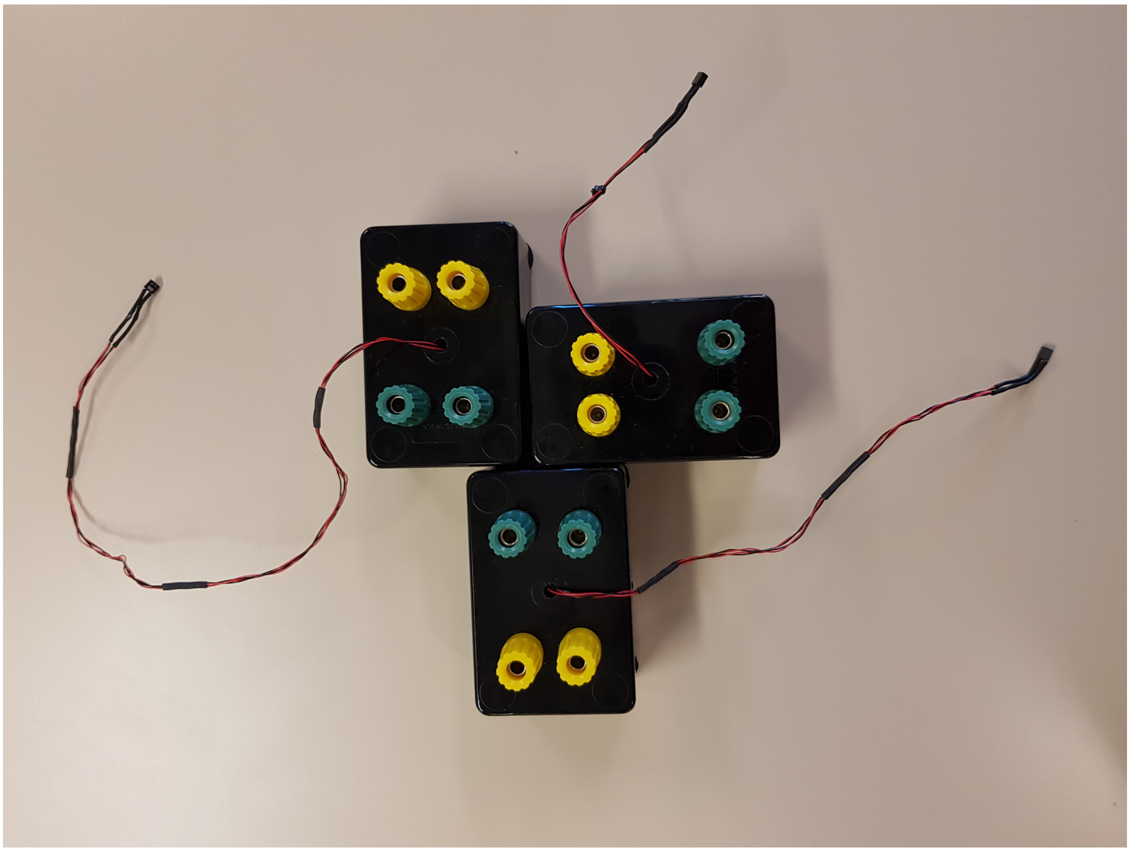
CHALMERS
UNIVERSITY OF TECHNOLOGY



Wireless Data Transfer System Bachelor Thesis

From Permanent Magnet Synchronous Machine To Measurement System

Bachelor's thesis in Electric engineering

Erik Benjaminsson and Filip Kaiser

Institution of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Bachelor's thesis

Wireless data transfer system bachelor thesis from permanent magnet synchronous machine to measurement system

Erik Benjaminsson
Filip Kaiser

Institution of Electrical Engineering

Chalmers University of Technology
Gothenburg, Sweden 2019

Wireless Data Transfer System Bachelor Thesis From Permanent Magnet Synchronous Machine To Measurement System

Bachelor's Thesis 2019

Students:
Erik Benjaminsson
Filip Kaiser

Supervisor:
Anders Mellbin, Electric Drive Analysis & Verification Test Engineer at Volvo Cars

Examiner:
Jian Yang, Institution Communication and Antenna Systems at Chalmers University of Technology

Institution of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Wireless Transfer System for Permanent Magnet Synchronous Machine.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2019

Wireless data transfer system bachelor thesis from permanent magnet synchronous machine to measurement system

ERIK BENJAMINSSON
FILIP KAISER
Electrical engineering, Chalmers University of Technology

Abstract

Volvo Cars is developing electric machines for cars, these machines should be durable, efficient and produce high power. When new machines are developed the verification process is crucial to determine the best design to meet the right specifications. In a Permanent Magnet Synchronous Machine (PMSM) the magnets are located inside the rotor which causes problems when trying to analyse permanent magnets temperature one element which is of high importance. Connecting the measurement system to the rotor is hard and demands high accuracy to read the right temperatures. During this project a wirelesses transmitter circuit used for reading temperature status were constructed. Using a microcontroller (PIC16F1827) as a temperature decoder. IR-diode to send and a phototransistor to receive the data wirelessly and reading and translating the data using a computer. To power the circuit a wireless power supply where planned to be built but could not be designed due to lack of time and experience.

Keywords: electric motor, wireless, circuit, A/D, temperature, magnet, Arduino, microcontroller, PIC16f1827, RS-232

Acknowledgements

For helping us complete this bachelor's thesis, we would like to thank our supervisor Anders Mellbin, at Volvo Cars, for being a great support and for assigning this thesis to us and helping us getting a hold of necessary components and equipment needed for our experiments, and last but not least: giving us the opportunity to perform a real rewarding bachelor thesis at Volvo cars.

We would also like to thank everyone who helped us at Volvo Cars for trusting us and giving us the information and support we needed. We would also like to thank the people at Electric maintenance for offering us a workstation to sit and test our experiment, tours of the factory and offices as well as sponsoring us with material needed for the experiment.

Thanks to Göran Hult, Arto Heikkilä, Erik Agrell, Thomas Hammarström and Sakib Sistek at Chalmers University of Technology, for helping us whit questions and excellent sounding board. Supporting us with tips, giving us leads to find results and providing us with a work space at Lindholmen.

Special thank to our examiner Jian Yang at Chalmers University of Technology, for the support and the final consultation of this report.

Erik Benjaminsson and Filip Kaiser, Gothenburg, 2019

Contents

List of Figures

1 Introduction

This chapter the topic for this thesis about wireless transmission in a electrical machine will presents. The background, purpose and scope, as well as the set limitations of this project.

1.1 Background

Volvo Cars is developing a new electric machine in the pursuit of fabricate and designing the next generation of electric cars. The machines has to be durable, efficient and produce a high torque. When new electric machines are developed the verification process is crucial to determine the best possible machine design which reaches the set specifications and requirements. In this report the scope is the verification of the permanent magnets temperature inside the Permanent Magnet Synchronous Machine (PMSM). This is because if the magnets get to hot they will lose its magnetizing characteristic which will break the machine since the magnets is a vital part of its construction. Volvo cars is now using a third party machine to transfer the magnets temperature and now they wants to build a in-house product that can replace the third party wireless data transfer system for their test rigs. The goal is to make the measurement more custom fitted to their own test rigs and a in-house product will be more cost and service efficient.

1.2 Purpose

The purpose of the project is to design a wireless transfer system for Volvo cars electrical machine test rigs, because Volvo Cars is now using a third-party machine to do this which is expensive to buy and is to advanced for what it is supposed to do. The project is needed when testing temperature requirements in new electric machines.

1.2.1 Scope

The scope is divided into packages, this is done to get one working module at the time, mostly for Volvo to easily finish the project if there is not enough time to finish it during the bachelor thesis. So depending on the time every chapter takes the scope will be extended as following.

- Analog to Digital converter
- Transmitter
- Receiver
- Stable transmission
- Induction power
- Assembly and testing

1.3 Limitations

The limitations of the project are only focus on measuring and transferring data for the temperature. Another limitation of the circuit that it is just needed to transfer in a very slow frequency (at least 1Hz), and to just finish as many packages/modules that is possible in the time-frame.

1.3.1 Requirements

- The data needs to be transferred wirelessly, at least 1Hz transfer frequency. This is because a wired connection can not handle a rotation speed. The cable would most certainly tare and break after extensive use.
- The components used should be able to handle and operate fully when spinning in 20 000 rpm.
- The components and circuit should also withstand any vibrations and temperatures up to 130 °C.
- The machine could have some frequencies where the transmitter and receiver could be disrupted. This should not be a problem for the designed transfer system.
- The transmitter circuit should be small enough to fit on the axis, around 40mm in diameter.
- The circuit should be able to read from 8 channels in parallel and output the 8 analog inputs in series converted to a bit pattern.
- The transmission should be able to travel at least 4-10mm.
- The system should have a minimal of 1 HZ transfer speed for each channel, so for the 8 channels it has to faster then 8 Hz per cycle.

2

Theory

The theory for this project is knowledge of induction, wireless transferring, electronic construction and microcontroller programming. Following this introduction there is some theory that is needed to understand the project and a word list to see some of the abbreviation used in the text. The necessary software's and technique, what will be needed for it and how it was used in testing and experiments.

2.1 Word list

- **A/D-converter**: Analog to digital conversion.
- **I/O**: Input and output channels.
- **PMSM**: Permanent magnet synchronous machine.
- **RTD**: Resistance temperature detector.
- **Rotor**: The part of the PMSM that rotates.
- **Circuit**: The electrical layout of the components.
- **Arduino**: A microcomputer that is programmable.
- **MSB**: Most significant bit.
- **LSB**: Least significant bit.
- **PWM**: Pulse width modulation.
- **Potentiometer**: A resistor that can change resistance analogously.
- **Protocol**: A way to send data in a specified standard.
- **4-wire method**: A method used for more precise resistance measurement.
- **EMS**: Electro magnetic Susceptibility, which stands for how good the circuits are at withstanding the EMI (electromagnetic interference).
- **VDD**: The label to an input with positive voltage (3.3 to 5.5 V).
- **VSS**: The label to an input with negative voltage (ground).
- **VPP**: Programming voltage.
- **Fosc**: Frequency of Oscillator.
- **Tx**: Transmission.
- **Rx**: Receiver/ Receive/ Reception.
- **TTL**: Transistor-Transistor logic, explain what level the voltage need to be for a logic 1 or 0.
- **bsp**: Bits per second.

2.2 The Electric Machine

The electric machine that this system is supposed to be for is of type permanent magnet synchronous machine (PMSM). This means that the magnets is embedded inside the steel frame, also called the rotor, that has a permanent magnetic field [1]. The machine also has a stator which has a fluctuating magnetic field. The magnetic field is rotating around the stator which makes the rotor rotate [3]. Because of the constant magnetic field inside the rotor this machine can not use induction windings to start, instead it has to use a voltage which varies in frequency [2]. Inside this machine there are temperature sensors, RTDs (PT-100), that are measuring the temperature of the magnets inside the machine. This is because if the magnets gets to hot they will lose their magnetic capabilities which will render the machine useless since the magnets are a vital part for it to work. The PT-100 cables are then drawn out to the axis of the machine where they are connected to the transfer system, the connection to the transfer system are by the 4-wire method which will make it more precise.

2.3 PT-100

The temperature receivers inside of the electrical machine is a type of RTDs, as mentioned in section "The Electric Machine", which stands for resistance thermometer detector, could also be called thermoelement. The RTDs is changing its resistance depending on the temperature of its surroundings [4]. If the temperature rises the resistance will rise and if the temperature decreases the resistance will also decrease. The PT-100 is built of platina and has a temperature alpha (α) of $3.85 * 10^{-3}$ and a base resistance of 100 Ω. The formula for calculating the temperature is: $temp = ((R_{meas}/100) - 1)/\alpha$. In this project there will be 8 different PT-100 that the measurements will come from and each, because of all EMS that is in the machine, will be wired by the 4-wire method [5].

2.4 Transmitter and Receiver

The transmitter will be designed with the microcontroller which sends data wireless trough an IR-diode. There are other ways to design a transmitter, for example using Bluetooth or WI-FI. IR was chosen because it is the least sensitive to disturbances from the magnetic fields and frequencies from the machine. The transmitter will operate after a protocol in 9600 bps and follow RS-232 standard which is a serial protocol for transmitting data.

The receiver will use a photo-transistor to pick up the IR waves and a computer, alternatively Arduino that can processes the data from the receiver. It is needed for the receiver to be close to the transmitter for the best transfer signal. Arduino was mainly chosen because it has a complete CANN-protocol which can be connected directly to Volvo Cars industry. The photo-transistor receiver works like a "pull down resistor".

The chosen transmitter is a IR-LED diode from Kjell & Company (1,5-1,6 V, 60 mA (120 mA max @ 10 % duty cycle), this will maybe be changed to another IR-led for the final product [6]. The chosen receiver is a photo-transistor receiver with dark lens (1,2 - 1,3 V, 940 nm) from Kjell & Company, bought in the same kit as the IR-LED diode [6]. Se figure 2.1 to make out the two different diodes.

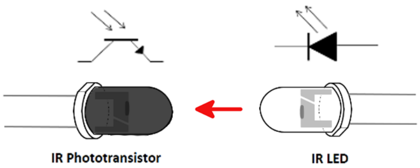


Figure 2.1: IR transmitter (IR LED) and receiver (Phototransistor

2.4.1 Microcontroller

A microcontroller is a small chip used to preform tasks programmed by the user. The chip consist of one or more CPU along with a small memory and programmable pin peripherals. These peripherals can be configured with MPLAB, described in section 2.5.1 MPLAB® X IDE . For this project the chip PIC16F1827 [7] i used. It is a microcontroller with a military grade standard, which means that the chip can operate in temperatures from -55 ℃ to 125 ℃. Military grade chips are the highest grade and with the preset requirements by Volvo Cars, this chip is the only one that comes close to operate in 130 ℃. The chip was provided and recommended by Göran Hult at Chalmers Lindholmen. The specified data-sheet and protocol is called "PIC16(L)F1826/27" [8] and this data-sheet describes how step by step set up the PIC16F1827. From the data-sheet common information about the device can also be read. PIC16F1826 (See figure 2.2) contains a mid-range 8-bit CPU with 18 pins I/O where 15 pins are programmable. Pin 4 (VPP), pin 5 (VSS) and pin 14 (VDD) are for driving and powering the chip. This chip will be used as the transmitter decoder since it is a small microcontroller, it can do the set tasks for the project analog to digital conversion, serial output and its durable. It is also small and light which makes it suitable for space and weight limitations.

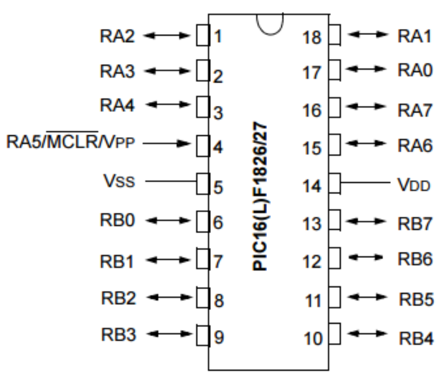


Figure 2.2: PIC16F1827

2.5 Software

To get better understanding in the project process and how it was performed. Certain programs and steps have to be explained in order to fully grasp how the research and experiments was done.

2.5.1 MPLAB® X IDE and Dataman

MPLAB® X IDE is a software used to program the microcontroller [9]. It runs on PC and uses the programming language C, C++ or assemble. It supports project management, script editing, ,file structure, debugging and programming. It can compile and program either for 8-bit, 16-bit and 32-bit PIC microcontrollers. For this project the compiler "MPLAB XC8-C compiler for 8-bit PIC and AVR devices" is used and it is compatible with PIC16F1827. An additional driver was installed to MPLAB® X IDE. A plugin called "MPLAB® Code Configurator" or MCC for short. MCC is a supplementary to the MPLAB® X IDE program that simplifies the microcontroller programming with menus and graphic tutorials. The MCC automatically generates code using a Graphical User Interface (GUI). The GUI provides easy settings to configure the peripherals on the microcontroller and directly sets the necessary bits. For MCC menu layout see appendix 1 figure A.1. To syntheses and print the code to the microcontroller hardware and software from Dataman is used. Dataman is a universal syntheses tool for microprocessors up to 48 pins [10].

2.5.2 KiCAD

KiCad is an open source software for Electronic Design Automation (EDA) [11]. The programs handle Schematic Capture, and PCB (Printed Circuit Board) Layout. KiCad runs on any PC. In this project KiCad is used to illustrate the circuits, both the Transmitter and Receive circuits (See electrical schematics in Appendix B).

2.5.3 Arduino

Arduino is an open-source computer chip, in this project it is used to read the data from the transmitter. It is fully programmable for any need which makes it suitable for this project since it can use preprogrammed CANN-protocol or other protocols. With the Arduino IDE (Integrated Development Environment) the users can program the chip with a standard computer connected trough USB. The IDE is easy to use, with instructions and a single button press to upload the code to the physical Arduino board [12].

2.6 Datasheet and protocol

2.6.1 Enhanced Universal Synchronous Asynchronous Receiver Transmitter

To activate necessary registers associated with the chips specifications for data transmissions a methods called "Enhanced universal synchronous asynchronous receiver transmitter" (EUSART) and "Baud Rate Generator" (BRG), page 285-305 in datasheet [7] which describes in detail how to initiate registers and PORTS for the transmitter part. The EUSART module is for serial data transmission via I/O communication from the microcontrollers. The data can be transmitted between two devices in serial communication, either as synchronous which has a separate clock, crystal and data line in the microcontroller or asynchronous transmission which uses an internal clock signal in the microcontroller. Therefor only one pin needs to be used for transmission and since this project uses one IR-diode transmission serial asynchronous data communication is the best choice. This is set up in the config settings as "FOSC = INTOSC". Because with asynchronous the data can run through single cable from the microcontroller to the IR diode and then transmitted bit- by- bit in a serial pattern (See Figure 2.3). It also requires less circuitry which reduces overall weight. Compared to parallel data communication and synchronous operation which would need multiple wires and multiple components to transfer the data.

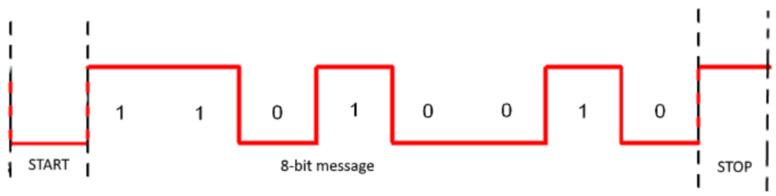


Figure 2.3: 8-bit data message with , start and stop-bit

One of the most common use of the EUSART in asynchronous mode is to communicate with the standard RS-232 protocol, more information can be found [13], [14]. The EUSART can be configured to transmit 8-bits by configuring bit TX9 in the TXSTA register. Once data has been read from the ADC input and converted to a 8-bit string message it will then move into the transmit shift register TXREG. From there they are clocked out to the TX pin on the microcontroller with a start bit and followed after the message a stop bit. Start and stop-bit sequence is used to indicate the right start and finish of the data transmission. The start-bit/start sequence is placed before the actual data. It has a set time period and set voltage arrangement with a high (logic "1") and low (logic "0") bit sequence. Same with the stop-bit but its placed after the data has been transmitted (see example on figure 2.3). In IDLE state the signal is high and goes low when the transmission start-bits is activated and when transmission is over it ends with a high stop-bit before IDLE again. One thing to remember is that the data is sent with least significant bit first, so the pattern looks mirrored in comparison to when it is written as a binary number.

As mentioned there are a handful of registers that is used to control the EUSART. The PIR1 and PIE1 registers contain interrupt flag bits. These will allow the EUSART to generate interrupts signal when the data is moved into the transmission shift register, so for example it is used before new data is written to TXREG. TXREG and RCREG is used for transmitting and receiving data on TX or RX pins and TXSTA and RCSTA registers are used to control transmission and reception. So the interesting register are TXREG and TXSTA for transmitting data. INTCON register contains bits for global interrupts and peripheral interrupt enable bits and must be used in order for the receive or transmit interrupts to occur correct. BAUDCON is used when generating the desired Baud Rate and choosing how the data should be sent on TXREG, for example if the data should be inverted or not. For more information read datasheet for PIC16F1827 [8] on pages 285-313.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON0	RXDTSEL	SDO1SEL	SS1SEL	P2BSEL(1)	CCP2SEL(1)	P1DSEL	P1CSEL	CCP1SEL	119
APFCON1	—	—	—	—	—	—	—	TXCKSEL	119
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	296
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	91
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	295
SPBRGL	BRG<7:0>								297*
SPBRGH	BRG<15:8>								297*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	127
TXREG	EUSART Transmit Data Register								287*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	294

Figure 2.4: Summary of registers associated with asynchronous transmission and baud rate. Page 293 in PIC16(L)F1826/27 data-sheet

The SPBRG register is used to set the Baud Rate. Baud Rate is a standard communication and describes how many bits per second that are be transmitted or received. Some standard Baud Rates are 300, 1200, 2400, 4800, 9600 and 19200 bps. To calculate the right value for the register this formula is used (See example in Figure 2.5).

Formulas for Baud Rate:

$$BaudRate = Fosc/(16 * (SPBRG + 1)) \quad if \quad BRGH = 1$$

$$BaudRate = Fosc/(64 * (SPBRG + 1)) \quad if \quad BRGH = 0$$

Formulas for SPBRG:

$$SPBRG = (Fosc/(16 * Baudrate)) - 1 \quad if \quad BRGH = 1$$

$$SPBRG = (Fosc/(64 * Baudrate)) - 1 \quad if \quad BRGH = 0$$

BRGH is a bit in the TXSTA register that determines in asynchronous mode and it is used to get an accurate number from the formula. This i used to get the sum as close as possible to an integer.

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$\text{Desired Baud Rate} = \frac{F_{OSC}}{64([SPBRGH:SPBRGL] + 1)}$$

Solving for SPBRGH:SPBRGL:

$$X = \frac{\frac{F_{OSC}}{\text{Desired Baud Rate}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{Error} = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

Figure 2.5: Calculating Baud Rate and the miscalculation from datasheet. Page 297 in PIC16(L)F1826/27 data-sheet

2.6.2 Analog-To-Digital Converter

Analog-To-Digital converters is used to convert an analog input signal to a digital typically 8-bit, 10-bit or 12-bit resolution which is either 255, 1023 or 4095 bits respectively, the EUSART can only transfer maximum of 8-bits with TXREG. With an 8-bit resolution and the 0-5 volt interval means that the bit sequences will change every 20mV $(5V - 0V)/255 = 0,0196 \approx 20mV$.

2.7 Wireless induction

When talking about electrical induction, by definition there mid-range is when the distance between the coils is larger than the coil dimensions, and near-range is defined as a system with a distance between coils smaller than the coil dimensions.

The basics of a wireless power transfer/supply is that some materials is needed for it to work. First you need two coils suitable for the circuit/task, a circuit that works like a oscillator (e.g a Hartley oscillator or a full bridge inverter) mostly to get a circulating magnetic field [15] which will magnetize the primary coil and transfer to the secondary and in the secondary coil the magnetic field will create a current which can be rectified to a DC power source.

2.7.1 Diode-rectifier

This project will use a single phase full-bridge diode rectifier. The rectifier will take the alternating current (A/C) and convert it to a positive current. This is working because the diode will just pass the positive current and neglect the negative, this forces the output to be a positive current. The capacitance at the end of the rectifier will limit the ripple of the output voltage and make it more stable for the microprocessor to operate with [16].

2.7.2 Oscillator/Inverter

The oscillator/inverter is needed to make an AC from a DC source [16]. This oscillator is usually designed with one or more transistors, usually MOS-transistors or PNP/NPN-transistors since they are easy to work with. The main difference between the oscillator and the inverter is the number of transistors that its consisting of and in which power ranges it is used in the inverts is used in high power circuits while the oscillator is used in low power circuits [17].

2.7.3 Coils

The coils will be wired by a wire called litz wire, which is like a woven copper thread wire. This type of wire is often use for the purpose of constructing inductors since copper has great characteristics for just this purpose. The coils will then be winded like soloids since they will face each other in the final construction. A solid is like a circle with a specified number of windings to either increase or decrease the current/voltage [16].

3

Method, Studies and Thought Process

This chapter will cover the necessary steps and how they were approached in this project. This section will also describe how the steps where executed, describe the tests that were executed and the design of the circuit.

3.1 Pilot study

To get a better understanding, certain theories described in chapter Theory had to be reached deeper in a consecutive order. This to understand and grasp the methodology of the components and experiments that were executed during the project.

Reading and understanding the tools and data-sheet was the first step in the pilot studies. This gave the fundamental understanding on how every component worked and should be used. Finding useful information and data relied heavily on other literature studies, microcontroller data-sheets and internet research such as forums and other tutorials. Reading through others thesis papers was also a great factor in the pilot study since it is unnecessary to invent the wheel once more.

3.1.1 Transmitter and receiver

The transmitter circuit will be designed with IR-diode and a photo diode transistor as described in Theory. The IR is connected to the output pin from the microcontroller and the photo diode is the receiver. To get the same output as input positive +5 volt is connect to the collector of the transistor and a voltage divider circuit is connected to the emitter. This will make the diodes pull down it is characteristic as a way to turn on and off the transistor in a bit sequence. So depending on the on and off sequence from the IR-diode this will show the output bit pattern from the emitter pin. The bit pattern and the frequency can be read from an oscilloscope at the emitter and to check if the IR-diode output is working use a cellphone camera and point it towards the diode, if it emits a purplish light the diode is active and transmitting, make sure to have the camera facing directly in the middle of the IR-diode since it has quite a narrow emitting angle.

3.1.2 Powering circuit

This part of the project is prioritized last since this whole part is extremely complex and wirelesses power transfer such a new technology. Therefore the other parts will be in main focus and this part would be something that is done if the other parts is fully completed. But two different methods are reached and tested as an possible power source.

3.1.2.1 Induction

The inductor circuit would consisting of one oscillator which will make DC to AC since that is the only way a inductor could work. Firstly build the full-bridge diode rectifier which will be on the receiving inductor coil to rectify the AC voltage to a DC voltage. After the diode rectifier is designed a simple oscillator got designed mostly to get knowledge of how the wireless transferring worked. Later it will be improved and optimized to be able to supply the transmitter with power. Inductors consist of a winding of electrically conductive wire, usually isolated copper wire. Winding revelations can be controlled depending on the inductor. Note that the inductor usually has increasing inductance with increasing temperature, it wants say it has a positive temperature coefficient. But since time is limited this was as far as the studies on this part came.

3.1.2.2 Battery powered circuit

Using batteries to power the transmitter circuit was reached. But when designing and planning the circuit all criteria and requirements set by Volvo Cars (See table 1.3.1 Requirements) where carefully followed. Rotation of 20 000 rpm, a temperatures rise up to 130 °Cand that an engine test could last for over 24 hours. Resulted in that it would be hard to use battery, because the battery would have to be quite large to power the entire circuit during long periods of time and had to be robust to withstand all external forces and temperatures. It would also add weight, that is not needed, to the circuit. These concerns lead to that a battery powered circuit was excluded from the list of possible power sources completely.

3.2 Working process

3.2.1 Test 1: A/D Conversions shown with 8 LED's

The first test circuit were built to understand and control the 8-bit ADC convention from an single analog input. A potentiometer that is serially connected to a power supply of 5 volts and resistor of 1kΩthis will act as a PT-100 as analog input. To display that the 8-bit ADC conversions and to see that it was correct converted 8 individual LED´s where connected to 8 output pins (See bottom left corner in figure 3.1). If the 8-bit sequence had a binary "1" the LED turned on and if a binary "0" kept the LED off. This design showed the 255 different bit combination.

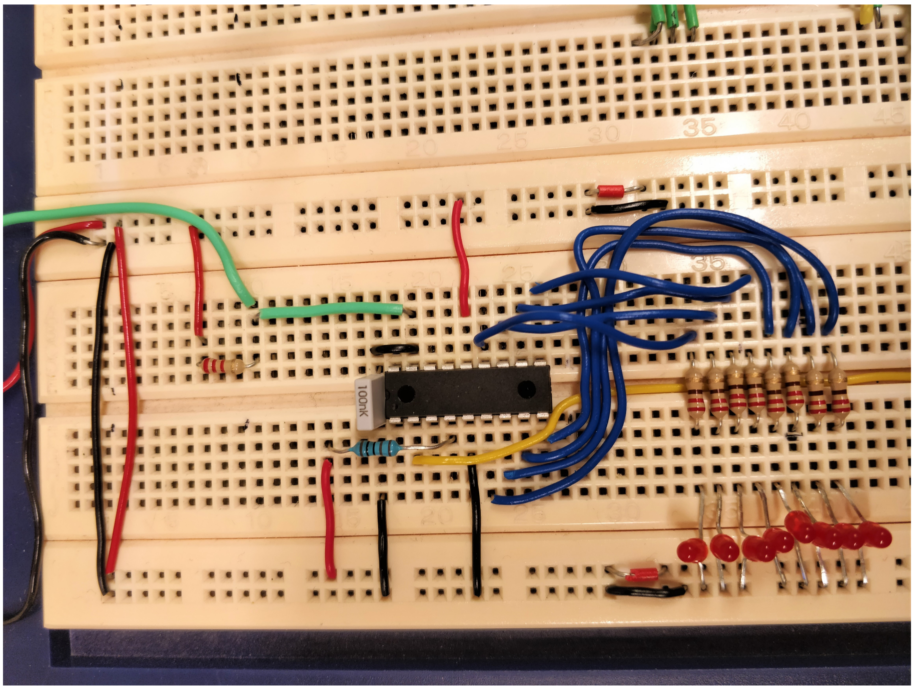


Figure 3.1: Breadboard and circuit for ADC to 8-bit LED indicator

The circuit board is colour coded to easily see the different inputs and outputs:
- "Red cables" are positive voltage from the power supply (+5V).
- "Black cables" are negative voltage from the power supply (-5V/Ground).
- "Green cables" is a 10k Ωpotentiometer analog input.
- "Yellow cables" is serial digital output.
- "Blue cables" is to the 8 LED indicators for binary output in Test 1, later on blue represents all analog input channels.

The potentiometer with analog voltage is connected to input pin RA0 on the microcontroller. To counter some disturbances in the AD conversion, an 100 nF capacitor is connected between the positive 5V input and ground, a recommendation from Göran Hult. The 8 signal output to the LED uses PIN - RB0, RB1, RB2, RB3, RB4, RB5, RB6 and RB7. The ADC value is then shown in binary "00000000" to "11111111" (RB7 = MSB and RB0 = LSB). 8 resistors (220 Ω) are placed between the output and the LED to limit the current flowing through the LEDs. To initialize and convert the analog signal input the functions "init_ADC" sets certain bits in register ADCON1. "ADC_Read" function reads from a selected input pin as variable "channel". The analog input (0-5V) is then divided up to 8 bits and returned as a 8-bit sequence. The function "init" initializes all the different ports on the microcontroller such as input pins, output pins and the internal clock (Fosc). See code in appendix 3.

3.2.2 Test 2: Analog input to serial output

After understanding how to read the analog input it is time to upgrade both the code and the circuit to transfer the data. To do this the 8 parallel output from Test 1 has to be transformed to one serial output, because the IR diode only uses one pin. For this EUSART (described in Section 2.8.1.) is used. EUSART makes it possible to send 8-bit signal directly to pre-planned TX pin on this microcontroller it is on pin RB2 on PIC16F1827 (see data-sheet PIC16(L)F1826/27, page 6).

The new plan is to read the analog input voltage on RA0, then converted with ADC to 8-bit digital code from test 1, then load the digital signal to EUSART and transmits the digital value through TXREG register which outputs on pin RB2. The desired baud rate is 9600 bits/s for the transmission and after studying the data sheet. 9600 is also a standard fro RS-232. Down below is described how to configure the registers and settings on PIC16F1827 (Everything is referred either to code in appendix 3 or the data-sheet).

A hand-full of bits has to be set in the function "init_EUSART" and configured in the config-settings:

- Special parameters and defines in start config: #pragma config Fosc = INTOSC for an internal oscillator clock and #define _XTAL_FREQ 4000000 = To set the clock oscillator frequency (Fosc) to 4 MHz.
- SPBRG = 25 to get the desired 9600 bits/s Baud Rate
- In register TXSTA: TXEN = 1 in register to enable transmission, BRGH = 1 to get a high speed division, SYNC = 0 so that the microcontroller runs in asynchronous mode.
- In register RCSTA: SPEN = 1 to enabled serial port.
- In register PIR1: TXIF = 1 is used as a transmitting interrupt flag.
- TXREG is as a buffer register in which the soon to be transmitted data is stored

In theory this settings and register initialization results in the set baud rate of 9600 bits/s. The function "ADC_Channel" can transfer any given integer (0-255) to the function "Transmit" which sends the integer to TXREG register. Conclusion, the circuit is now able to read one ADC input, convert it to 8-bit digital and transfer it at 9600 bits/s to any TXREG pin. Start and stop-bit sequence (int start = 0b11111110; //254 and int stop = 0b11111000; //248) are added to indicate when the transmission starts and when it ends. Because the output sequence should look like it was planned in figure 2.3.

3.2.3 Test 3: 8 parallel inputs to one serial output

Implementing more input channels is easily done because the code is written in such a way that it could be scaled later on for more inputs. The only thing that needs to be added are more case statements in the switch case function (See appendix 3). The switch case jump between the different cases and sets the variables "int Channel;" and "int AN_pin;" respectively to read from the different channels. Variable

"Channel_Select" increment after each case to loop trough the 8 different channel and resets after. Every case sets the specific channel and input pin that is then sent to the functions "ADC_Read" and "ADC_Channel". To keep the channel counter from getting to high a if statement resets the channel again.

3.2.4 Test 4: Reading converted output

To read the output from EUSART at 9600 bits/s the microcontroller has to be hooked up to a computer, the transferred message can be seen with a oscilloscope but the rate is to fast to read. To see the conversion from TXREG with RS-232 standard a component called MAX232 [18] is used. MAX232 is a chip that converts the output signal from TXREG to a readable TTL voltage [19] that a computer can read as input. The most common range for TTL-levels is 0-2,3V for a logic '0' and 3,3-5 for a logic '1'. The microcontroller and MAX232 are designed to work together so the connection is easy achieved by adding the MAX232 in series with the output pin RB2 (TXREG) on the PIC16F1827 to input pin 11 on the MAX232 chip (See appendix 2). To run the MAX232 chip four external electrolytic capacitors (each 1 uF) has to be added, they are required for the internal charge on the chip. A benefit is that since the MAX232 is supposed to sit on the receiver side extra components and added weight do not need to be accounted for. It is only crucial on the transmitter side. To send data a DB9 [20] connector is connected to the MAX232, see circuit in appendix 3 how to set up the entire circuit. With a PC terminal the TXREG output can be read directly on the screen and every value prints out in a long string (See figure 3.2). The blue highlighted text is one message with 8 channels. Every bit sequence is represented as a 8-bit unsigned integer in the terminal.

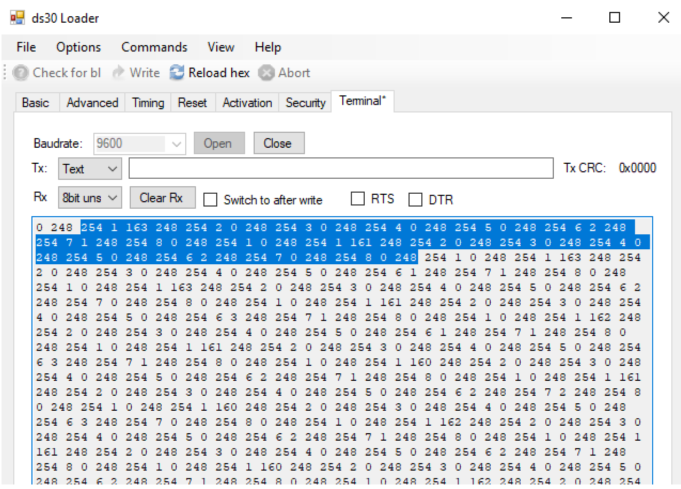


Figure 3.2: Screenshot of ds30 bootloader for transmitter test

One channel sequence start with number 254 as start bit and it is set in the code (int start = 0b11111110;), followed by the channel number from (int AN_pin;), after the channel comes the converted 8-bit ADC value (int ADC_Value_Result) and lastly the end bit 248 (int stop = 0b11111000;). This repeats in the same arrangement but with a new channel and ADC value for every channel until channel 8 then it restarts. One entire message is 64 bits long (4 sequences * (8-bit/sequence) = 64 bits/message), in theory this could send 150 messages per second ((9600 bits/s) / (64 bits/msg) = 150 messages/s). This is faster then what is needed since the temperature in the PMSM do not change that fast. To slow the transfer rate down some delays where added after each sequence in the code to slow the transmission down (const int delay2 = 100;), 100 ms per transmission gives at total of 400 ms delay per sequence. So for all 8 channels = 3200 ms delay and results in an slower transfer rate to 3-4 messages/s.

3.2.5 Test 5: Transmitting data via IR

When the serial output is established with wire the next task is to get the correct bit sequence without errors wirelessly over IR. With an oscilloscope a probe is connected on each side of the IR transmission. One probe on the microcontroller side to see the direct output from TXREG, connected to the IR diode. The other probe is connected after the photo-transistor on the other side of the transmission line. With this setup we could see that the same output where on both sides, See figure 3.3. But the voltages levels are not the same, see voltage mean TXREG = 939mV and photo-transistor = 3.57V. This was not a problem since the MAX232 chip automatically adjusts to the higher voltage. The only thing left is to switch the output wire on RB2 from the input on MAX232 (pin 11) to the IR-diode and then take the photo-diode output to input at pin 14 on the MAX232 chip intend. This results in a complete wirelesses data transmission and the output numbers should look the same as in 3.2.

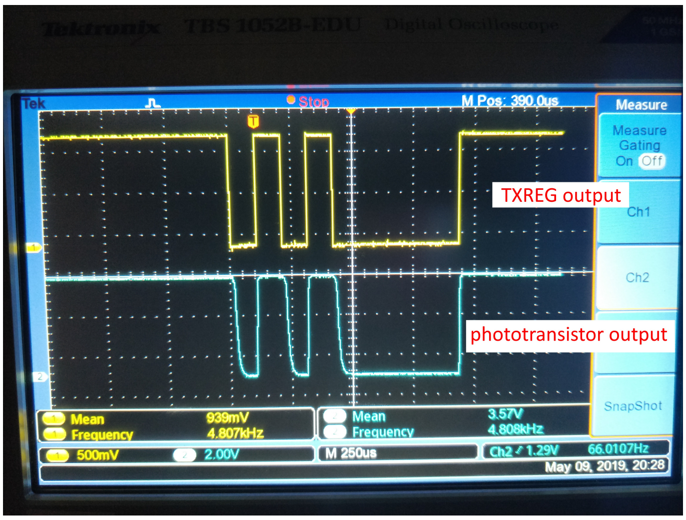


Figure 3.3: Picture from oscilloscope - TXREG and photodiode output

4 Results

In this chapter the result of the design will be stated and how far the project has come during the period of the thesis work. This section will also state if any of the parts was not completely finished and if it was left for Volvo to complete.

4.1 Results from work process

The transmitter was designed as stated in the previous section, see chapter Method, Studies and Thought Process, where the plan was to get one entire step done before moving on to the next one. Each test is a step in how we with a PIC18F1627 programmed both an A/D-converter and a transmitter in the same chip. The A/D conversion took an analog input and converted it to a binary value. This to use the IR-diode high (logic 1) and low (logic 0) function to represent the binary code. To see if the A/D conversion was correct first we represented the binary code with 8 LED's. They where connected to 8 different outputs pin on the microcontroller. After some errors in the code the LED´s showed the correct bit pattern ranging from "00000000" to "11111111". Observe the first complete electric circuit see figure 3.1. The circuit had one input and 8 outputs which later was reconfigured to one output and 8 inputs which where the plane. Programming the microcontroller was a trail and error story, the first program was designed to use PWM-signals, but that idea was scraped early after Test 1 since it could not full fill the requirements.

Instead "bit bashing" was chosen as transfer method. This was done on the TXREG-register, this meant that the bit-string was a serial output that sent one bit at the time in total 8-bits sequences more details on how it was done is written in "Test 2". The outcome was a conversion from analog to serial out thanks to the TXREG and EUSART. This was then further developed in "Test 4" to continuously output a longer message with four sequences per message in. First a start sequence, then channel number, followed by the ADC value and last stop sequence. But first the microcontroller had to be programmed to read from the 8 different channels as stated in the requirements. When planning the code structure for "Test 3" the 8 channel setup was thought of directly so the only thing needed was to add a switch case statement on to the code in "Test 2" to loop through the 8 different channels directly. The only thing chaining was the number for which input pin the microcontroller should read from.

The transmission was completed and got an output that later will be decoded to interpret a temperature from the PT-100 sensors in the PMSM. The transmitting data was first read with a wire as mention in "Test 4" from the microcontroller using MAX232, a serial cable and a PC terminal program "ds30 boot loader". After the value between the microcontroller and ds30 boot loader were correct the IR-transmission was implemented in the same way which gave the same values as before so the conclusion from this was that the transmitter and receiver was completed since the values that were acquired were the same as input ed in the microcontroller as analog input.

4.2 Final circuit

The finished prototype circuit can be found in Appendix 1 A.2. The PIC16F1827 is powered from the top with the black wire as -5V/ground and red wire as +5V. The green wire from the potentiometer 0-5V. On the circuit board there are two separate lines with 5V and ground extension. Which is used to get nice circuit board with shorter and more efficient wire management. When testing we only used one potentiometer as input instead of 8. We did that because we only had one 10k potentiometer. The green wire from the potentiometer dose a voltage division with a resistance which gives the 0-5V interval. This was as as mentioned earlier in the rapport done to simulate the PT-100. Blue cable are analog inputs, so from the green potentiometer wire the voltage goes through the blur cable A1 to pin RA0. The 7 other blue wire are set to grounded which simulates 0V input to the dedicated channel (A2 to A8). The output from the microcontroller is shown with yellow wire. The yellow wire is connected to the input on the IR-diode and after the photo-diode transistor (wireless connection). From the photo-diode the yellow wire goes to pin 11 on the MAX232 chip. Where it converted to the right TTL levels and then it is sent to both the Arduino and DB9 cable output from pin 15. So that we can see that the same outputs is on both computers and is the same as the analog input from the potentiometer. Which after testing they where. When we input 4V at RA0 we get the number 204 (11001100 in binary) at the output so to check if it is the right number we only have to multiple 204 with 0,0196V (204 * 0,0196V = 3.9984 V) which almost exactly 4V as at the input.

4.3 Receiver

The receiver was a basic program that ran on Arduino to see if it got any outputs. The code that was used during the transmission test is linked in Appendix 3 C.2 for the Arduino. This code gave us some input from the receiver and a basic channel read, but it was not fully completed. So the receiver was not completely finished. The plan was to program the Arduino code to store the channel and ADC value in a spreed sheet.

4.4 Induction

The inductor circuit designed in the project was consisting of a simple oscillator at first, this mostly to check how the technology worked since wireless power supplying was a new thing for us. The oscillator was later improved and optimized for our project so it could be able to supply the components of the receiver. The time ran out for us to finish this part so this part was left to other engineers at Volvo to finish, and decided to scrap what was designed up to that point. We found studies and tutorials how to do it but time was not our side. One particular good study was "Inductive Charging for a Self Balancing Robot" and i highly recommended when continuing the induction work. There are also some finished products which can be bought online. But then the circuitry would have to be adjusted to fit the products.

5 Discussion

In this chapter the results and thought will be brought up. But also how to continue the project after this bachelor thesis is over.

To improve this bachelor project is to finish the whole design, this includes the wireless power supply and a circuit board. next up would be to test the finished circuit for any miss calculations in weight distribution of the entire transmitter since it has to have solid weight balance for minimal external centripetal force from the rotation of the axis on the components. What we recommend to continue with first in this project is to design a wireless power supply. This would be designed with a oscillator that would oscillate the DC-voltage to the right voltage levels. The step after that is implement both part to one entire circuit. See so that the inducting part and the transmission can cooperate without interfering with each other. When that is tested the circuit should be sent for construction. We think it would be best to build the circuit in 2 layers since it needs to be space efficient which will help keep the mass centre in the middle of the circuit. Which is important when the circuit has to spin in 20 000 rpm. Example on how to design the circuit would be to have the microcontroller and the inputs on the bottom layer and have solid connection to the top layer where the IR transmitter diode would sit freely and around it the copper wire to receive the wireless inducing magnetic fields. A prototype idea was made to show how we thought the finished product could look, see figure B.2.

We think that the way we worked during this project was optimal. Testing the different parts was a great way to approach the problem since it was easy to see which part worked and which did not. We tried different types of transfer solutions PWM and bit bashing, mostly to see what was the most suitable for this project. We decided that bit bashing was the better way because the PWM could not handle the 8 channels since PWM is just changing its duty cycle depending of the digital value from the A/D-conversion. Bit bashing was also fitting since it could utilize the RS-232 protocol needed to transfer and the 8 different channels.

In the final design we would recommend to have both the transmitter and receiver axial since it would make the transfer more stable. To have it radial is also a possibility but then a recommendation is to put more receivers in a circle so it can follow the transmitter this is to get the most stable transfer for a radial mounting. There is also a possibility to mount more transmitters but then the risk of getting the wrong input is greatly increased and that is why we recommend to mount more receivers if the transmitter is radially mounted.

The induction for the final circuit should be mounted so that the induction coils are wound around the circuit board in the middle. This is because it will be the easiest way to make sure that the mass is in the centre of the axis, see sketch on how we planned the prototype to look in appendix B B.2. To note is that this may create some disturbances for the wireless transfer but since the transfer is based on IR which is light emitting hopefully it will not matter. The only thing that would disturb the transfer is if something is put between the receiver and the transmitter, for example a paper or something that will block the transmission signal, or if the distance is to great between the two parts and the message gets disrupted. We would also recommend a self resonating circuit to get the AC-output that will be on the primary coil since this will get a higher voltage to transfer it to the secondary coil.

Things that could had been done more carefully now when looking back is to plan our time better, to see that the project was to big from the start instead of saying that it could be done, maybe scope down the project right away. This would had led us to not stress as much and for Volvo to maybe get more people involved. We could also have asked for more help, asking people who are specialized in these areas. We asked Göran Hult and he was great help. He had lot of knowledge in microcontrollers and how to program them which saved us many hours problem solving. Anders Mellbin also where a great help when figuring out the electrics for the circuit and when brainstorming for the induction part. But we could have asked others to. Instead of reading forums on the internet we could maybe have asked other professors and engineers at Chalmers and Volvo.

The cooperation between us was good and we completed each others knowledge such as most of the code was written by one person and the hardware design was done by the other and the writing of the report was divided almost equally between both partners. The report were also written during the time of the project and was not postponed until the last week which made it easier since we did not have to stress with the report and could calmly read through the whole thing before turning it in.

6
Conclusion

This chapter we will state what we would continue to do if we had more time to finish and if we would change any time distribution or if we should had focused on other things earlier, but also the conclusion of the whole project and the design that was done.

The work that has been done for Volvo Cars did meet our expectations but of course it is a bit disappointing because we could not finish the whole design in time which lead to prioritizing to finish as many parts as we could. Also we prioritized the most important parts like the data gathering, conversion and the transfer through the diode. The parts that we had the time to finish were the receiver and transmitter, in other words we could send and get the data which were the main part of the thesis work, but the time was not sufficient enough to begin to design a wireless power supply. Therefore we decided to leave that to Volvo Cars to finish.

The planning that we had (the gantt-scheme) showed to be incorrect since some problems with the coding took longer than anticipated which then postponed other tasks. Mostly it took longer time since we tried to design the transmission in two different ways as stated in discussion to see which way was more suitable for this project, the final conclusion was that bit bashing was more suitable then PWM. The receiver was also very basic it could decode all data that we sent from the transmitter but it could not store the data. The transmitter was fully designed with maybe some minor bugs that we could not identify during the project.

But the conclusion that we have is that it is a job that is well executed and after we started the project we realized that it was a bit to big to finish which we stated in the discussion and we are happy with our results. We think that it was necessary to try the different methods to transfer via IR since that gave us knowledge of what was needed for the design. We think that the transmitter and transmission is good enough to be implemented in the final circuit in its current state.

But the overall conclusion is that the task was well executed and with a good results, even when we realized the job was to big to finish. It was a fun experience to be a "real engineer" for once and not only calculate and plan it from the school bench but trying to design something in real life that would have a real application area, it was also a extremely teaching moment for us since we had to try so many different approaches and evaluate different methods to solve the problem. Thanks again Volvo Cars, Chalmers and Anders Mellbin for giving us the opportunity to work with this project.

A

Appendix 1

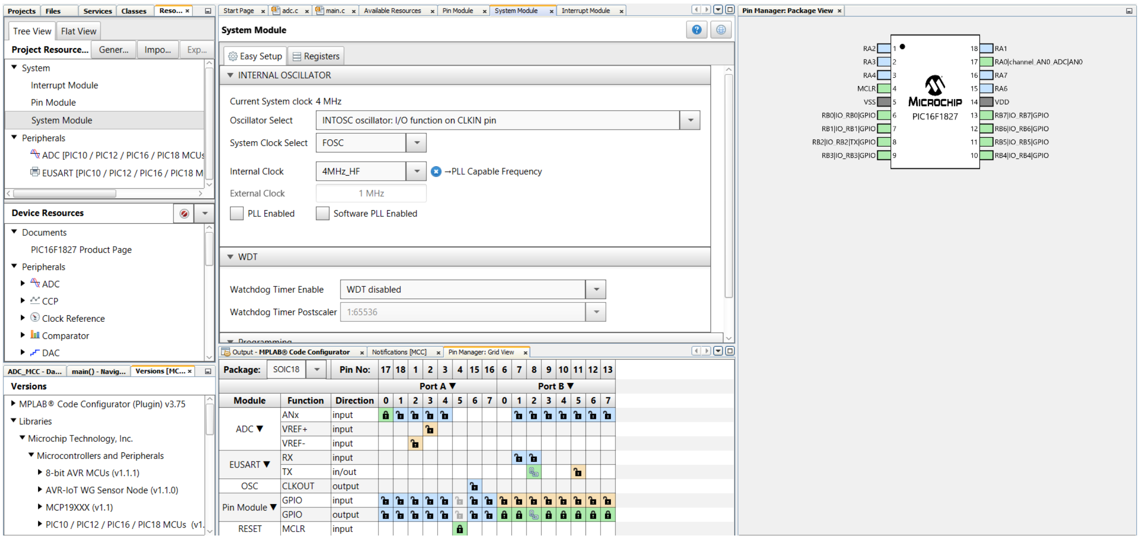


Figure A.1: Screenshot on MPLAB® Code Configurator (MCC) menu.

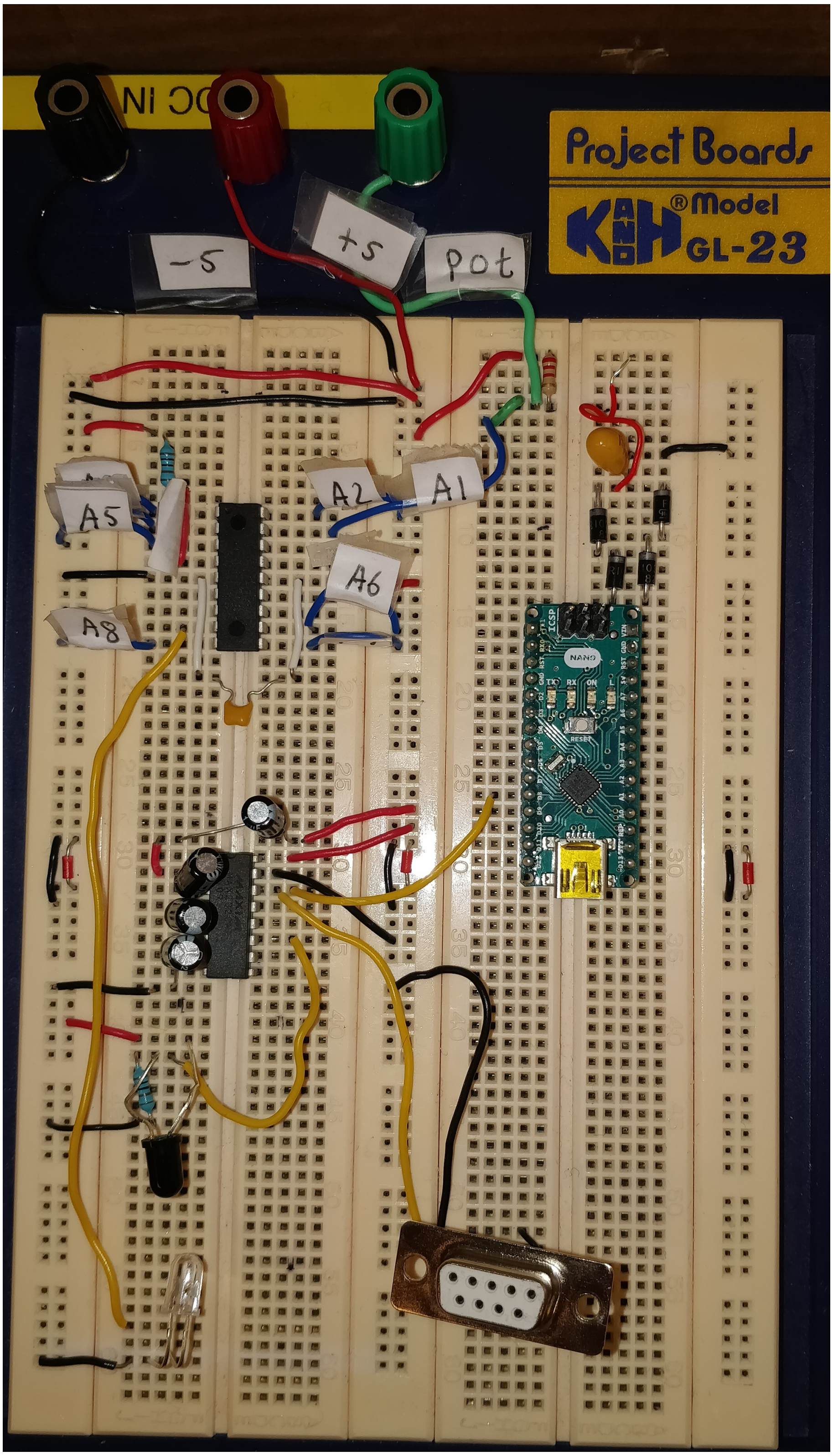


Figure A.2: Circuit

B

Appendix 2

Electrical schematics from KiCad

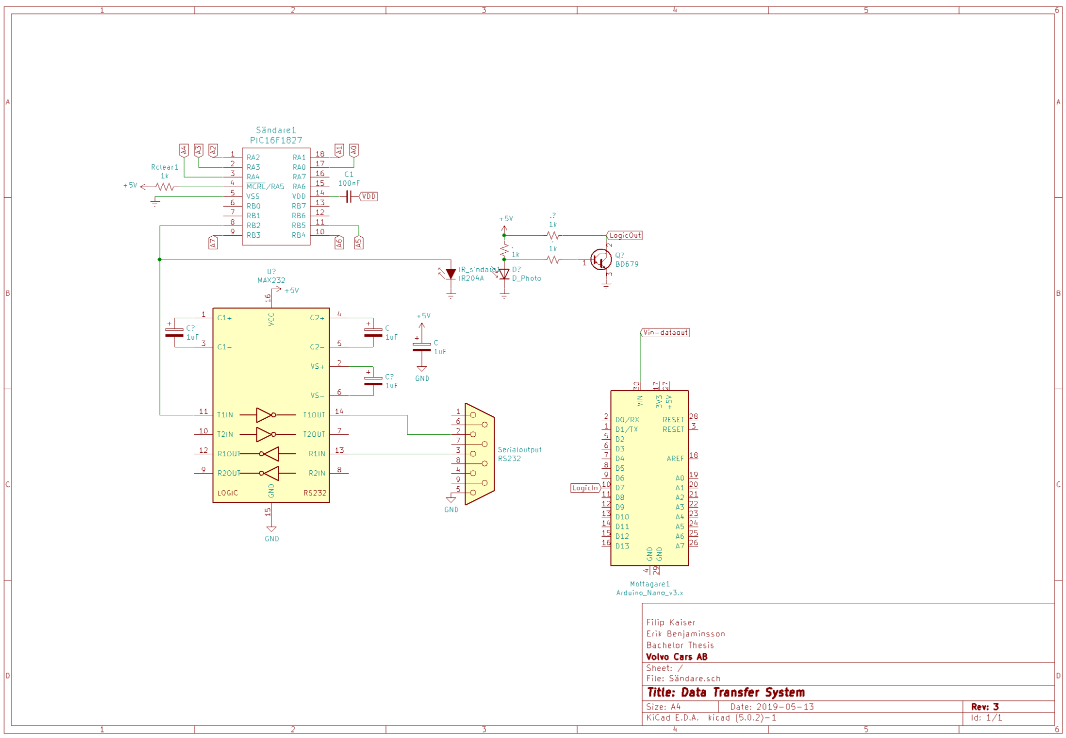


Figure B.1: Whole schematic of the circuit that was constructed for the transmission test.

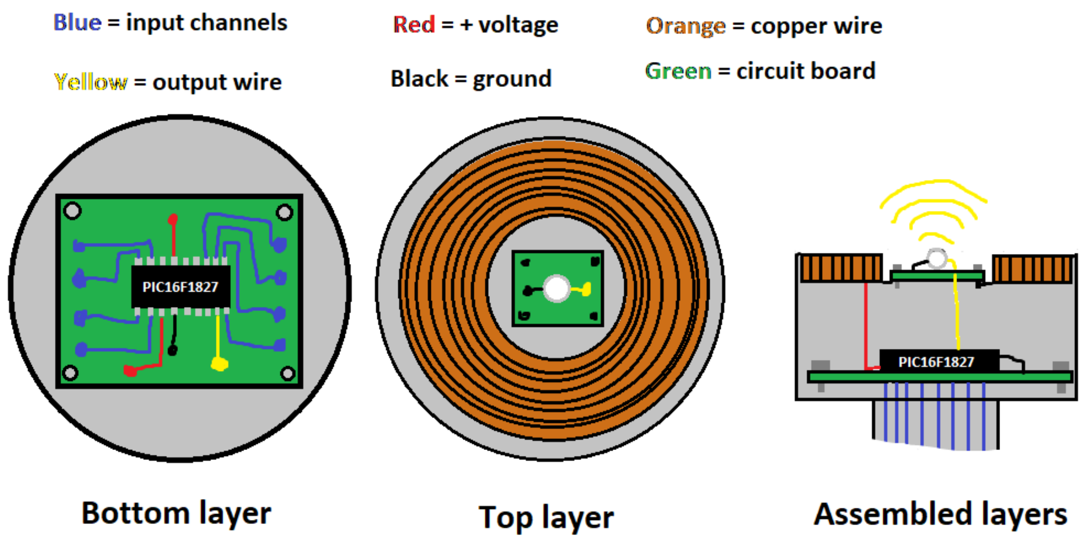


Figure B.2: Prototype on how the finished product could look

C

Appendix 3

C.1 MPLAB code for PIC16F1827

```
/*
 * File:     main_ADC_UART.c
 * Author: FKAISER1
 * Created 2019
 */
//Headerfile-----------------------------------------------------------

#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include <xc.h>
#include <math.h>


// CONFIG 1
#pragma config FOSC = INTOSC     // Oscillator Selection
(INTOSC oscillator: I/O function on CLKIN pin)
#pragma config WDTE = OFF        // Watchdog Timer Enable
(WDT disabled)
#pragma config PWRTE = OFF       // Power-up Timer Enable
(PWRT disabled)
#pragma config MCLRE = ON        // MCLR Pin Function Select
(MCLR/VPP pin function is MCLR)
#pragma config CP = OFF          // Flash Program Memory
Code Protection (Program memory code protection is
disabled)
#pragma config CPD = OFF         // Data Memory Code
Protection (Data memory code protection is disabled)
#pragma config BOREN = OFF       // Brown-out Reset Enable
(Brown-out Reset disabled)
#pragma config CLKOUTEN = ON     // Clock Out Enable (CLKOUT
function is enabled on the CLKOUT pin)
#pragma config IESO = OFF        // Internal/External
```

```
Switchover (Internal/External Switchover mode is disabled)
#pragma config FCMEN = OFF       // Fail-Safe
Clock Monitor Enable (Fail-Safe Clock Monitor is disabled)

// CONFIG2
#pragma config WRT = OFF         // Flash Memory Self-Write
Protection (Write protection off)
#pragma config PLLEN = OFF       // PLL Enable (4x PLL
disabled)
#pragma config STVREN = OFF      // Stack Overflow/Underflow
Reset Enable (Stack Overflow or Underflow will not cause a
Reset)
#pragma config BORV = LO         // Brown-out Reset Voltage
Selection (Brown-out Reset Voltage (Vbor), low trip point
selected.)
#pragma config LVP = OFF         // Low-Voltage Programming
Enable (High-voltage on MCLR/VPP must be used for
programming)
#define baudrate 9600
#define _XTAL_FREQ 4000000  //Klockoscillatorfrekvens
Fosc=4 MHz
// Prototype -----------------------------------------//
void init_EUSART();
void Transmit (unsigned char data);
unsigned int ADC_Read(unsigned char channel);
void init_ADC();
void Channel_Switch();
void Channel_Switch_Full();
void ADC_Channel();
void main();
//------------------------------------------------------ //
void init_EUSART()
{
    //----- Set baud rest register -----
    // Formulas for baud rate
    // Baud rate = Fosc/(16(SPBRG+1)), BRGH=1
    // Baud rate = Fosc/(64(SPBRG+1)), BRGH=0
    // Formulas for SPBRG
    // SPBRG = (Fosc/(16 x Baud rate)) - 1, BRGH=1
    // SPBRG = (Fosc/(64 x Baud rate)) - 1, BRGH=0
    SPBRG = 25;  //Baud Rate generator '

    TXSTA = 0b00100100;
    // TXEN = 1 in TXSTA for transmit enable
    // BRGH = 1 in TXSTA for high speed
    // SYNC = 0 in TXSTA for asynchronus mode
```

```
    RCSTA = 0b10000000;
    // SPEN = 1 in RCSTA for serial port enbale

    PIR1 =  0b00010000; // Waiting until byte can be
    transmitted
    //  TXIF = 1 in PIR1 flag gets set when the data in the
    TXREG

    INTCON= 0b11000000;
    // GEI = 1 in INTCON for enable global interrupt
    // PEIE = 1 in INTCON for Enable peripheral interrupt

    BAUDCON = 0b00000000;
    // SCKP = 0 in BAUDCON for Transmit inverted data to
    the TX/CK pin

}

void init_ADC() // Initialize ADC
{
    ADCON0 = 0b00000001;      //ADON
    ADCON1 = 0b01000000;      //Left written, AD-clock =
    Fosc/4, Vref VDD och VSS
    ADRESL = 0b00000000;
    ADRESH = 0;
}

void init() // Initialize PORTs and settings on PIC
{
    OSCCON =     0b01101000;     // Intern clock 4 MHz
    OSCTUNE =    0b00000000;     // TUN 0;
    BORCON =     0b00000000;     // SBOREN disabled;

    ANSELA =     0b00011111;
    TRISA =      0b00111111;
    // PORTA - (RA0=AN0),(RA1=AN1),(RA2=AN2),(RA3=AN3),(RA4
    =AN4)=analog(1),  (RA5=MCLR)=digital(0)
    // PORTA - (RA0=AN0),(RA1=AN1),(RA2=AN2),(RA3=AN3),(RA4
    =AN4)=input(1),   (RA5=MCLR)=input(1)

    ANSELB =     0b11110000;
    TRISB =      0b11110010;
    // PORTB - (RB6=AN5),(RB7=AN6),(RB5=AN7),(RB4=AN8)=anal
    og(1) ---- (RB2=Tx),(RB3=out)=Digital(0)
    // PORTB - (RB6=AN5),(RB7=AN6),(RB5=AN7),(RB4=AN8)=inpu
```

```
    t(1)  ——  (RB2=Tx),(RB3=out)=output(0)

    LATA =       0b00000000;     // Reset all bit PORTA
    LATB =       0b00000000;     // Reset all bit PORTB
}

void Transmit(unsigned char data) // Transmitt TXREG on Tx
pin
{
    while(!TRMT); // wait for flag
    TXREG = data;  // output
}

unsigned int ADC_Read(unsigned char channel) // Read
voltage and convert ADC
{
    int ad_value = 0;
    ADCON0 = (ADCON0 & 0b10000011)|(channel << 2);
    // ADC-channel
    __delay_ms(1);            //Delay 5us. Macro i HITECH C.
    ADCON0bits.GO = 1;        //AD-converter start
    while(ADCON0bits.GO);     //Wait for AD-convertion to
    finish
    ad_value = ADRESH;
    return ad_value;            //Return 8 MSB of
    AD-convertion
    //return ((ADRESH<<8) +ADRESL); //Returns 10 bit Result
}


void ADC_Channel(int adc_value, int channel) // Set
transmission que
{
    int start = 0b11111110; //
    int stop =  0b11111000; //
    int idle =  0b00000000; //
    const int delay2 = 100;

    // Send start bit sequens
    Transmit(start);
    __delay_ms(delay2);

    // Send selected Channel
    Transmit(channel);
    __delay_ms(delay2);
```

```
    // Send ADC channel value
    Transmit(adc_value);
    __delay_ms(delay2);

    // Send stop-bit sequense
    Transmit(stop);
    __delay_ms(delay2);

    // 400 ms dealy total so 8 channels take 400*8 = 3200
    ms
    // for 8 channels to tranmitt
}

void main()
{
    int ADC_Value_Result = 0;
    int Channel_Select = 0;
    int Channel;
    const int delay = 100;
    int AN_pin;

    // Initialize
    init_EUSART();
    init_ADC();
    init();

    while (1)
    {
        switch(Channel_Select)
            {
                case 0: // Channel 1 = RA0
                    AN_pin = 0; // AN0
                    Channel = 1;
                    break;

                case 1: // Channel 2 = RA1
                    AN_pin = 1; // AN1
                    Channel = 2;
                    break;

                case 2: // Channel 3 = RA2
                    AN_pin = 2; // AN2
                    Channel = 3;
                    break;

                case 3: // Channel 4 = RA3
```

```
                AN_pin = 3; // AN3
                Channel = 4;
                break;

            case 4: // Channel 5 = RA4
                AN_pin = 4; // AN4
                Channel = 5;
                break;

            case 5: // Channel 6 = RB5
                AN_pin = 7; // AN7
                Channel = 6;
                break;

            case 6: // Channel 7 = RB4
                AN_pin = 8; // AN8
                Channel = 7;
                break;
            // ** add more channels if needed **

            default: // Channel 8 = RB3
                AN_pin = 9; // AN9
                Channel = 8;
                break;
        } // end switch case

        // Get adc from input pin
        ADC_Value_Result = ADC_Read(AN_pin);

        // Send channel and ADC to output
        ADC_Channel(ADC_Value_Result, Channel);

        // to next channel
        if (Channel_Select < 7)
            {
                Channel_Select++;
            }
        else
            {
                Channel_Select = 0;
            }

        // For IR transmission set TTL volt

    } // infinity loop
} // end script
```

C.2 Arduino IDE code for Arduino Nano

```
#include <ctype.h>

#define bit9600Delay 100
#define halfBit9600Delay 50

#define bit4800Delay 188
#define halfBit4800Delay 94

#define bit2400Delay 416
#define halfBit2400Delay 208

byte rx = PD4;
byte SWval;
byte CHval;
int offset;

void setup() {
  pinMode(rx,INPUT);
  Serial.begin(2400);
}

int SWread()
{
  byte val = 0;
  while (digitalRead(rx));
  //wait for start bit
  if (digitalRead(rx) == LOW) {
    delayMicroseconds(halfBit2400Delay);
    for (offset = 0; offset < 8; offset++) {
     delayMicroseconds(bit2400Delay);
     val |= digitalRead(rx) << offset;
    }
    //wait for stop bit + extra
    delayMicroseconds(bit2400Delay);
    delayMicroseconds(bit2400Delay);
    return val;
  }
}

int Channel()
{
  char channelnr;
  byte channel =0;
  while(digitalRead(rx));
```

```
  if (digitalRead(rx)==LOW){
  for(offset = 0; offset < 3; offset++){
    delayMicroseconds(bit2400Delay);
    channel |= digitalRead(rx) << offset;
  }
    switch (channel){
      case 0: channelnr =1;
              break;
      case 1: channelnr =2;
              break;
      case 2: channelnr =3;
              break;
      case 3: channelnr =4;
              break;
      case 4: channelnr =5;
              break;
      case 5: channelnr =6;
              break;
      case 6: channelnr =7;
              break;
      case 7: channelnr =8;
              break;
    }
      return channelnr;
  }
}


void loop()
{
    CHval = Channel();
    Serial.print("CHANNEL:␣␣");Serial.print(CHval);
    delay(100);
    SWval = SWread();
    Serial.print("V rde:"); Serial.println(SWval);
    delay(100);
}
```

Bibliography

[1] Adaptive speed control for permanent-magnet synchronous motor system with variations of load inertia, 2009, IEEE Transactions on Industrial Electronics, Li, Shihua and Liu, Zhigang, 8, 3050–3059, 56, IEEE, 0278-0046, 10.1109/TIE.2009.2024655, 02780046, Adaptive control, Extended state observer (ESO), Fuzzy inferencer, Inertia identification, Permanent-magnet synchronous motor (PMSM), Speed regulation.

[2] Parameters Using the Single-Phase Test, 1999, Energy, Gastli, A., 1, 51–56, 14, 0885-8969, 10.1109/60.749147, 08858969, equivalent circuit, induction motor, single-phase test.

[3] Permanent-magnet synchronous motor magnet designs with skewing for torque ripple and torque reduction, 2009, IEEE Transactions on Industry Applications, Islam, Rakib and Husain, Iqbal and Fardoun, Abbas and McLaughlin, Kevin, 1, 152–160, 45, IEEE, 1424403642, 10.1109/TIA.2008.2009653, 00939994, 18707883, Back-EMF (BEMF) harmonics, Phase advancing, Phase lagging, Step skew, Torque ripple.

[4] What is a RTD?, Omega, UBID., [Online] Available:https://www.omega.com/prodinfo/rtd.html

[5] What's the Difference Between PT1000 vs 100? , Omega, Ubid., [Online], Available: https://www.omega.com/prodinfo/rtd- pt100-vs-pt1000.html.

[6] IR-dioder 10-pack, Ubid., [Online], Available: https://www.kjell.com/se/sortiment/el-verktyg/elektronik/optokomponenter/ir-dioder-fototransistorer/ luxorparts-ir-dioder-10-pack-p89281.

[7] PIC16F1827, Microchip Technology Inc., 2019, Available: https://www.microchip.com/wwwproducts/en/PIC16F1827.

[8] PIC16F1827 data sheet, 2010, Microcontrollers, Pin Flash and Technology, L, 4, 16, Microchip Technology Inc., http://ww1.microchip.com/downloads/en/DeviceDoc/41303G.pdf, United States of America, 9781609327347.

[9] MPLAB X IDE, Microchip, 2019, [Online], https://www.microchip.com/mplab/mplab-x-ide.

[10] Dataman mainpage, Dataman, 2018, [Online], Available: https://www.dataman.com/.

[11] KiCAD Main Page, KiCAD, 2019, [Online], Available: http://kicad-pcb.org/.

[12] Arduino Main Page, Arduino, 2019, [Online], Available: https://www.arduino.cc/.

[13] Vad är RS-232? -Den ultimata guiden, Induo, 2 Oktober 2018, [Online], Available: https://www.induo.com/b/vad-ar-rs-232/ .

[14] RS232 Protocol-Basics, Electronicshub, 1 Juli 2017, [Online], Available:https://www.electronicshub.org/rs232-protocol-basics/.

[15] Schmitt, Ron, Changing Magnetic Fields, Electromagnetics Explained: A Handbook for Wireless/ RF, EMC, and High-Speed Electronics EDN Series for Design Engineers Electronics & Electrical Referex collection.Electrónica y electricidad, Newnes, 2002, ISBN: 0750674032, 9780750674034, 75-90.

[16] Inductive charging for a self-balancing robot, 2016, Bååth, Ola. Carlsson, Fredrik, 8–12, 19–21, 1, Available:http://publications.lib.chalmers.se/records/fulltext/239184/239184.pdf, Department of Energy and Environment, Gothenburg, Power Electronics.

[17] What is the difference between oscillator and inverter?, Quora, 10 Mars 2016, [Online], Available:https://www.quora.com/What-is-the-difference-between-oscillator-and-inverter.

[18] Data sheet MAX232, Texas Instruments, 1 November 2018, [Online], Available:http://www.ti.com/lit/ds/symlink/max232.pdf, United States of America.

[19] Logic Signal Voltage Levels, All about circuits, UBID., [Online], Available:https://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/.

[20] What is a DB9 connector?, L-com.com, UBID., [Online], Available:http://www.l-com.com/what-is-a-db9-connector.