful for this project since the function name is needed, but if there is a way to know the new function name, Cython could be used.

L The execution time for the LabVIEW code

Author: Viktor Aronsson Karlsson

Related documents

Execution time.xlsx

12.1. Description

From each timed loop in LabVIEW, which is every gathering and control loop, the time interval of the latest iteration can be extracted. By utilising this function, each time interval can be stored after its the execution. These stored values can then be logged after the execution is done in a TDMS file, which can then be opened in excel for calculating the average value of the execution time for each loop.

12.1..1 Test with the duration of 5 minutes

The program was running for about 5minutes, where each loop stored its loop period. When the program was stopped, all timing data was logged in a TDMS file. The periods were then averaged and can be seen in the table and graph below.

	No. Samples	Average Time(us)	Worst Case Time(us)
IMU	77.5K	1137	1300
Hall	103K	210	250
Encoder	103K	111	200
GPS	1K	17	30
E-Stop	109K	112	150
Steering	104K	208	250

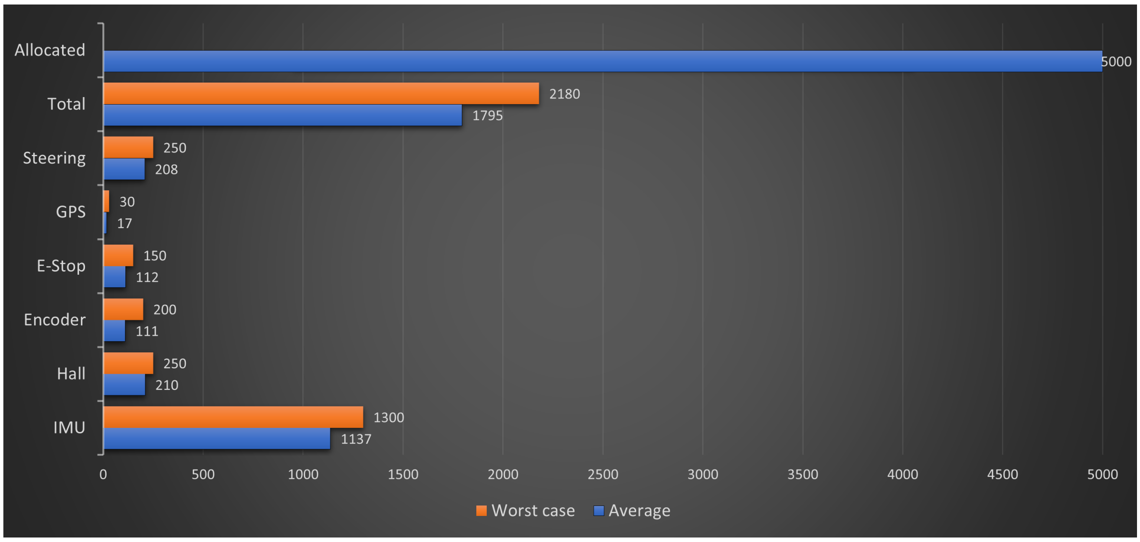


Figure 29: The execution time for the Lab VIEW code

M Project Description

Author: Viktor Aronsson Karlsson

<table>
<tr><td>Project Name</td><td colspan="3">Autobike</td></tr>
<tr><td>Background</td><td colspan="3">From the year 2017 Chalmer University and Mälardalens University have collaborated for the purpose of creating an autonomous bicycle. Where its mission is to be used in the car industry for the purpose of validating the crash avoidance algorithms of self-driving cars.

In earlier years of the Autobike project, the self-driving aspect of the project have been achieved. Now the projects’ goal is to improve the dependability of system by transferring the existing python code into a LabVIEW interface for increased performance. The end goal is the creation of a product that both researchers and users at Volvo Cars can comfortability use with minimal knowledge about how Autobike operates in the background.</td></tr>
<tr><td>Project owner/ Sponsor</td><td colspan="3">Jonas Sjöberg and Henrik Falk</td></tr>
<tr><td>Project Manager</td><td colspan="3">Viktor Aronsson Karlsson</td></tr>
<tr><td>Effective Goal</td><td colspan="3">Construction of an LabView interface for the python code.
The validation and verification of the existing functionalities, in both the old and new software.
The validation and verification of the hardware components.
Comparison between the Red and Black bicycle, they should be similar.</td></tr>
<tr><td>Project goal</td><td colspan="3">The construction of an autonomous bicycle that will be used in an educational purpose, training the crash avoidance of self-driving cars.</td></tr>
<tr><td>Time for project</td><td colspan="2">Start: 7th of September 2021</td><td>Stop: 10th of December 2021</td></tr>
<tr><td>Priorities</td><td>Time %</td><td>Resources %</td><td>Quality %</td></tr>
<tr><td>Budget</td><td colspan="3">---</td></tr>
<tr><td>The material received by</td><td colspan="2">Viktor Aronsson Karlsson,
Love Bridén,
Maya Zawawi,
Gabriel Sherif</td><td>Date: 9th of September 2021</td></tr>
<tr><td>Appendixes</td><td colspan="3">Bachelor Projects from Chalmers: Electronics Group, Mechatronic Group and Plastic Bike Group.
MDH reports: Year 2017, 2018 and 2019</td></tr>
</table>

Figure 30: Project description, created at the beginning of the project

N Scope of the project

Author: Viktor Aronsson Karlsson

14.1. Document description

The overall goal of the Autobike project is the creation of an autonomous bicycle that have the purpose of training the crash avoidance algorithms in autonomous cars. Since the end goal of this student project is ambitious, which also means the time needed to finish the project will exceed the available time. That is why the project is divided into several smaller sub-project that focuses on only a few aspects at a time.
The rest of this document will describe the scope of this sub-project in general terms, more detailed depictions of the scope can be found in their relevant documentation. The document will also contain the limitations of the project, areas that are outside of the scope of this sub-project.

14.2. Related documents

Project description, see Appendix M

14.3. Scope

ID	Scope	Priority
1	Creation of a LabVIEW interface for the Python code	High
2	Validation of the existing software	High
3	Validation of the new software	High
4	Comparison of performance between new and old software	High
5	Validation of the new hardware	High
6	Validation of the existing functionalities (on the new hardware and software)	High
7	Reorganisation of the hardware inside of the compartment	Low
8	Find and document the limitations of the system	Medium
9	Comparison and compatibility between the Red and Black bicycle	Medium

14.4. Limitations

ID	Limitation	Reasoning
1	Alterations to the existing algorithms.	No further development is needed at this point
2	Replacement of the hardware	The existing hardware is sufficient for the correct functionality of the system. A strong reasoning is needed for the change of a component.
3	Weatherproofing and -validation of the Autobike	Validation from external events is outside the scope of this project.
4	Reinforcement of the box for added shock resistance	Validation from external events is outside the scope of this project.
5	Alterations to the existing python code.	Only minor changes are allowed. Depending on the LabVIEW architectures demand.
6	Validation and verification in an outdoor environment.	When the validation and verification can be performed indoors, it should be performed indoors.
7	External disturbances during validation and verification	Validation and verification will be performed in an optimal environment.

O Requirements management plan for project AutoBike 2021

15.1. Introduction

Author: Love Bridén

This document describes the requirements management plan for project Autobike carried through at Mälardalen University during the autumn of 2021. The document is structured as follows: subsection 15.1..1 is a general overview of the project, section 16.3. describes the stakeholders of the project, section 16.4. provides information of the goals of the stakeholders, section 15.4. describes how the requirements for the project are elicited, section 15.5. describes the formatting of the requirements, section 15.6. contains information describing the requirements review process, section 15.7. describes how traceability is ensured among the requirements, section 15.8. describes how changes in the requirements are dealt with and lastly, section 15.9. describes how changes in this document are dealt with throughout the project.

15.1..1 Project overview

Author: Love Bridén and Viktor Aronsson Karlsson

The overall goal of this research project is to modify a self-driving bicycle with the purpose of testing the crash avoidance ability of self-driving cars designed by Volvo Cars. The so called Autobike is the red bicycle produced by Chalmers University of Technology in a previous iteration of this project. The overall goal of the current year's project is to increase the dependability of the bicycle mostly concerning maintainability and fault forecasting to increase the availability of the bicycle. To reach this, optimization and verification efforts of some of the different functionalities of the autonomous bicycle designed and manufactured by Chalmers University of Technology were conducted. This involves, among others, the conversion from pure python code to a LabVIEW wrap-around for use in a National Instruments (NI) myRIO device. The goal is also (partly) reached by the sensors and actuators already present in the previously mentioned autonomous bicycle to be verified and validated for the purpose(s) of the project.

15.2. Stakeholders

Author: Love Bridén and Viktor Aronsson Karlsson

The stakeholders for research project Autobike are:

- Mälardalen University
- Chalmers University of Technology
- Volvo Cars
- Veoneer
- Autoliv
- Cycleurope
- AstaZero

To some extent the environment is a stakeholder, however, this iteration of the project is small enough in scale to have a negligible effect on the environment regardless of what possible outcomes of the project are reached. The students partaking in this project are also stakeholders for it as the project is part of their education received at Mälardalen University, however, their goals are fulfilled by fulfilling the goals of the other stakeholders of the project.

15.3. Stakeholder Goals

Author: Love Bridén and Viktor Aronsson Karlsson

The general goals of the project as a whole is to have an autonomous bicycle that visually looks and behaves like a traditional bicycle for use in validation of safety systems designed by Volvo Cars. For this iteration of the project the stakeholder goals are not as concrete. The goal regarding the general resemblance of a traditional bicycle still stands for this iteration. Other than that, reduction of the complexity of the Autobike and enabling the usage of a myRIO device as the microcontroller for the Autobike is of highest interest for the stakeholders during this iteration of the project. Following are general stakeholder goals of this iteration of the project. Properly elicited stakeholder goals are elicited in appendix P.

- The AutoBike shall generally resemble a traditional bicycle.
- The Beagle-Bone present on the AutoBike shall be swapped for an NI myRio device.
- LabVIEW code shall be designed to act as an interface between the NI myRIO device and the python code designed by Chalmers University.
- The actuators of the Autobike shall be verified and validated against the purpose of the Autobike.
- The sensors of the Autobike shall be verified and validated against the purpose of the Autobike.
- The box containing most of the electronics of the Autobike shall be swapped for a smaller box (or smaller boxes).
- The electronics inside the box of the Autobike shall be rearranged and attached to the new electronics box.
- Integration testing for the system as a whole.
- A built in test for the AutoBike shall be designed in LabVIEW.
- Two separate user interfaces shall be designed, one for researchers and one for end users.

As for the students, their goals is to pass this course and along the way learn how to work in a project as well as increasing the quality of their engineering capabilities. This is assumed (by the students) to be reached by striving, to the best of their capabilities, to meet the stakeholder goals of this project.

15.4. Requirements elicitation

Author: Love Bridén

For eliciting requirements, primarily, a model based approach is used. First off a black box perspective will be used, that is, the system is viewed as nothing more than a black box to highlight the interactions of the system with the environment via the interfaces of the system. Using this model, the stakeholder level requirements can be elicited by evaluating how the system shall respond to inputs at said interfaces. When the stakeholder level requirements are elicited from the stakeholder goals presented in the previous section (and properly presented in appendix P) a second set of models will be designed. This second set of models would be an internal block diagram to highlight the subsystems of the Autobike along with their interactions via their interfaces. Using these internal block diagrams the system and subsystem level requirements could be elicited. These system requirements are elicited from the stakeholder requirements and the subsystem requirements are elicited from the system level requirements. To complement the diagrams created context diagrams, use case diagrams and a state diagram could be used to further elicit requirements and thereby complete the detailed architecture of the system. Lower level requirements are, of course, verified against the higher level requirements/goals of the system. Both formal and informal verification efforts will be conducted to evaluate if what was intended to be implemented actually was implemented. The formal verification efforts are meant to evaluate if the requirements for the Autobike are implemented, the informal verification efforts will be conducted in a similar fashion, however, the informal efforts will not be based on the requirements but the functionality that the requirements present. If a verification effort is failed some modifications has to be made. First off further testing and investigating has to take place in order to isolate the fault(s). When the fault has been identified the system has to be redesigned to tackle this, this includes both the system and the requirements describing the systems. If changes are made, the procedure described in section 15.8. will be followed.

15.5. Requirements specification

Author: Love Bridén

For the requirements of the system to be properly documented and structured the requirements shall contain the following: a trace to above level requirement/goal, a reference to the author of the requirement, a comment or rationale (if necessary), a unique identifier and a trace to lower level requirements.

The requirements shall be written following the syntaxes below:

- The <stakeholder> shall be able to <capability>.
- The <System name> shall be able to <system response>.
- When <optional precondition/trigger>, the <system name> shall <system response>.
- While <in a specific state> the <system name> shall <system response>.

[46]

The structure of the requirements document will be the following:

ID	Description	Comment	Acceptance criteria	Backwards trace	Forward trace	Author	Review status

The requirements shall be structured in sections, that is, according to; stakeholder requirements, system level requirements, subsystem requirements and component level requirements.

15.6. Requirements review

Author: Love Bridén

In order to assure quality within the requirements, both as a set and individually, a review of the requirements will take place. The review will be conducted by all members of the project in order to limit effects of biases. The review process will be supported by a review protocol which is found in **reqspec**. The review protocol will contain information about what requirement or set of requirements that are changed, if no change is presented the requirement is accepted as is. Each change shall be motivated in the protocol. If the requirements contain errors they will be changed and if a change takes place the child requirements of the changed requirements has to be checked if they have to change as well until all child requirements of a changed requirement are eventually changed or at least checked and decided to be left unchanged. Some of the different types of errors that the requirements will be analysed against are related to; redundance, ambiguity, preciseness, verifiability, feasibility and legality of the requirements. The status the requirements could have are awaiting review, accepted and rejected. The contents of this section are influenced by [47].

15.7. Traceability

Author: Love Bridén and Viktor Aronsson Karlsson

All documents produced during the project must contain the name of the author and if there are multiple authors each section has state who contributed to what section. All derived requirements must have traceability to their top level requirements/goals through the use of reference numbering. The requirements document must contain at least one column that have the explicit purpose of traceability. To ease the navigation of the requirements documents all requirements should have a trace both to the parent and the child requirement/goal. In the case of top/bottom goal/requirement the parent/child goal/requirement does not exist and such information is therefore not included in the documents.

15.8. Dealing with changes in the requirements

Author: Love Bridén and Viktor Aronsson Karlsson

All changes in official documents has to be documented and stored for review in the projects' Teams channel, where later, a third party can come in and validate the changes. In an optimal situation the third party would have been an entity from outside the project but in this project, due to the limited amount of resources, the third party will be another group member (different from the author of the document). After a requirement is changed all requirements originating from the changed requirement has to be evaluated if they could be left unchanged, has to change or even be removed. A change in a requirement is accepted when all child requirements to a changed requirement has been evaluated and, if necessary, changed. If child requirements change, the child requirements of those requirements has to be evaluated until either all child requirements of a changed requirement are left unchanged or all child requirements are changed after a change to the parent requirement.

When a change has been approved and all affected documents have been altered, the change need to be documented in an excel file with the name "Revision History - All documents" under the correct excel tab located in the Teams channel.

15.9. Dealing with changes in the requirements management plan

Author: Love Bridén and Viktor Aronsson Karlsson

Changes to this document will be handled in utmost care, since changes that take place in this documents can have cascading effects throughout the whole project. This requires a clear trace between the different documents to produce. In the case that a change is needed, the procedure will follow as in section 15.8.. The difference will be that instead of a third party validating the change, a meeting with all group members shall be arranged to discuss the proposed change. The proposed change is not accepted until all affected documents/requirements/goals are evaluated and eventually corrected (or even removed) to fit the proposed change.

P Requirements Specification Document for project AutoBike 2021

Author: Love Bridén

16.1. Introduction

This document describes the requirements for project AutoBike 2021. The requirements are sectioned in stakeholder requirements, system-level requirements, subsystem-level requirements and lastly, component-level requirements. The syntaxes for the requirements are as follows:

- <Stakeholder> shall be able to <Capability>.
- The <System name> shall be able to <System response>.
- When <Optional precondition/trigger>, the <System name> shall <System response>.
- While <In a specific state> the <System name> shall <System response>.
- The <Property> of <System name> shall be <Property of property>.

The structure of the requirements will be the following:

ID	Description	Comment	Acceptance criteria	Backwards trace	Forward trace	Author	Review status

This information is stated in appendix O.

In the ID column of the requirements the type of requirements are highlighted according to:

- Stakeholder Requirement: STRQ###.
- System-level Requirement: SYRQ###.
- Subsystem-level Requirement: SURQ###.
- Component-level Requirement: CRQ###.

The signatures to fill in the requirements matrix are as follows:

- Love Bridén : LB

16.2. Statement of needs

The overall aim of the project as a whole is to produce autonomous bicycles. These bicycles are to be used to test the safety systems of autonomous cars produced by Volvo Cars. For this iteration of the project a functioning autonomous bicycle is the starting point, the need is that it should be modified. The modifications desired by the stakeholders are mostly within the subject of dependability.

16.3. Stakeholders

The stakeholders for the project as a whole are:

- Mälardalen University
- Chalmers University
- Volvo Cars
- Veoneer
- Autoliv
- Cycleurope
- AstaZero

(The information is presented in appendix O.)

However, the main stakeholders for this specific iteration of the project are:

- Mälardalen University, researchers and students
- Chalmers University, researchers.
- Volvo Cars, test personnel.

16.4. Stakeholder Goals

The stakeholder goals for this specific iteration of the project are as follows:

- SG01: Swap the BeagleBone Black for a myRIO device.
- SG02: Create wrap-around software to be able to use the existing software in the swapped microcontroller.
- SG03: Modify the hardware layout to reduce the volume of the electronics box harnessing most of the electronics of the bicycle.
- SG04: Modify the hardware architecture for optimization purposes.
- SG05: Increase the modularity of the bicycle (to ease maintenance, repairs and system alterations).
- SG06: Create Built In Tests (BIT) for fault detection.
- SG07: Create user interfaces for the Autobike.
- SG08: Analyse if reliable speed data can be obtained without using the custom made speed sensor.
- SG09: Analyse the reliability of the sensors of the bicycle.
- SG10: Analyse the reliability of the actuators of the bicycle.
- SG11: Swap the motor controller for a VESC 6 MkV.
- SG12: Data from all the sensors (excluding GPS) shall be obtained by the myRIO device with a frequency of 100 Hz.
- SG13: GPS data shall be obtained by the myRIO device with a frequency of 10 Hz.
- SG14: Actuator data shall be sent from the myRIO device with a frequency of 100 Hz.

16.5. Stakeholder Requirements

Due to space limitations in this document the requirements are presented in appendix Q.
SG08, SG09 and SG10 did not result in any stakeholder requirements, instead activities from said goals are presented in documents regarding the verification and validation of the Autobike. Figure 31 was used to elicit stakeholder requirements from the stakeholder goals.

16.6. System Requirements

When starting this iteration of the project some parts of the architecture were given. For the context of STRQ001 to STRQ003 (presented in appendix Q) the given components were the myRIO, the Inertial Measurement Unit (IMU), the HALL speed sensor, the steering angle encoder, the steering motor controller, the forward motor controller and the finder. The two motor controllers controls the actuators of the bicycle. Among the sensors (GPS excluded) (the four remaining components) three of them are "passive", in the sense that they always output data without having the microcontroller to "ask" for the sensor to send the data. The IMU, however, is not passive in that sense, it requires the microcontroller to "ask" for the IMU to send data, and so, twice in order to receive all data required. To conclude, communications between all the previously mentioned components are one way, while the communications for the IMU are two ways (data is sent both to and from the sensor) and with two iterations. The sampling frequency shall be 100Hz, that gives each iteration a period of 10ms. This results in the following equation:

Equation 1: $10ms \leq Calculate + 3Read + 6Send + 2Sample$

Budgeting half of the time for calculating actions, that is 5ms, leaves 5ms for the rest of the activities:

Equation 2: $5ms \leq Calculate$

Equation 3: $5ms \leq 3Read + 6Send + 2Sample$

These activities are 11 in total, thus one 11:th of the remaining time can be allocated towards each activity.

Equation 4: $454\mu s \leq Read$

Equation 5: $454\mu s \leq Send$

Equation 6: $454\mu s \leq Sample$

Above, "Calculate" is the total time elapsed from that the myRIO has received all relevant data for the sampling iteration until it is ready to send data towards the actuators. "Read" is the activity of reading the data output by a "passive" sensor. "Send" is the activity of sending a data message through one of the communication protocols used in the Autobike (e.g. Inter-Integrated Circuit (I^2C) or Universal Asynchronous Receiver/Transmitter (UART)). Sample is the activity of the IMU to sample data about the orientation of the component in three axes and the angular velocities in these three axes.

Some of the stakeholder requirements are not derived into system requirements, instead, some of the stakeholder requirements are repeated as system requirements and derived in the lower level requirements. This applies to STRQ004 and STRQ005.

Regarding SYRQ012 and SYRQ015 the GPS module takes quite some time to sample all relevant GPS data using the antenna. In total the GPS module samples 9 different samples that are later sent to the myRIO device through a UART interface. The sampling frequency of the GPS data shall be 10Hz according to the stakeholder of the project. That means that each iteration is allowed up to 100ms to be completed. Budgeting a tenth of the iteration time for sending the messages (10ms) leaves 90ms for the sampling of all data. Since nine different messages has to be sent in order to send all relevant GPS data 1.11ms is budgeted for each message. Since 9 samples has to be made 10ms is the maximum allowed time for one sample. Figures 32, 33 and 34 were used to elicit system-level requirements from the stakeholder requirements.

16.7. Subsystem Requirements

Some of the system level requirements are copied to the subsystem requirements, in the case a derivation is necessary, requirements will be derived between lower level sets of requirements. If no derivation is necessary, the requirement is copied until it reaches the correct level of requirements. Figures 32, 33 and 34 were used to elicit subsystem-level requirements from the system-level requirements.

16.8. Component Requirements

Figures 32 and 33 were used to elicit component-level requirements from the subsystem-level requirements.

16.9. Requirements Review

The requirements review took place with all group members. Each requirement were reviewed, both individually and as a set. The review checked the requirements against the properties presented in [47]. Following are descriptions of what requirements changed, if no comment is present below for a specific requirements, it was accepted as is.

In STRQ003 the words "proper action" were changed for "control command", this sparked changes in SYRQ003, SURQ007 and CRQ004 as they are child requirements to STRQ003. STRQ007 was not singular, thus it was changed to two requirements by adding STRQ013 and modifying STRQ007. The trace from STRQ007 to SYRQ008 was changed to become a trace from STRQ013 to SYRQ008. In SYRQ001, the word "relevant" was removed for clarification purposes, affected child requirements were changed as well. In SURQ004 "...state of the..." was changed for "...the current state of the...". In SURQ005 and SURQ006 the word "relevant" was removed for clarification purposes. In SURQ016, "us" was changed to "ms" as it was accidentally erroneously typed.

Regarding STRQ011 (and child requirements) "its current functionalities" is referring to the functionalities of the Autobike when received. In other words, reducing the amount of components of the Autobike should not affect the functionalities of the Autobike. This means that only redundant components can be removed. Regarding SYRQ006 "standardised interfaces" means that the software interfaces should be similar. This is meant to ease alterations and maintaining efforts of the software of the Autobike. Of course different software interfaces might need to be designed differently, in that case it is not practical to design them similarly.

16.10. Diagrams

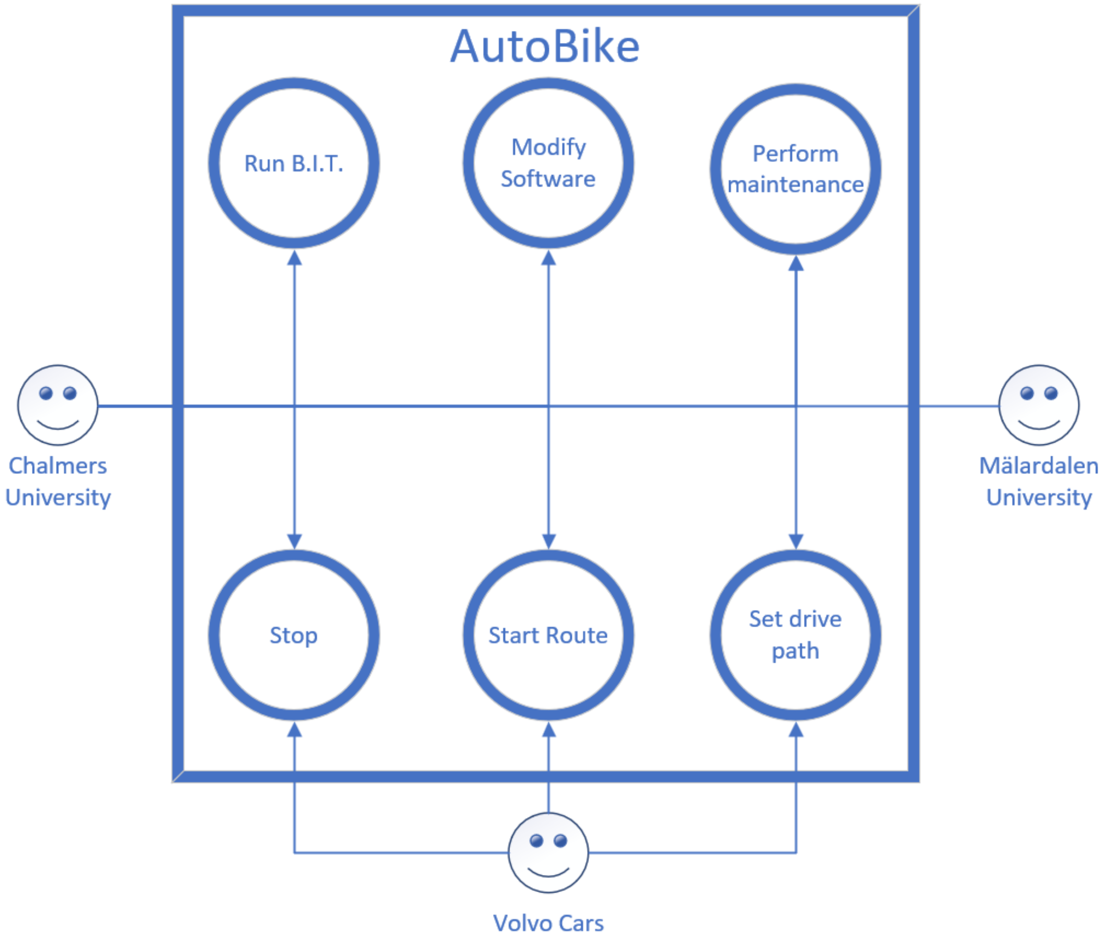


Figure 31: Use Case Diagram for the Autobike. The rectangle represents the Autobike and the stakeholders are presented as the smiley faces. The circles in the rectangle are functionalities that stakeholders desire and the arrows are interactions. This diagram was used to elicit stakeholder requirements from their goals.

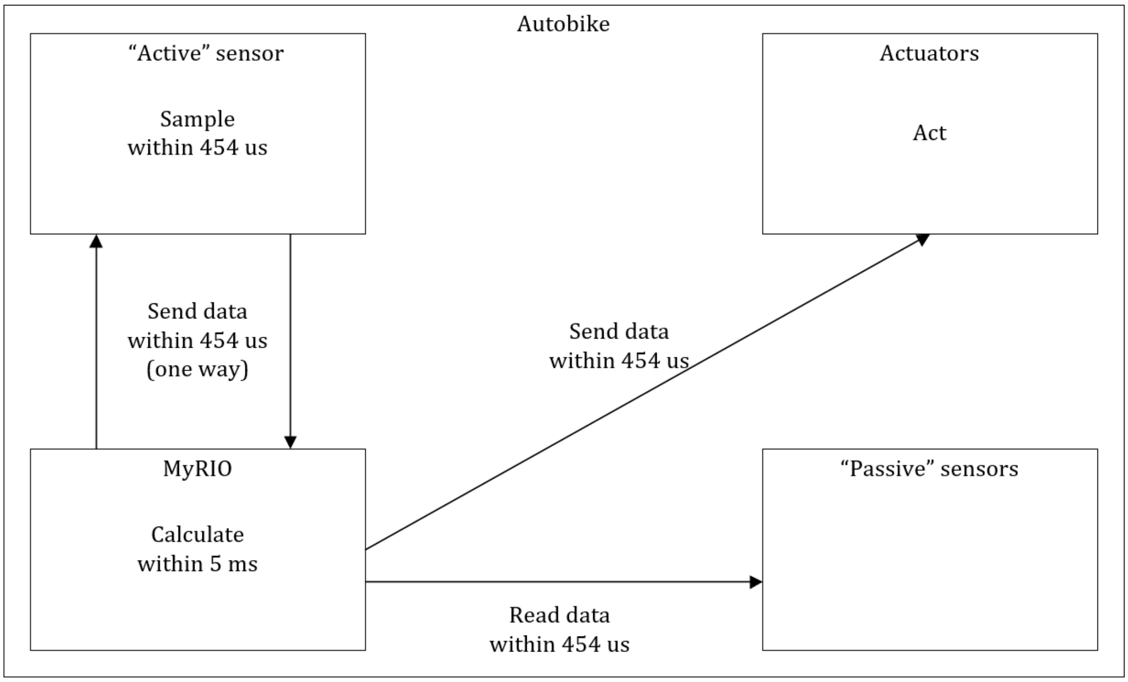


Figure 32: Diagram of the data transfers and other time consuming activities for the Autobike. The squares are systems/subsystems/components, the arrows represents data transfer or gathering, the times mentioned for the activities are the maximum allowed time for the activity. The "Active" sensor require the microcontroller to "ask" for its data. The "Passive" sensors always output their data, the microcontroller has to read the data to collect it. The Active sensor (IMU), as said, requires the microcontroller to "ask" for it to send its data, and so two data transfers and one sampling effort has to be made for the microcontroller to receive any data. Two types of data are collected by the IMU, the total amount of sampling and data transfer efforts to gather all relevant data from the IMU during one sampling iteration is two sampling efforts and four data transfer efforts.

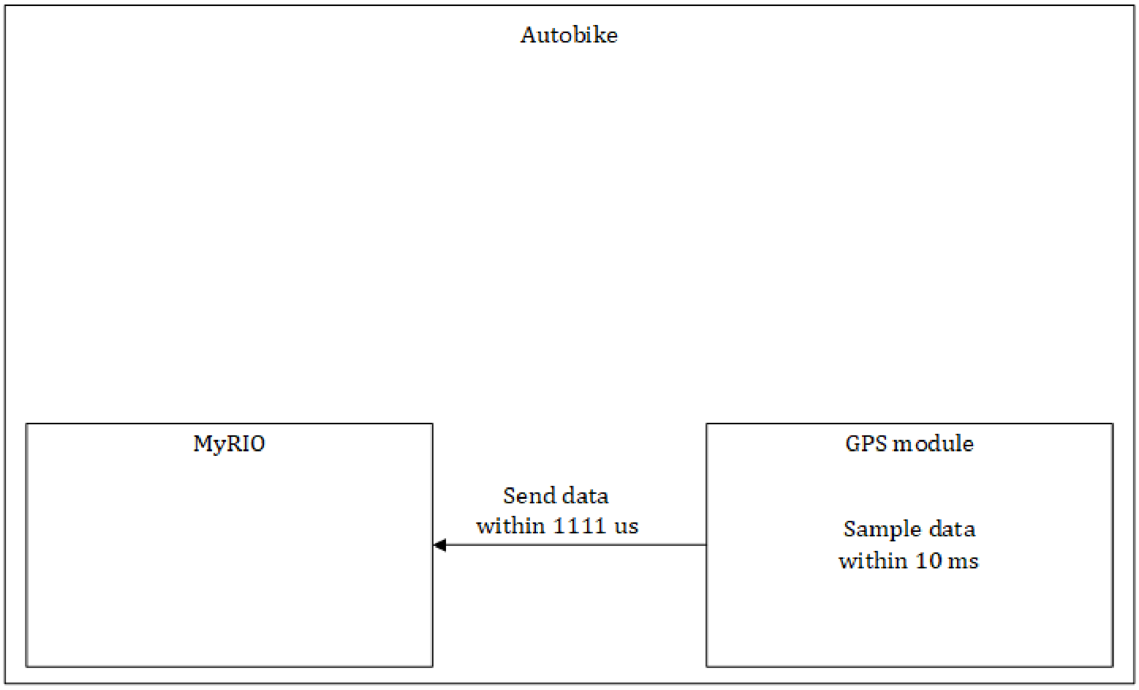


Figure 33: Diagram of the data transfer and sampling of the GPS data to the microcontroller of the Autobike. For all relevant GPS data to be sampled nine sampling efforts has to be conducted. Nine data transfer efforts has to be made in order to send all data to the microcontroller.

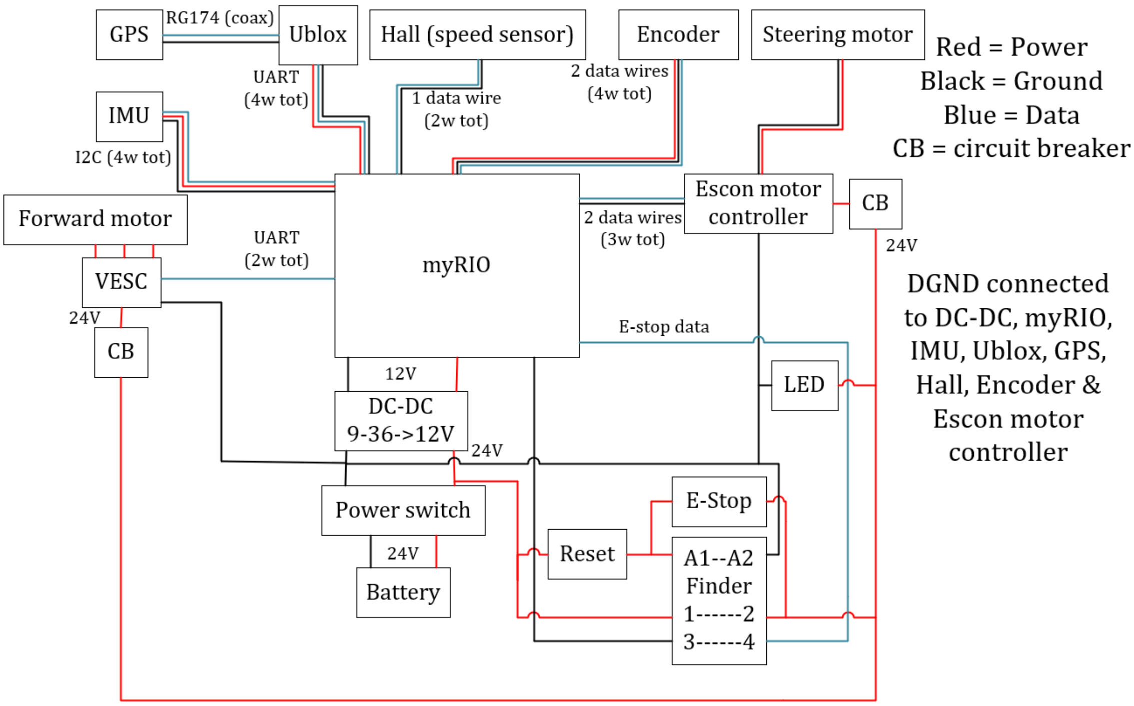
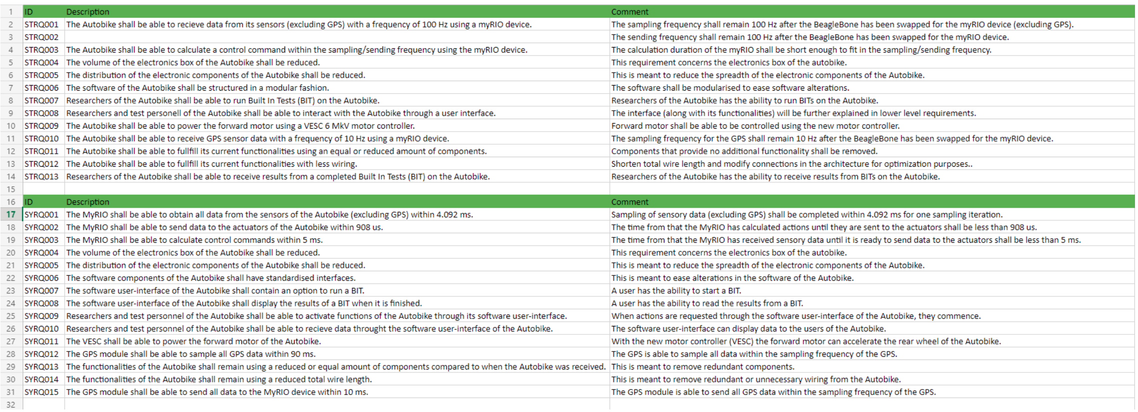


Figure 34: Internal block diagram of the Autobike

Q Requirements

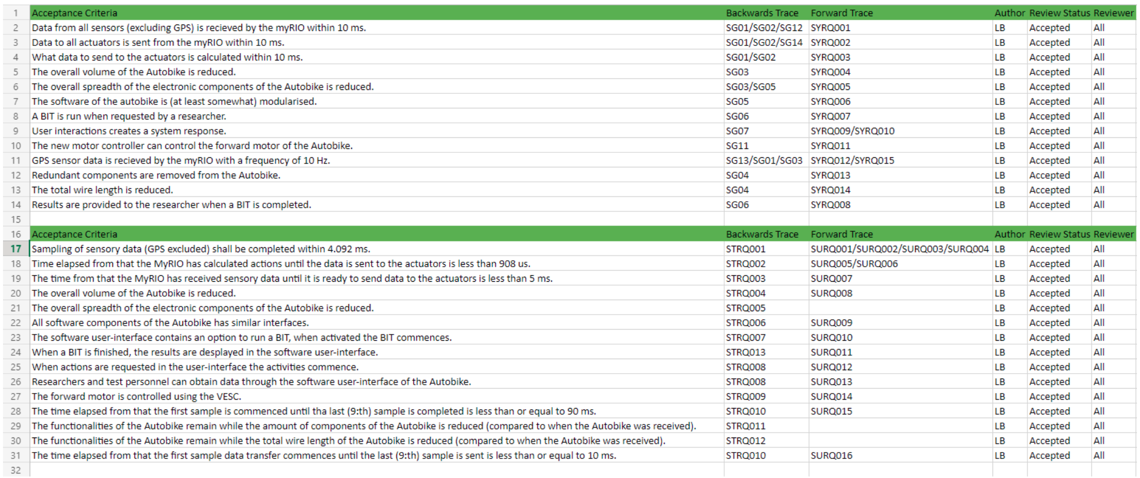
Author: Love Bridén

Below are pictures of the requirements in their original document. The rows highlighted in the column on the very left of the pictures are meant to be used to be able to pair the pictures according to the rows of the original document.



1	ID	Description	Comment
2	STRQ001	The Autobike shall be able to recieve data from its sensors (excluding GPS) with a frequency of 100 Hz using a myRIO device.	The sampling frequency shall remain 100 Hz after the BeagleBone has been swapped for the myRIO device (excluding GPS).
3	STRQ002		The sending frequency shall remain 100 Hz after the BeagleBone has been swapped for the myRIO device.
4	STRQ003	The Autobike shall be able to calculate a control command within the sampling/sending frequency using the myRIO device.	The calculation duration of the myRIO shall be short enough to fit in the sampling/sending frequency.
5	STRQ004	The volume of the electronics box of the Autobike shall be reduced.	This requirement concerns the electronics box of the autobike.
6	STRQ005	The distribution of the electronic components of the Autobike shall be reduced.	This is meant to reduce the spreadth of the electronic components of the Autobike.
7	STRQ006	The software of the Autobike shall be structured in a modular fashion.	The software shall be modularised to ease software alterations.
8	STRQ007	Researchers of the Autobike shall be able to run Built In Tests (BIT) on the Autobike.	Researchers of the Autobike has the ability to run BITs on the Autobike.
9	STRQ008	Researchers and test personell of the Autobike shall be able to interact with the Autobike through a user interface.	The interface (along with its functionalities) will be further explained in lower level requirements.
10	STRQ009	The Autobike shall be able to power the forward motor using a VESC 6 MkV motor controller.	Forward motor shall be able to be controlled using the new motor controller.
11	STRQ010	The Autobike shall be able to receive GPS sensor data with a frequency of 10 Hz using a myRIO device.	The sampling frequency for the GPS shall remain 10 Hz after the BeagleBone has been swapped for the myRIO device.
12	STRQ011	The Autobike shall be able to fullfill its current functionalities using an equal or reduced amount of components.	Components that provide no additional functionality shall be removed.
13	STRQ012	The Autobike shall be able to fullfill its current functionalities with less wiring.	Shorten total wire length and modify connections in the architecture for optimization purposes..
14	STRQ013	Researchers of the Autobike shall be able to receive results from a completed Built In Tests (BIT) on the Autobike.	Researchers of the Autobike has the ability to receive results from BITs on the Autobike.
15			
16	ID	Description	Comment
17	SYRQ001	The MyRIO shall be able to obtain all data from the sensors of the Autobike (excluding GPS) within 4.092 ms.	Sampling of sensory data (excluding GPS) shall be completed within 4.092 ms for one sampling iteration.
18	SYRQ002	The MyRIO shall be able to send data to the actuators of the Autobike within 908 us.	The time from that the MyRIO has calculated actions until they are sent to the actuators shall be less than 908 us.
19	SYRQ003	The MyRIO shall be able to calculate control commands within 5 ms.	The time from that the MyRIO has received sensory data until it is ready to send data to the actuators shall be less than 5 ms.
20	SYRQ004	The volume of the electronics box of the Autobike shall be reduced.	This requirement concerns the electronics box of the autobike.
21	SYRQ005	The distribution of the electronic components of the Autobike shall be reduced.	This is meant to reduce the spreadth of the electronic components of the Autobike.
22	SYRQ006	The software components of the Autobike shall have standardised interfaces.	This is meant to ease alterations in the software of the Autobike.
23	SYRQ007	The software user-interface of the Autobike shall contain an option to run a BIT.	A user has the ability to start a BIT.
24	SYRQ008	The software user-interface of the Autobike shall display the results of a BIT when it is finished.	A user has the ability to read the results from a BIT.
25	SYRQ009	Researchers and test personnel of the Autobike shall be able to activate functions of the Autobike through its software user-interface.	When actions are requested through the software user-interface of the Autobike, they commence.
26	SYRQ010	Researchers and test personnel of the Autobike shall be able to recieve data throught the software user-interface of the Autobike.	The software user-interface can display data to the users of the Autobike.
27	SYRQ011	The VESC shall be able to power the forward motor of the Autobike.	With the new motor controller (VESC) the forward motor can accelerate the rear wheel of the Autobike.
28	SYRQ012	The GPS module shall be able to sample all GPS data within 90 ms.	The GPS is able to sample all data within the sampling frequency of the GPS.
29	SYRQ013	The functionalities of the Autobike shall remain using a reduced or equal amount of components compared to when the Autobike was received.	This is meant to remove redundant components.
30	SYRQ014	The functionalities of the Autobike shall remain using a reduced total wire length.	This is meant to remove redundant or unnecessary wiring from the Autobike.
31	SYRQ015	The GPS module shall be able to send all data to the MyRIO device within 10 ms.	The GPS module is able to send all GPS data within the sampling frequency of the GPS.
32			

Figure 35: ID, description and comment of the stakeholder and system requirements of this iteration of the Autobike project.



1	Acceptance Criteria	Backwards Trace	Forward Trace	Author	Review Status	Reviewer
2	Data from all sensors (excluding GPS) is recieved by the myRIO within 10 ms.	SG01/SG02/SG12	SYRQ001	LB	Accepted	All
3	Data to all actuators is sent from the myRIO within 10 ms.	SG01/SG02/SG14	SYRQ002	LB	Accepted	All
4	What data to send to the actuators is calculated within 10 ms.	SG01/SG02	SYRQ003	LB	Accepted	All
5	The overall volume of the Autobike is reduced.	SG03	SYRQ004	LB	Accepted	All
6	The overall spreadth of the electronic components of the Autobike is reduced.	SG03/SG05	SYRQ005	LB	Accepted	All
7	The software of the autobike is (at least somewhat) modularised.	SG05	SYRQ006	LB	Accepted	All
8	A BIT is run when requested by a researcher.	SG06	SYRQ007	LB	Accepted	All
9	User interactions creates a system response.	SG07	SYRQ009/SYRQ010	LB	Accepted	All
10	The new motor controller can control the forward motor of the Autobike.	SG11	SYRQ011	LB	Accepted	All
11	GPS sensor data is recieved by the myRIO with a frequency of 10 Hz.	SG13/SG01/SG03	SYRQ012/SYRQ015	LB	Accepted	All
12	Redundant components are removed from the Autobike.	SG04	SYRQ013	LB	Accepted	All
13	The total wire length is reduced.	SG04	SYRQ014	LB	Accepted	All
14	Results are provided to the researcher when a BIT is completed.	SG06	SYRQ008	LB	Accepted	All
15						
16	Acceptance Criteria	Backwards Trace	Forward Trace	Author	Review Status	Reviewer
17	Sampling of sensory data (GPS excluded) shall be completed within 4.092 ms.	STRQ001	SURQ001/SURQ002/SURQ003/SURQ004	LB	Accepted	All
18	Time elapsed from that the MyRIO has calculated actions until the data is sent to the actuators is less than 908 us.	STRQ002	SURQ005/SURQ006	LB	Accepted	All
19	The time from that the MyRIO has received sensory data until it is ready to send data to the actuators is less than 5 ms.	STRQ003	SURQ007	LB	Accepted	All
20	The overall volume of the Autobike is reduced.	STRQ004	SURQ008	LB	Accepted	All
21	The overall spreadth of the electronic components of the Autobike is reduced.	STRQ005		LB	Accepted	All
22	All software components of the Autobike has similar interfaces.	STRQ006	SURQ009	LB	Accepted	All
23	The software user-interface contains an option to run a BIT, when activated the BIT commences.	STRQ007	SURQ010	LB	Accepted	All
24	When a BIT is finished, the results are desplayed in the software user-interface.	STRQ013	SURQ011	LB	Accepted	All
25	When actions are requested in the user-interface the activities commence.	STRQ008	SURQ012	LB	Accepted	All
26	Researchers and test personnel can obtain data through the software user-interface of the Autobike.	STRQ008	SURQ013	LB	Accepted	All
27	The forward motor is controlled using the VESC.	STRQ009	SURQ014	LB	Accepted	All
28	The time elapsed from that the first sample is commenced until tha last (9:th) sample is completed is less than or equal to 90 ms.	STRQ010	SURQ015	LB	Accepted	All
29	The functionalities of the Autobike remain while the amount of components of the Autobike is reduced (compared to when the Autobike was received).	STRQ011		LB	Accepted	All
30	The functionalities of the Autobike remain while the total wire length of the Autobike is reduced (compared to when the Autobike was received).	STRQ012		LB	Accepted	All
31	The time elapsed from that the first sample data transfer commences until the last (9:th) sample is sent is less than or equal to 10 ms.	STRQ010	SURQ016	LB	Accepted	All
32						

Figure 36: Acceptance criteria, backwards trace (to parent goals or requirements), forward trace (to child requirements), author of the requirement, review status and the reviewer of the stakeholder and system requirements of this iteration of the Autobike project.

33	ID	Description	Comment
34	SURQ001	The MyRIO shall be able to obtain all data from the IMU within 2.724 ms.	The IMU data is received within the sampling frequency of the sensors with time to spare for sampling of the other sensors.
35	SURQ002	The MyRIO shall be able to obtain all data from the speed sensor within 454 us.	The speed sensor data is received within the sampling frequency of the sensors with time to spare for sampling of the other sensors.
36	SURQ003	The MyRIO shall be able to obtain all data from the encoder within 454 us.	The encoder data is received within the sampling frequency of the sensors with time to spare for sampling of the other sensors.
37	SURQ004	The MyRIO shall be able to obtain all data about the current state of the E-stop button within 454 us.	The E-stop state data is received within the sampling frequency of the sensors with time to spare for sampling of the other sensors.
38	SURQ005	The MyRIO shall be able to send data to the VESC (forward motor controller) within 454 us.	Actuator data is sent to the VESC within the budgeted time for actuator data transfer with time to spare for the data transfer to the ESCON.
39	SURQ006	The MyRIO shall be able to send data to the ESCON (steering motor controller) within 454 us.	Actuator data is sent to the ESCON within the budgeted time for actuator data transfer with time to spare for the data transfer to the VESC.
40	SURQ007	The MyRIO shall be able to calculate control commands within 5 ms.	The time from that the MyRIO has received sensory data until it is ready to send data to the actuators shall be less than 5 ms.
41	SURQ008	The volume of the electronics box of the Autobike shall be reduced.	This requirement concerns the electronics box of the autobike.
42	SURQ009	The software components of the Autobike shall have standardised interfaces.	This is meant to ease alterations in the software of the Autobike.
43	SURQ010	The software user-interface of the Autobike shall contain an option to run a BIT.	A user has the ability to start a BIT.
44	SURQ011	The software user-interface of the Autobike shall display the results of a BIT when it is finished.	A user has the ability to read the results from a BIT.
45	SURQ012	Researchers and test personnel of the Autobike shall be able to activate functions of the Autobike through its software user-interface.	When actions are requested through the software user-interface of the Autobike, they commence.
46	SURQ013	Researchers and test personnel of the Autobike shall be able to recieve data throught the software user-interface of the Autobike.	The software user-interface can display data to the users of the Autobike.
47	SURQ014	The VESC shall be able to power the forward motor of the Autobike.	With the new motor controller (VESC) the forward motor can accelerate the rear wheel of the Autobike.
48	SURQ015	The GPS module shall be able to sample each GPS data sample within 10 ms.	The GPS is able to sample all relevant data within the sampling frequency of the GPS.
49	SURQ016	The GPS module shall be able to send each GPS data sample to the MyRIO within 1.111 ms.	The GPS module is able to send all relevant GPS data within the sampling frequency of the GPS.
50			
51	ID	Description	Comment
52	CRQ001	The IMU shall be able to sample once within 454 us.	The IMU is able to sample within the sampling/sending frequency of the Autobike.
53	CRQ002	The IMU shall be able to transfer one data message to the MyRIO within 454 us.	The IMU is able to communicate with the MyRIO within the sampling/sending frequency of the Autobike.
54	CRQ003	The MyRIO shall be able to transfer one data message to the IMU within 454 us.	The MyRIO is able to communicate with the IMU within the sampling/sending frequency of the Autobike.
55	CRQ004	The MyRIO shall be able to calculate control commands within 5 ms.	The time from that the MyRIO has received sensory data until it is ready to send data to the actuators shall be less than 5 ms.
56	CRQ005	The volume of the electronics box of the Autobike shall be reduced.	This requirement concerns the electronics box of the autobike.
57	CRQ006	The GPS module shall be able to sample each GPS data sample within 10 ms.	The GPS is able to sample all relevant data within the sampling frequency of the GPS.

Figure 37: ID, description and comment of the subsystem and component requirements of this iteration of the Autobike project.

33	Acceptance Criteria	Backwards Trace	Forward Trace	Author	Review Status	Reviewer
34	Time elapsed from that the MyRIO has started requesting data from the IMU until all relevant data is obtaine by the MyRIO is less than or equal to 2.724 ms.	SYRQ001	CRQ001/CRQ002/CRQ003	LB	Accepted	All
35	Time elapsed from that the MyRIO has started sampling the speed sensor data until the sampling is finished is less than or equal to 454 us.	SYRQ001		LB	Accepted	All
36	Time elapsed from that the MyRIO has started sampling the encoder data until the sampling is finished is less than or equal to 454 us.	SYRQ001		LB	Accepted	All
37	Time elapsed from that the MyRIO has started sampling data about the state of the E-stop button until the sampling is finished is less than or equal to 454 us.	SYRQ001		LB	Accepted	All
38	Time elapsed from that the MyRIO has started sending data to the VESC until the data transfer is finished is less than or equal to 454 us.	SYRQ002		LB	Accepted	All
39	Time elapsed from that the MyRIO has started sending data to the ESCON until the data transfer is finished is less than or equal to 454 us.	SYRQ002		LB	Accepted	All
40	The time from that the MyRIO has received sensory data until it is ready to send data to the actuators is less than 5 ms.	SYRQ003	CRQ004	LB	Accepted	All
41	The overall volume of the Autobike is reduced.	SYRQ004	CRQ005	LB	Accepted	All
42	All software components of the Autobike has similar interfaces.	SYRQ006		LB	Accepted	All
43	The software user-interface contains an option to run a BIT, when activated the BIT commences.	SYRQ007		LB	Accepted	All
44	When a BIT is finished, the results are desplayed in the software user-interface.	SYRQ008		LB	Accepted	All
45	When actions are requested in the user-interface the activities commence.	SYRQ009		LB	Accepted	All
46	Researchers and test personnel can obtain data through the software user-interface of the Autobike.	SYRQ010		LB	Accepted	All
47	The forward motor is controlled using the VESC.	SYRQ011		LB	Accepted	All
48	The GPS module is able to sample one GPS data sample within 10 ms.	SYRQ012	CRQ006	LB	Accepted	All
49	The time elapsed from that a sample data transfer commences until it is sent to the MyRIO is less than or equal to 1.111 ms.	SYRQ015		LB	Accepted	All
50						
51	Acceptance Criteria	Backwards Trace	Forward Trace	Author	Review Status	Reviewer
52	Time elapsed from that the IMU commences to sample until the sampling effort is completed is less than or equal to 454 us.	SURQ001		LB	Accepted	All
53	Time elapsed from that the IMU commences to send one data message to the MyRIO until the message is sent is less than or equal to 454 us.	SURQ001		LB	Accepted	All
54	Time elapsed from that the MyRIO commences to send one data message to the IMU until the message is sent is less than or equal to 454 us.	SURQ001		LB	Accepted	All
55	The time from that the MyRIO has received sensory data until it is ready to send data to the actuators is less than 5 ms.	SURQ007		LB	Accepted	All
56	The overall volume of the electronics box of the Autobike is reduced.	SURQ008		LB	Accepted	All
57	The GPS module is able to sample one GPS data sample within 10 ms.	SURQ015		LB	Accepted	All

Figure 38: Acceptance criteria, backwards trace (to parent goals or requirements), forward trace (to child requirements), author of the requirement, review status and the reviewer of the subsystem and component requirements of this iteration of the Autobike project.

R Verification and Validation Management Plan for project AutoBike 2021

Author: Maya Zawawi

18.1. Introduction

The main purpose of this document is to outline the generic verification and validation testing activities and methodologies, which can be performed through the AutoBike project lifecycle in both Hardware and software from high to low levels.
The Agile methodology prefers to be used in verification and validation for more efficient testing. Using Agile methodology can provide iteratively and incrementally test throughout the project life cycle,which can allow more flexible and reliable tester. But due to some delays the waterfall methodology can be efficient to achieve the desired result as well, so some of these tests will be done after the hardware and software be accomplished.

18.2. Scope and Background

AutoBike project is a cooperative project between Chalmers and Mälardalens Universities since 2017. To develop an automated bicycle, that will be tested on the field in order to reduce road accidents against cyclists. The stakeholders of this project are Volvo Cars, Veoneer, Autolive, and Cycleurope.

18.3. Terminology and Notation

A MoSCow table will help with defining the importance of each test in order to accomplish the Verification and validation.**1**

1. Must have: Top level tasks or procedures that must be tested and fulfilled.
2. Should have: The tester is strongly recommended to verify and approve the described process or function.
3. Could have: Not top priority for this phase of the project.

18.4. V model for the system

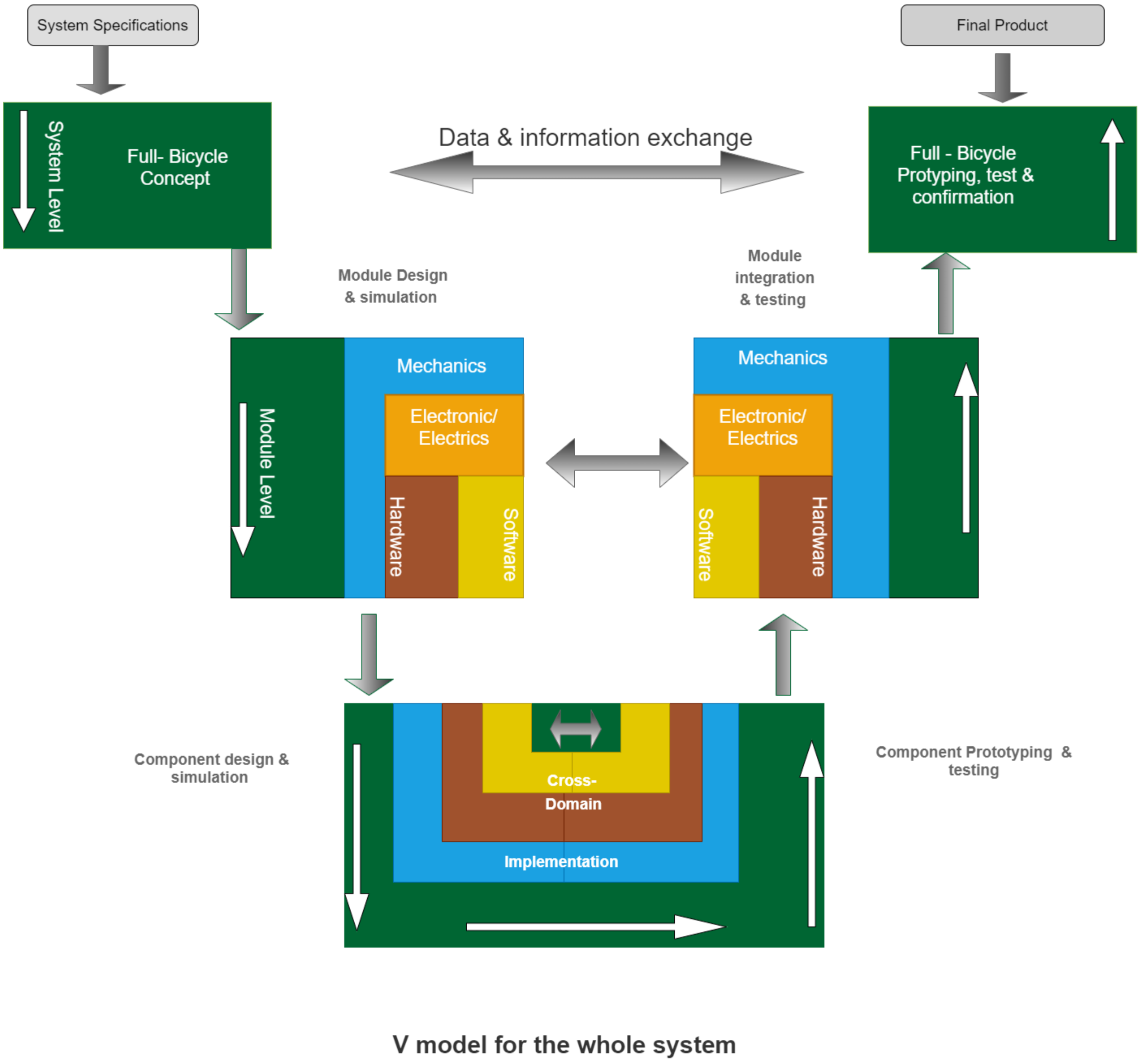


Figure 39: V model of the Auto-Bike

18.5. Software Verification and Validation

The main mission of the software verification validation is to conduct continuous testing during each life-cycle phase of the project which can guarantees that all requirements of different classification & levels will be fulfilled, against the hardware functionality, sub-system execution, any possible user, and any other interfaces that will interact with the software, which in turn leads to an optimal end product that can fulfil the primary requirements. **2**

18.6. Hardware Verification and Validation

The verification will ensure that the system meets the requirements and specifications in the development phase with regular tests if it is possible, to ensure that the system performs correctly according to the initial requirements. While the validation will ensure that the system as a final product is meeting the operational needs of the user and achieving its intended purpose. **3**

18.7. Recommended V& V phases and tasks

4

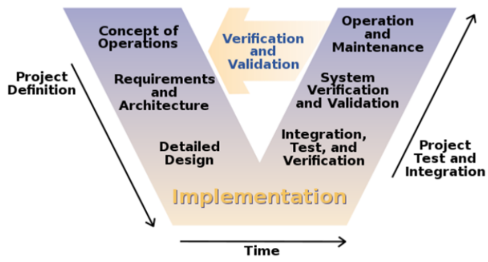


Figure 40: Phases and tasks of V model**4**

18.7..1 Concept

The main concept of the project and the top-level requirement to be tested at the end of the project is to update and develop the red bicycle, that received by MDH university, and achieve a comparison and make sure that there is a kind of compatibility between the Red and Black bicycle in Chalmers University.

18.7..2 Requirement Analysis (System Definition)

During this phase, the requirements of the system will be elaborated as completely and accurately as possible in terms of the system's software and hardware. In order to achieve this task, a regular meetings with the stakeholders shall be held for more understanding and to gather the requirements depending on the stakeholder's needs and expectations. During this phase the requirements will be tested to verify that they are correct and testable, traceable, written with a correct syntax, and describe the needed functional and non-functional performance to assure that they are compliance with acceptance criteria.

18.7..3 System Design

The mutually agreed requirements will be the output of the previous phase and will be used to analyse those requirements to conduct a feasibility studying and design the complete system. and the output will be the high-level system design. Which in our case is to modify and developed the autonomous red bicycle to imitate both the visual appearance and the behave of a normal bicycle. The team will verify that the auto-bike can drive forward, can brake, change velocity and the path of driving. After this phase we will have the complete picture of various components, like user, software, hardware interfaces, and data base.

18.7..4 System Implementation

In this stage of verification and validation of the system, all the technical decisions will be taken like which sensors and actuators are suitable, which programming language will be used, communications protocol and many more. The main task in this phase for the Auto-Bike' project will be to try to minimise the hardware size by eliminating the unnecessary equipment, swap the Beagle-Bone for an NI myRio device, create a LabVIEW code to control the sensors and actuators through installing a python code within the NI myRio.

18.7..5 Project test

This phase consists of three levels :
Unit test: we can say that this is the white box testing to test individual components like the hall sensor the encoder of the bicycle and other actuators, the main aim is to validate unit component with its performance and test the correctness of the isolated code of the software such as data flow

testing, and control .
Integration test: This is the second level of testing, this test will be conducted to test the software and hardware internal modules and internal communications. The main aim of this test is to find the defects at the time of interaction of integrated unites or components. The output of this test will be checking the correctness of communications between different modules.
System test: will be conducted just before the product delivering to test the entire system functionality including all the internal and external components.
The AutoBike will be controlled by a LabVIEW code to send signals from sensors in order to run the actuators, as a result the bicycle will start to run in specific path.
Tools:
Function generator
LabVIEW environment
Stationary computer or laptop
Multimeter
The following table shows the steps to be followed in order to complete V& V process:

ID	Test Scenario	Test Case	Pre-condition	Expected Result	Actual Result	Pass/Fail	Author

18.7..6 Verification &Validation Documentation

The final report will includes all the tests conducted on system's software and hardware and the tests results , the errors faced the development process, and how they have been eliminated if so. All the documents that need to be validated shall be identified and assigned a number. such as the requirements and the safety documents. This section will provide assistance to understand what is needed to complete the validation process. The following table shows the steps of tests that will be conducted.

References

[1] MoSCow table
https://sv.overleaf.com/read/ybzkxgbjnjnp

[2] https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

[3] https://www.pharmaceuticalonline.com/doc/how-to-write-an-effective-validation-master-plan-000

[4] https://www.labvantage.com/elements-of-a-successful-validation-plan/

[5] Fig2:\https://en.wikipedia.org/wiki/V-Model#V-Modell

S System Safety Management Plan

Author: Gabriel Sherif

19.1. Introduction

The primary purpose of the System Safety Management Plan (SSMP) is to serve basis as a framework for Autobike safety, to be able to manage the safety risks, and consistently identify and reduce the potential for an accident. Throughout this project, the result will be documented and decision making. This paper presents the primary step and basis methods appropriately to identify hazards before they occur and analyze them. Note. All diagrams and definitions are based on a general system safety process, adapted by the book [5], but modified to the Autobike project.

19.2. Background

The project aims to create an autonomous bicycle, and it has been a collaborative project between Chalmers University and Mälardalens University since 2017. The goal of this project is to improve the dependability of the system. An essential safety work needs to produce a System Safety Management Plan, which requires in-course guidelines.

19.3. Definition

19.3..1 Safety

Safety can be defined as being safe and being protected from danger or injury. Security can be referred to as customs control to achieve an acceptable level of risk of recognized hazards. There, the concept of system security is about risks that endanger the environment or human lives in many cases. However, this project is about creating an autonomous bicycle. This cycle is based on an idea that should be run automatically and cannot be defined as a general environment or people's lives in danger. This document must be followed up and regularly renewed, with the aim of "thinking before" being prepared to minimize negative consequences for unforeseen events for both individuals and the Autobike projects.

19.3..2 Hazard

Hazard can be defined as a condition or a cause of harm that can lead to loss of equipment, environmental damage, or in the worst case, to death. People who work in the system can stand and contribute to most accidents, but the hazard does not mean a fault. Danger can come in many different forms that can be a source of a potential failure.

19.3..3 Risk

Risk description is based on what is to be done in the AutoBike project. There are many risk decisions that should be included in the AutoBike project, but it will focus on two main areas, namely policy and procedure.

The policy will convey what to do in different situations, and the procedure will describe how to do it. There, all routines will be developed to consider the risks of operational maintenance and cabinet safety to check human error on a checklist as well as the controls that the required inspection items protect against any mistakes that may occur in operation. Risk analysis from hazard to risk can be an operational failure and another set of underlying unfavourable workplace conditions. Active failures are often cited in probable causes that may be the result of human activity. Hazard identification will be used to determine the aspects of the system and environments that constitute hazardous conditions. Dangers are often incorrectly identified as consequences, and hazards are not events. Hazards do not arise, but they do exist in the environment. All workplace conditions or sets of conditions may be a danger or not. Data analytics must concentrate on being facts so that accurate inference about their potential to cause system errors can be achieved.

19.4. Hazard Severity Categories

19.4..1 Catastrophic

Catastrophic is a significant risk, event or accident with extensive destruction, it counts as a Disaster if one of the below happened.

- Any event that may cause death
- Any event that may cause permanent total disability
- Any event that may cause serious personal injury
- Any event that may cause loss of system
- Any event that may irreversible significant environmental impact

19.4..2 Critical

e.g.e Bringing the system down for some days because of damaging equipment

- Any event that may cause severe injury
- Any event that may cause loss of mission-critical hardware equipment
- Any event that may cause loss of mission-critical hight-drive equipment
- Any event that may cause permanent partial disability
- Any event that may cause occupational illness that results in hospitalization of at least three personnel
- Any event that reversible significant environmental impact

19.4..3 Minor

e.g., Loss of associated electronic control or monitoring signals

- Any event that may cause minor injury
- Any event that may cause minor system damage
- Any event that may occupational illness resulting in 1 lost workday, or more
- Any event that may reversible moderate environmental impact

19.4..4 Negligible

e.g., lose commercial power, causing a shutdown of AutoBike

- Any event that does not result in injury
- Any event that does not result in system damage
- Any event that does not result in cycle damage
- Any event that does not affect the mission
- Any event that does not occupational illness does not result in a lost workday
- Any event that may have a minimal environmental impact

19.5. Hazard Probability Categories

The hazard probability categories have levels from A to F, there F have low, and A have extremely high levels.

Level	Hazard category Frequency description	Specific Individual Item
A	Frequent	Likely to occur in the lift of the AutoBike
B	Probable	It will occur several times in the AutoBike project
C	Occasional	Likely to occur sometime in the life of the AutoBike
D	Remote	Unlikely, but possible to occur in the life of AutoBike
E	Improbable	So unlikely that it can be assumed occurrence may not be experie
F	Eliminated	Potential hazards and later eliminated

19.6. Assessment Matrix for Hazard Risk

The hazard risk assessment matrix is based on hazard probability categories and hazards probability categories. Here the hazard risk index (HRI) is a combination of severity and probability levels. With different categories, can we decide? Later this matrix will help the risk evaluation phase, e.g., where a danger must be addressed with a high degree of severity and probability.

Hazard Category Frequency

Hazard category	frequency	Catastrophic	Critical	Minor	Negligible
Frequent	x > 10E-1	A1	A2	A3	A4
Probable	10E-1> x > 10E-2	B10	B2	B3	B4
Occasional	10E-2> x > 10E-3	C1	C2	C3	C4
Remote	10E-3> x > 10E-6	D1	D2	D3	D4
Improbable	10E-6 > x	E1	E2	E3	E4
Eliminated	10E-7 > x	F	F	F	F

19.7. Risk Decision Criteria

The below diagram shows the Risk decision Criteria and Hazard Risk Index (HRI), where the categories show which HRIs belong to which risk decision criteria. For example, risk level A1 (with catastrophic-Probability and frequent-Severy) which are in unacceptable group (matrix down), the A1 have both high level in severity categories and high-level probability categories, then must rectify immediately and stop the operations. In this project we will not receive any risk categories in this level. Another example C2 (with critical severity categories and occasional probability), in this case the risk are in level 3, undesirable, here the upper manager take decision to accept or reject.

Risk Decision Criteria	HRI
Acceptable I	C4, D4, E4
Acceptable II	E1, E2, D3, E3, A4, B4
Undesirable	D1, C2, D2, B3, C3
Unacceptable	A1, B1, C1, A2, B2, A3
Eliminated	F, Unacceptable, undesirable

Risk Decision Criteria

- Acceptable I: Acceptable without review
- Acceptable II: Acceptable with management review
- Undesirable: upper-management decision to accept or reject
- Unacceptable: stop operation and rectify immediately
- Eliminated: take action for unacceptable risk

19.8. Safety Requirement

The safety requirement for the Autobike project is those requirements for safety purposes. They are related to any other requirement, the Autobike project requirement. During the different phases of the project with a different method inside hazard analysis, the autobike project requirement can at a high level be specified and produced by the safety manager, which the condition will take with the project leader and discussed. After the discussion, the approved requirements will remain, and continue to work with them to find solutions or change or delete them. There will be a formal answer as to why a requirement has been removed.

19.9. Risk Analysis Schematic

19.9..1 System Safety Process

The system security process is a systematic and comprehensive way to ensure that the system's security is considered when designing the system. The system description is a system design function that involves the entire AutoBike project. It is used to develop processes. Based on AutBike projects to ensure that potentially dangerous conditions are identified and controlled. The effectiveness of controls monitors the emergence of new hazards - where risk management is applied at two primary levels: process and assignment levels. To design a new system process or procedure - a system safety process is used for all changes in the AutoBike project - to identify potential hazards in the project. The figure below is an essential visual representation of a method for the system safety process.

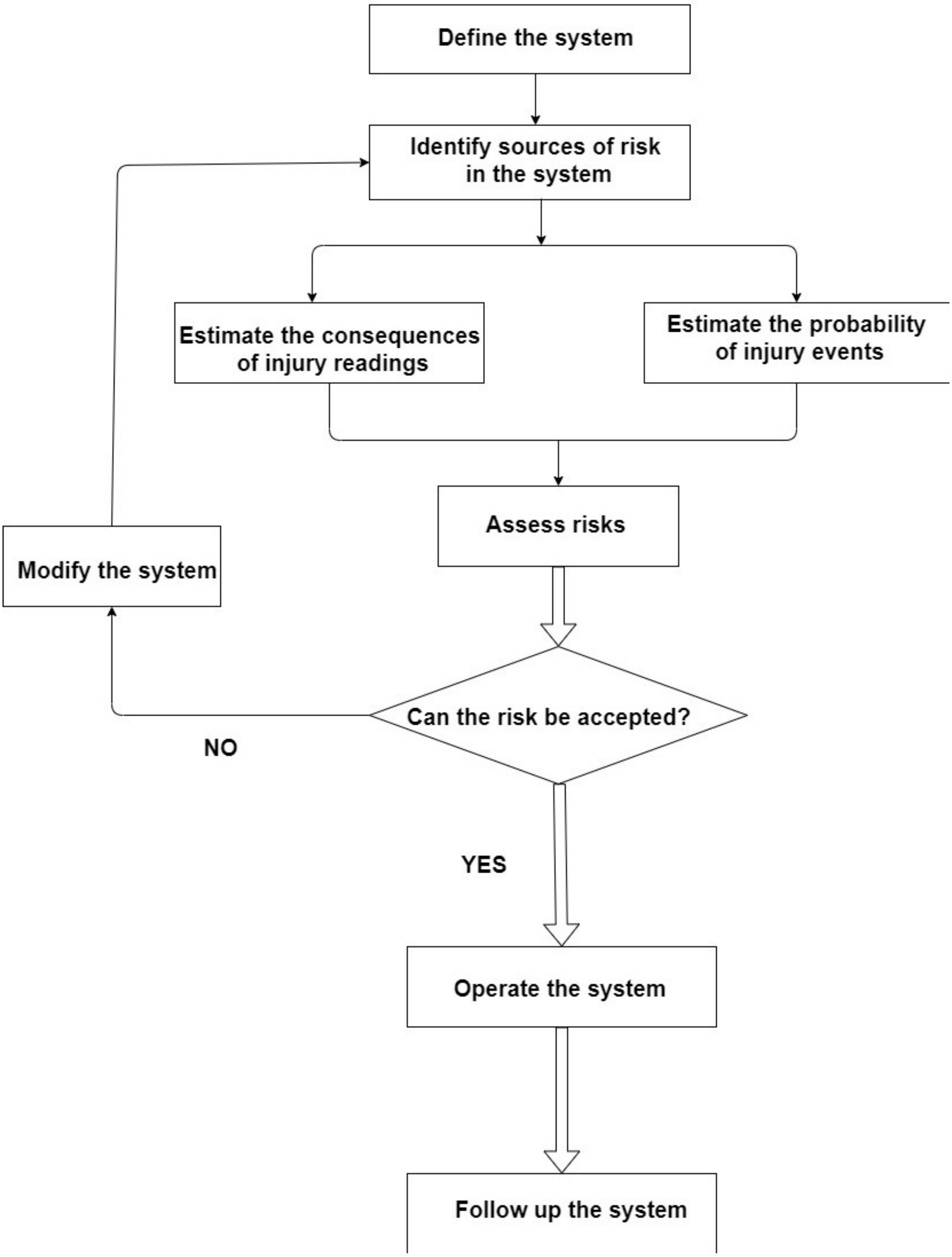


Figure 41: Risk Analysis Schematic

19.9..2 Define system

The first part is defining systems, where all hazards in the system are limited, and the analysis will be done within narrow limits. The purpose of this is to know the exact level desired for the safety analysis to then be able to write all definitions of danger levels and identify and describe all aspects. Before the concept of hazards can be touched upon, it is of great value to know the limits and limitations of the analysis. Should mention that this project is not like a large project in a large company. The system also involves connections between different components such as the software, hardware part, people, and environment. To make formal definitions of the different levels of criticality, one must know what level of security is desired for the analysis. This step is essential because the new step of the system depends on the previous steps. The risk analysis will be affected if the system is not complete and well-defined. But we cannot be sure for all risk are taken in the right way - can this be because everyone has a different view of a risk.

19.9..3 Identify sources of risk in the system

Identifying sources of risk in the system is next step. Where the possible hazard to the system that has been searched has been found. For this part, a - this part a Hazard and operability study, Hazop, diagram will be made to identify all the dangers with their consequences. There, the safety manager to analyse and to detect the risk go through one by one in each component. However, Hazop and this system will not guarantee to find all the hazards in the system. After this step and with the help of the Hazop diagram, the safety manager will be able to build a Preliminary Hazard List (PHL), which is similar to a brainstorm. PHL will identify most hazards one step further from the Hazop chart. The PHL will be rewritten and then the unnecessary risk will also remove from this list.

19.9..4 Estimate the probability of injury events and estimate the consequences of the injury readings

Estimating the probability of damage events and estimating the consequences of the Damage readings are the next steps in the system that analyse further in this part. This is done for each hazard to look at them and examine the consequences and probabilities. Through this step, the security manager will know the values to monitor a hazard in the system or remove the hazard from the system with an explanation of why.

19.9..5 Assess risk

Can the risk be accepted as an additional step to control effects on the identified hazards, it is a verification process to ensure that the dangers are removed or that the risk is spacious enough to proceed with it without changing anything. Here, the head of security will focus mainly on high-risk hazards concerning the severity and probability aspects. In this part, the security manager will discuss with project leaders to move forward where project managers will decide on a specific hazard with a high or low level of risk. There may be project leaders approving the risk of moving forward or taking the time to address the danger.

19.9..6 Can the risk be accepted

Can the risk be accepted as an additional step to control effects on the identified hazards, it is a verification process to ensure that the dangers are removed or that the risk is spacious enough to proceed with it without changing anything. Here, the head of security will focus mainly on high-risk hazards concerning the severity and probability aspects. In this part, the security manager will discuss with project leaders to move forward where project managers will decide on a specific hazard with a high or low level of risk. There may be project leaders approving the risk of moving forward or taking the time to address the danger.

19.9..7 Modify the system

This step is a control measure, where if the risk is not accepted, the system must be modified so that the danger does not continue and that it no longer exists. Here the risk level is further investigated, and the system will be modified with the help of project leaders and other project members. Safety devices must be implemented to prohibit the development of a hazard, and it will serve as an additional function to the system.

19.9..8 Operate the system

Using the system is the step in the system where the associated risk that has been identified has risk levels of zero. As previously mentioned, the system will not be completely secure, and there will be some risk left and further associated. Therefore, project leaders and the security chief must be involved in moving forward with risk and accepting it or taking action. If the project manager believes that the risk will not affect the system, it will be necessary to justify why the risk is accepted. The security manager will look at it again and go through the risk analysis schedule to ensure that changes comply with all parts.

19.9..9 Follow the system

It is the last step in the system that has zero risk levels on all identified risks. And the safety manager has look at risk for this project.

T Safety

20.1. System safety process

Author: Gabriel Sherif

20.2. Preliminary Hazarad List (PHL)

The Preliminary Hazard List (PHL) is a significant step list of anything that can go wrong - for the safety manager to ensure that all potential hazards have been identified and taken care of and achieve system security - for the Auto bike project. Input is primarily needed: a subsystem compote list and a project system description document, which declares all subsystems, dependencies, schedule, system functions, etc.

Must of all hazards have been predicted and identified. For this process, a HAZOP diagram was made to identify all hazards for all components. The HAZOP diagram went through all the pieces one by one, focusing on finding hazards and consequences. There, you can identify most of the hazards and produce measures, except that you cannot evaluate all the identified risks.

Note: It cannot be mentioned that PHL considers all the Hazard, and some of them should remove and separete (below). All part come from system safety management plan and Risk analysis Schematic, with some changes.

20.2..1 Before operatition Preliminary Hazarad List

- Loss of software program
  - Loss control over the bike
  - Loss the motors
  - Loss the power
- Loss of communication
  - Loss WIFI connection
  - Loss SW
- Loss of power
  - Battery start fire
  - Battery start explosion
- Loss Hardware
  - Loss control over the bike

20.3. Hazard Analysis

- Collision with another object
  - Stop the bike
  - loss control
  - HW or SW stop
  - The bike falls
  - Late action for direction changes
  - An incorrect direction rates
  - To fast or to low speed

20.3..1 Hazarad Analysis

To better understand PHL and then analyses the identified hazards, further analysis is needed - the hazard analysis. With the hazard analysis, one can try to find effects and underlying causes for all the hazards. Where a risk can have several underlying causes, and with the help of the hazard analysis, you can analyse them one by one. You may focus on a single danger for a long time and increase to find solutions to it miss other underlying causes. It can be said that all hazards have two identifying reasons, either hull damage or hull explosion.

20.4. Risk Evaluation

After PHL and hazard analysis focused on the effects and underlying causes, we need a Risk evaluation that looks at difficulty and probability aspects. All the severity is affected by hazards and the underlying cause of the risks. And the probability is affected by the underlying causal factor. All the evaluations come from assumptions concentrated on the project. Where it is essential to identify different causes of the hazard and then categorize each cause.

Then, based on the combination of the degree of difficulty and the probability, one can determine how much attention and care is required for each danger. Using the HRI index, all hazards can be categorized into four different criteria groups. Each category has its degrees of severity and shows the need to implement risk controls and risk acceptance.

- Level1: Acceptable without review
- Level2: Acceptable with management review
- Lavel3: Undesirable, the manager decisions needs
- Level4: Unacceptable, stop operations "and maybe" go to level3

Hazard analysis worksheet

Below is hazard analysis worksheet, where every number e.g., from 1.0 to 2.0 are in one hazard part.

Note, those hazards (which are necessary) are in below diagram. And some riks mentioned in PHL are removed and separated because they do not need and was unnecessary for this project.

20.4..1 Before operation Preliminary Hazard List

ID	Hazard	Causes	Effect	Probability	Severity	HRI
1.0	Loss of software program	Poorly designed user interfaces	Crash the system	Remote	Minor	D3
1.1	Loss of software program	Direct programming errors	Crash the system	Remote	Minor	D3
1.2	Loss of software program	Unexpected software program crash	Cannot send command/receive status	Remote	Minor	D3
2.0	Loss of communication	Connection to WIFI missing/disconnect	Cannot send command	Remote	Minor	D3
2.1	Loss of communication	Unsecured cable connections	Cannot send command	Remote	Minor	D3
2.2	Loss of communication	Unexpected communication crash	Cannot send command	Remote	Minor	D3
3.0	Loss of GPS	Radio wave interferences	The position of bike missing	Improbable	Negligible	E4

20.4..2 Under operation Preliminary Hazard List

ID	Hazard	Causes	Effect	Probability	Severity	HRI
1.0	Collision with another object	Unwanted objects get in the way	Damages the bike	Improbable	Minor	E3
1.1	Collision with another object	Late action for direction changes	Damages the bike	Remote	Minor	D3
1.2	Collision with another object	An incorrect direction handles	The bike falls	Remote	Negligible	D4
1.3	Collision with another object	To fast speed	Damages the bike	Remote	Minor	D3
1.4	Collision with another object	To low speed	Damages the bike	Remote	Minor	D3
2.0	Collision with other vehicles	Late actions for direction changes	Major damages of the bike	Remote	Minor	D3
2.1	Collision with other vehicles	Incorrect directions handle	Damages the bike	Improbable	Minor	E3
2.2	Collision with other vehicles	Unwanted vehicles get in the way	Damages the bike	Remote	Minor	D3
2.3	Collision with other vehicles	Wrong angular handle	Damages the bike	Remote	Minor	D3
2.4	Collision with other vehicles	To fast speed	Major damages of the bike	Improbable	Critical	E2
2.5	Collision with other vehicles	To low speed	Damages the bike	Remote	Minor	D3
3.0	Loss of GPS	Radio wave interferences	The position of bike missing	Probable	Negligible	B4
4.0	Loss of communication	The WIFI disconnects	Missing the true data	Probable	Negligible	B4
5.0	Loss of power	The battery discharges	Missing the power and the bike cannot move	Remote	Critical	D2
5.1	Battery start fire	The battery is too old	Missing the power and the bike cannot move	Remote	Critical	D2
5.2	Battery start explosion	The battery has damages	Missing the power and the bike cannot move	Remote	Critical	D2
5.3	Battery start explosion	The battery is overcharging	Missing the power and the bike cannot move	Remote	Critical	D2
6.0	Loss of speed sensors	To old sensors	Inability to save value	Improbable	Minor	E3
6.1	Loss of speed sensors	Device wear out	Wrong value	Improbable	Minor	E3

ID	Hazard	Causes	Effect	Probability	Severity	HRI
7.0	Loss of motor function	Unsafe power to the motor	Motor stop	Remote	Minor	D3
7.1	Loss of motor function	Collision with another vehicle	Damages the bike	Remote	Minor	D3
7.2	Loss of motor function	Collision with another object	Motor stop	Remote	Minor	D3
7.3	Loss of motor function	Broken cables	Motor stop	Improbable	Minor	E3
7.4	Loss of motor function	Used of damaged motor	Motor stop	Improbable	Minor	E3
8.0	Inaccurate input data information to MyRio	Wrong value incoming from MyRio	Other part can misunderstand the input	Probable	Negligible	B4
8.1	Inaccurate input data information to MyRio	Wrong value incoming from MyRio	Wrong registration	Probable	Negligible	B4
8.2	Inaccurate input data information to MyRio	Wrong value incoming MyRio	Missing the true status of the bike	Probable	Negligible	B4
8.3	Loss of the bike in the start	SW crash	System shutdown	Remote	Minor	D3
8.4	Loss of the bike in the start	HW (motor) crash	System shutdown	Remote	Minor	D3

20.4..3 After operation Preliminary Hazard List

ID	Hazard	Causes	Effect	Probability	Severity	HRI
1.0	Data not registred	Wrong record data from Myrio	No data to status of the bike	Occasional	Negligible	C4
1.5	Data not registred	Hardware destruction	Insufficient value	Remote	Negligible	D3

20.4..4 Result for every levels

Level 4: Unacceptable, stop operation and rectify immediately.
This project has no Hazard on Level 4, and therefore we remove that from project.

Level 3: Undesirable, upper-management decision needs to accept or reject .

20.4..5 Before operation has not any hazard in level 3

Under operation the Preliminary Hazard List (PHL) have three HRI with high level. Here the Upper manager need to take decision to accept or reject the risk.

ID	Hazard	Causes	Effect	Probability	Severity	HRI
5.0	Loss of power	The battery discharges	Missing the power and the bike cannot move	Remote	Critical	D2
5.1	Battery start fire	The battery is too old	Missing the power and the bike cannot move	Remote	Critical	D2
5.2	Battery start explosion	The battery has damages	Missing the power and the bike cannot move	Remote	Critical	D2

After operation has not any hazard in level 3

Level 2: Acceptable II, acceptable with management review

20.4..6 Before operation Preliminary Hazard List

ID	Hazard	Causes	Effect	Probability	Severity	HRI
1.0	Loss of software program	Poorly designed user interfaces	Crash the system	Remote	Minor	D3
1.1	Loss of software program	Direct programming errors	Crash the system	Remote	Minor	D3
1.2	Loss of software program	Unexpected software program crash	Cannot send command/receive status	Remote	Minor	D3
2.0	Loss of communication	Connection to WIFI missing/disconnect	Cannot send command	Remote	Minor	D3
2.1	Loss of communication	Unsecured cable connections	Cannot send command	Remote	Minor	D3
2.2	Loss of communication	Unexpected communication crash	Cannot send command	Remote	Minor	D3

20.4..7 Under operation Preliminary Hazard List

ID	Hazard	Causes	Effect	Probability	Severity	HRI
1.0	Collision with another object	Unwanted objects get in the way	Damages the bike	Improbable	Minor	E3
1.1	Collision with another object	Late action for direction changes	Damages the bike	Remote	Minor	D3
1.3	Collision with another object	To fast speed	Damages the bike	Remote	Minor	D3
1.4	Collision with another object	To low speed	Damages the bike	Remote	Minor	D3
2.0	Collision with other vehicles	Late actions for direction changes	Major damages of the bike	Remote	Minor	D3
2.1	Collision with other vehicles	Incorrect directions handle	Damages the bike	Improbable	Minor	E3
2.2	Collision with other vehicles	Unwanted vehicles get in the way	Damages the bike	Remote	Minor	D3
2.3	Collision with other vehicles	Wrong angular handle	Damages the bike	Remote	Minor	D3
2.4	Collision with other vehicles	To fast speed	Major damages of the bike	Improbable	Critical	E2
2.5	Collision with other vehicles	To low speed	Damages the bike	Remote	Minor	D3
3.0	Loss of GPS	Radio wave interferences	The position of bike missing	Probable	Negligible	B4
4.0	Loss of communication	The WIFI disconnects	Missing the true data	Probable	Negligible	B4
6.0	Loss of speed sensors	To old sensors	Inability to save value	Improbable	Minor	E3
	Loss of speed sensors	Device wear out	Wrong value	Improbable	Minor	E3
7.0	Loss of motor function	Unsafe power to the motor	Motor stop	Remote	Minor	D3
7.1	Loss of motor function	Collision with another vehicle	Damages the bike	Remote	Minor	D3
7.2	Loss of motor function	Collision with another object	Motor stop	Remote	Minor	D3
7.3	Loss of motor function	Broken cables	Motor stop	Improbable	Minor	E3
	Loss of motor function	Used of damaged motor	Motor stop	Improbable	Minor	E3
8.0	Inaccurate input data information to MyRio	Wrong value incoming from Labview?	Other part can misunderstand the input	Probable	Negligible	B4
8.1	Inaccurate input data information to MyRio	Wrong value incoming	Wrong registration	Probable	Negligible	B4
8.2	Inaccurate input data information to MyRio	Wrong value incoming	Missing the true status of the bike	Probable	Negligible	B4
8.3	Loss of the bike in the start	SW crash	System shutdown	Remote	Minor	D3
8.4	Loss of the bike in the start	HW (motor) crash	System shutdown	Remote	Minor	D3

20.4..8 After Operation Preliminary Hazard List

After operation has no hazard in level 2.

Level 1 Acceptable I, Acceptable without review

20.4..9 Before operation Preliminary Hazard List

ID	Hazard	Causes	Effect	Probability	Severity	HRI
3.0	Loss of GPS	Radio wave interferences	The position of bike missing	Improbable	Negligible	E4

20.4..10 Under operation Preliminary Hazard List

ID	Hazard	Causes	Effect	Probability	Severity	HRI
1.2	Collision with another object	An incorrect direction handles	The bike falls	Remote	Negligible	D4

20.4..11 After operation preliminary Hazard List

ID	Hazard	Causes	Effect	Probability	Severity	HRI
1.0	Data not registred	Wrong record data from MyRio	No data to status of the bike	Occasional	Negligible	C4

Causes have we take care of is level 3,, there HRI has an undesirable level. After writing recommendations and the effect of recommendations – the upper manager will take the decision to accept or reject the request. Those Hazard happened during the operation part, so that is important to take care of.

The recommendation comes from the head of security, which aims to reduce the probability and consequence of these hazard. When the probability and severity get a lower categorization, then there is a change in the HRI. New and lower HRI will change the level of danger. Then it is easier to see if we need to implement the new recommendations or not.

To make decisions for just such dangers, a senior manager needs to make decisions. But because our project is not like ordinary big projects, and we are limited with people. Then it is necessary for our team leader, with the help of the safety manager, to decide for each danger in level 3.

After that, the security chief will decide to proceed with it.

Hazard Control

ID	Hazard	Causes	Effect	Probability	Severity	HRI	Recommendation	Effect of recommendation
5.0	Loss of power	The battery discharges	Missing the power and the bike cannot move	Remote	Critical	D2	Voltage regulations needs	It will change the HRI from D2 to D3 (Remote, Minor)
5.1	Battery start fire	The battery is too old	Missing the power and the bike cannot move	Remote	Critical	D2	Change the battery to new one	It will change the HRI from D2 to D3 (Remote, Minor)
5.2	Battery start explosion	The battery has damages	Missing the power and the bike cannot move	Remote	Critical	D2	Change the battery to new one	It will change the HRI from D2 to D3 (Remote, Minor)

Manager Decision

- ID 5.0: Buy a new one.
- ID 5.1: change the battery, buy a new one.
- ID 5.2: change the battery, buy a new one.

Safety Manager Because the Project manager has decided to use a new battery (bought last week). Then the safety manager interprets to move on. Therefore, the safety manager looks at the safety part for battery. What we should do under operation when the battery start burning.

Recommendation:

- Go from the burning battery to a safe place
- Let the battery burn
- Call 112

U Safety Requirement

Author: Gabriel Sherif

21.1. Safety Requirement

The primary purpose of this document is safety requirements. The safety requirement for the Autobike project is those requirements for safety purposes. They are related to any other requirement for the Autobike project requirement. During the different phases of the project with varying methods inside hazard analysis, the requirement has been specified and produced by the safety manager. The condition took with the project leader and discussed. After the discussion, the approved requirements remainders and continued to work with them to find solutions or change or delete them.

The safety manager took care of level 3; theoretical, the HRI has an undesirable level. Those hazards happened during the operation, so that is important to take care of. The safety manager recommends some changes for those Hazards at this level. The safety requirement will change the probability and severity classification, which changes the Hazard Risk index (HRI). There, the HRI from undesirable level will go down at least to an acceptable level 2 - acceptable with management review. But, because our project is not like ordinary big projects, and we are limited with people. Then, the safety requirement will change the hazard levels.

Here, we take care of those hazards that happened during the operation, so that is important with recommendations and look at the effect of recommendation. Those hazards have high levels of Hazard Risk Index (HRI). Whoever, the upper manager decided to accept the recommendation for those hazards. There, every recommendation will change the HRI-level to a lower level.

ID	**Hazard**	**Causes**	**Effect**	**Probability**	**Severity**	**HRI**	**Recommendation**	**Effect of recommendation**
5.0	Loss of power	The battery discharges	Missing the power and the bike cannot move	Remote	Critical	D2	Voltage regulations needs	It will change the HRI from D2 to D3 (Remote, Minor)
5.1	Battery start fire	The battery is too old	Missing the power and the bike cannot move	Remote	Critical	D2	Change the battery to new one	It will change the HRI from D2 to D3 (Remote, Minor)
5.2	Battery start explosion	The battery has damages	Missing the power and the bike cannot move	Remote	Critical	D2	Change the battery to new one	It will change the HRI from D2 to D3 (Remote, Minor)

Note: D2 in Hazard Risk Index are in the high level – level 3 - an undesirable level.

21.1..1 Safety Requirement for future works with goal to hazard reductionsub

- Loss of power, the battery discharges
  - The battery power levels shall be in safe level before operation
  - The battery shall have a voltage regulation (Old battery)
- Battery start fire, the battery is too old

  - The battery shall have untouchable terminal for both negative and positive
  - The battery shall have cover to separate the batteries from rest of the Bike

- Battery start explosion, the battery has damages
  - The battery shall not overcharge
  - The battery shall not short circuit
  - The battery shall not near an external source of heat
  - The battery shall not near a fire exposure

21.2. General proposal for safety requirement

Note: The general proposal for the safety requirement includes not all parts for the safety. Below are some requirements for those hazards with a high level - the undesirable level.

- The battery power levels shall be in safe level before operation
  - Be sure the battery power levels are in safe level before operation
  - Measure both poles with a voltmeter before using
- The battery shall have a voltage regulation (Old battery)
  - Adopt a voltage regulation, to watch the over- and under voltage
- Tha battery shall have untouchable terminal for both negative and positive
  - Do not touch both terminal of the battery, the negative and the positive
- The battery shall have cover to separate the batteries from rest of the Bike
  - Cover the battery to separate the batteries from the rest of the Bike
- The battery shall not overcharge
  - Do not overcharge the battery by using an intelligent charger
- The battery shall not short circuit
  - Be sure to place the battery in fully enclosed inner packing
  - Replace the battery if some component makes heat
- The battery shall not near an external source of heat
  - Be sure to place the battery in fully enclosed inner packing
  - Replace the battery if some component makes heat
- The battery shall not near a fire exposure
  - Be sure to place the battery in fully enclosed inner packing
  - Be sure to not have loose wire connections (ARC flash)

An important recommendation is: if the battery start burning under operation

- Go from the burning battery to a safe place
- Let the battery burn
- In case of injury or harm Call 112

V Quality management plan for project AutoBike 2021

Author: Viktor Aronsson Karlsson
M.Sc.Eng Dependable Systems (in progress)
Mälardalens University, Västerås, Sweden
School of Innovation, Design and Engineering
Email: vkn17002@student.mdh.se

Introduction

Author: Viktor Aronsson Karlsson

This document will contain how the project is structured and how the quality of the project will be guaranteed. The document contains the rolls and responsibilitys in section 22.1., the desired deliverables of the project in section 22.3., how the documentation shall be structured in section 22.4., how changes in the documents will be handled in sections 22.5. and 22.7. and how the review process shall be performed in section 22.6..

For an overview and understanding of the project, see related documents: "Project description" and "Scope of the project"

Related documents

- Project description, Appendix M
- Scope of the project, Appendix N
- Requirements Specification Document for project AutoBike 2021, Appendix P

22.1. Rolls and responsibility

Author: Viktor Aronsson Karlsson

Roll	Group member	Initials	Responsibility
Project manager	Viktor Aronsson Karlsson	VAK	Planing, distributing tasks and organizing the team/project
Chief engineer	Viktor Aronsson Karlsson	VAK	Have an overview of all technical aspects of the project
Requirement manager	Love Briden	LB	Creation and responsibility of all Requirements documents
Validation and Verification manager	Maya Zawawi	MZ	Creation and responsibility of all Validation and Verification documents and tasks
System Safety manager	Gabriel Sherif	GS	Creation and responsibility of all System safety documents
Quality manager	Viktor Aronsson Karlsson	VAK	Assuring that everything is documented and assuring the quality of the documents
Hardware lead	Love Briden	LB	Analysing of hardware and responsible for hardware related tasks
Minor Hardware	Gabriel Sherif	GS	Analysing of hardware and hardware related tasks
Software lead	Maya Zawawi	MZ	Creation of software and responsible for software related tasks

Minor Software	Viktor Aronsson Karlsson	VAK	Creation of software and software related tasks

Table 5: Describe which group member is responsible for which roll

22.2. Stakeholder

Author: Viktor Aronsson Karlsson
The stakeholders for this project are:

- Mälardalen University (MDH)
- Chalmers University
- Volvo Cars
- Veoneer
- Autoliv
- Cycleurope
- AstaZero

Where the main stakeholders that are concerned in this document are the ones from Chalmers and MDH, this also includes the students within the project group.

22.2..1 Updating Stakeholder

Author: Viktor Aronsson Karlsson
The way that is used to keep the stakeholders up to date on the projects progress, is by having weekly or biweekly meetings with the main stakeholder from Chalmers. In the meetings it is discussed on what have happened in the project since the last meeting, and the near future plans until the next meeting.

22.2..2 Stakeholder Goals

Author: Viktor Aronsson Karlsson
See related document "Scope of the project" and "Requirements Specification Document for project AutoBike 2021" for an extensive look at the stakeholders goals.

22.3. Project deliverables

Author: Viktor Aronsson Karlsson
The project involved many stakeholders where each have different wants from the project. This section will list of the main tasks and documents that will be produced during the course of this project.

- Documents
  - Requirements specification
  - Project Plan
  - Safety Management Plan
  - Quality and Process Assurance Plan (This document)
  - Validation and Verification Management Plan

  - Validation and Verification Results
  - The Final Project Report

- Tasks
  - LabVIEW Program for controlling the Autobike
  - Component diagram of the hardware, for easier assembly
  - Validation of the correct functionalities by the Autobike's sensors and actuators

22.4. Documentation specification

Author: Viktor Aronsson Karlsson
To create a form of uniformity and structure between the documentations in the project. There are some elements within all documents that need to exist, the elements can be seen in the list below. At the start of each document, both names and emails of each participant in the project group have to be stated. The name of their school and education need to be stated at the start as well.

Sections:

- Introduction to the document
- Related documents
- Revision table (If a revision have taken place)
- Stating the author at each section of the document
- References, at the end of the document

22.4..1 Microsoft Teams

Author: Viktor Aronsson Karlsson
The project groups Teams channel need to have structure when it comes to the organization of the documents. To ensure an comfortable experience while browsing the teams channel, folders will be used to organize the documentation. Each roll will have their own folder, where all documents that are related to that roll will be stored. All finalised documents will be stored in a separate folder for separation purposes, so the finalised document isn't tampered with without approval.

22.5. Traceability of documents

Author: Viktor Aronsson Karlsson
Traceability is an important aspect of a project, it is used in order to know who contributed to the different tasks. The method used to track who contributed to the document is by writing, at the start of every section in an document, the name of the author who wrote the text. When the first version of a document is finalised it will be noted down as version 1 in a separate document named "Revision History - All documents". This document exists for the purpose of tracking the major changes made to the various document through out the project. The things that shall be noted down are: the name of the one who made the change, the date of the change, what was changed, the other affected documents, which version of the document it is and lastly the name of the reviewer and approver.

22.6. Quality review

Author: Viktor Aronsson Karlsson
The reviewer of an documentation need to be someone not related to the document, someone that did not write the document or participated in task. The reviewer need to analyse the document and all other documents that reference the document in question, to see if the affect of the change

have been noticed and neutralised by the one who made the change. Comments of reviewed work shall not be erased after they have been resolved instead they shall be noted down in the revision document under the section "Reviewer Comment".

22.7. Dealing with changes in the documents

Author: Viktor Aronsson Karlsson

When a change is made to a document, the change have to be noted down in the revision document and the quality manager need to be notified that a change have occurred. The quality manager will then decide who should validate the change made. Lastly the project manager need to approve the change made, before the new version of the document can be considered finalised. In the case of alterations to paragraphs in documents, the authors name shall not be altered instead if major alteration to the text is made the name of the one who did the alteration will be added to the section.