



Univerza v Mariboru

*Fakulteta za elektrotehniko,
računalništvo in informatiko*

Andrej Božnik

ELEKTRONSKO VEZJE ZA POGON MOTORJEV HUMANOIDNEGA ROBOTA Z UPORABO CAN BUS

Diplomsko delo

Maribor, November 2011

Diplomsko delo univerzitetnega študijskega programa

**ELEKTRONSKO VEZJE ZA POGON MOTORJEV HUMANOIDNEGA ROBOTA
Z UPORABO CAN BUS**

Študent: Andrej Božnik
Študijski program: UN ŠP - Elektrotehnika
Smer: Mehatronika
Mentor: Miran Rodič, doc. dr.
Somentor: Karl Gotlih, izr. prof. dr

Maribor, November 2011

UNIVERZA V MARIBORU



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Številka: 93575550

Datum in kraj: 30. 11. 2011, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. l. RS, št. 01/2010)
izdajam

SKLEP O DIPLOMSKEM DELU

1. **Andreju Božniku**, študentu univerzitetnega študijskega programa ELEKTROTEHNIKA, smer Mehatronika, se dovoljuje izdelati diplomsko delo pri predmetu Servosistemi.
2. MENTOR: doc. dr. Miran Rodič
MENTOR: izr. prof. dr. Karl Gotlih
3. Naslov diplomskega dela:
ELEKTRONSKO VEZJE ZA POGON MOTORJEV HUMANOIDNEGA ROBOTA Z UPORABO CAN BUS
4. Naslov diplomskega dela v angleškem jeziku:
ELECTRONIC CIRCUIT FOR A HUMANOID ROBOT MOTOR DRIVER USING CAN BUS
5. Diplomsko delo je potrebno izdelati skladno z "Navodili za izdelavo diplomskega dela" in ga oddati v treh izvodih (dva trdo vezana izvoda in en v spiralo vezan izvod) ter en izvod elektronske verzije do 30. 11. 2012 v referatu za študentske zadeve.

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 3 delovnih dni.



Obvestiti:

- kandidata,
- mentorja,
- somentorja,
- odložiti v arhiv.

Elektronsko vezje za pogon motorjev humanoidnega robota z uporabo CAN Bus

Ključne besede: pogon motorja, načrtovanje, komunikacijski protokoli, CAN, humanoidni robot.

UDK: 621.3:681.5(043.2)

Povzetek

V tem diplomskem delu se predstavi načrtovanje elektronskega vezja. To je zasnovano na komunikacijskem protokolu CAN bus. Oblikovano je za uporabo v humanoidnem robotu pri krmiljenju sklepa (prostostne stopnje mehanizma). Načrtovani krmilnik motorja omogoča hkratno priključitev večih odjemalcev (identičnih krmilnikov), kjer poteka komunikacija na istem komunikacijskem protokolu (delitev z ostalimi napravami).

Preizkus delovanja sistema je narejen tako da se krmilniku pošljejo različne zelene vrednosti položaja. Zelena vrednost položaja je dosežene preko krmilnika z različnimi vrednostmi hitrosti. Krmilnik zagotavlja nadzor nad delovanjem sistema tudi ko je motor obremenjen. Ker je ta krmilnik zasnovan za humanoidnega robota, ta obremenitev odraža nestabilnost pri različnih postavitvah robota (obremenitev na sklepu robota).

Na koncu so predstavljeni rezultati in ugotovitve načrtovanega vezja za pogon motorjev.

ELECTRONIC CIRCUIT FOR A HUMANOID ROBOT MOTOR DRIVER USING CAN BUS

Key words: Motor Driver, Design Process, Communication Protocols, CAN, Humanoid Robot.

UDK: 621.3:681.5(043.2)

Abstract

In this thesis, the design of the motor driver electronic board is presented. The motor driver is designed on a Control Area Network bus based communication network. The motor driver is aimed to be used in a humanoid robot as controlling the joint actuators. The designed motor controllers in this project have the capability to be used in a network by multi clients in order to provide different joints control on a common bus.

The performance evaluation test of system is made by commanding different desired positions to the controller. The desired positions are approached by the controller by different velocities. The controller is designed in order to provide controllability for the system even in the case of load unstability of the joint. Since the motor controller is aimed to be used for a humanoid robot, the load reflected unstability causes the variation of the overall pose of the robot.

Finally, the results are compared and discussed for the evaluation of the designed motor controller.

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisors Professor Miran Rodič and Karl Gotlih, which have been particularly helpful on various aspects of my professional and career development.

Thanks to everyone on the Institute of Handling Devices and Robotics at Vienna University of Technology for their hospitality. I am grateful to Professor Peter Kopacek, for inviting me to participate in the project. Special thanks goes to Dr. Ahmad Byagowi, who helped me a lot through my work. His unique ability to explain complex concepts and share enthusiasm with me – a quality I hope will be a lifetime companion. It was both an honor and privilege to work with both of them. My days in the laboratory would not have been the same without Mr. Peter Unterkreuter. Thanks to Fatih, Jascha, Werner and Theresa for many interesting technical and non-technical discussions.

Finally, my sincere thanks go to my parents for their continual help and support inside and outside professional environment during the course of my studies.

TABLE OF CONTENTS

Acknowledgments.....	V
Table of Contents.....	VI
List of Symbols.....	VII
List of Acronyms.....	VIII
1 Introduction.....	1
1.1 Purpose of a Thesis.....	3
1.2 Archie.....	4
1.3 Control System.....	5
1.4 Chapter Outline.....	7
2 Literature Review.....	8
2.1 Data Bus Communication Networks.....	8
2.2 CAN (Controller Area Network).....	10
2.2.1 OSI Model.....	10
2.2.2 Electrical Topology.....	11
2.2.3 CAN Message Frame Format.....	14
2.2.4 CAN-based Higher Layer Protocol.....	15
3 Motor Actuators.....	17
3.1 Mathematical Model of the DC Motor.....	17
3.2 Motor Based Joints.....	20
4 Implementation.....	25
4.1 PCB Design Process Overview.....	26
4.2 Motor Controller's Requirements.....	28
4.3 Choosing the Components.....	29
4.4 Brushless DC Motor Driver.....	30
4.5 Brushed DC Motor Driver.....	32
4.5.1 Microcontroller AT90CAN128.....	34
4.5.2 H-Bridge.....	36
4.5.3 Sensing Detection.....	38
4.5.4 Communication Interfaces.....	40
4.6 Design Evaluation and Discussion.....	42
5 Simulation and Experimental Results.....	49
5.1 Experimental Setup.....	49
5.2 Experimental Results.....	52
6 Conclusion.....	60
Appendix A: Comparison List.....	63
Appendix B: Circuit Schematics.....	67
Appendix C: Bill of Materials.....	69
Appendix D: The PCB Layout Drawings.....	72

LIST OF SYMBOLS

Symbol	Subject
a	Acceleration
b	Damping factor
i_a, I_a	Inductor current
J	Moment of inertia
k_b	Back electro-motive force (BEMF) constant
K_I	Integral gain
K_P	Proportional gain
k_t	Torque constant
L	Inductance
m	Mass
R_a	Resistance of the motor armature
U_{dif}	Differential voltage
V_{BEMF}	Back electro-motive force voltage
V_{CC}	Supply voltage
V_{La}	Motor armature inductance voltage
V_{Ra}	Motor armature resistance voltage
θ_M	Motor angular position
$\dot{\theta}_M$	Motor angular velocity
T_b	Friction torque
T_J	Torque from angular acceleration
T_L	Load torque
T_M	Motor torque

LIST OF ACRONYMS

ACK	Acknowledgement
ADC	Analog-to-Digital Converter
AGV	Automated Guided Vehicle
BEMF	Back Electro-Motive Force
CAN	Control Area Network
CCP	CAN Calibration Protocol
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DC	Direct Current
DLC	Data Length Code
DOF	Degrees of Freedom
ECAD	Electronic Computer Aided Design
ECU	Electronic Control Unit
EMI	Electromagnetic Interference
EOF	End Of Frame
FPGA	Fundamental Programmable Gate Array
IC	Integrated Circuit
IDE	Identifier Extension
IFS	Intermission Frame Space
IMU	Inertial Measuring Unit
I/O	Input/Output
ISP	In-system Programming
JTAG	Joint Test Action Group
LAN	Local-Area Network
LIN	Local Interconnect Network
MCAD	Mechanical Computer Aided Design
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
HLP	Higher Layer Protocol
OSI	Open System Interconnection
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
RTR	Remote Transfer Request
SOF	Start Of Frame
SPI	Serial Peripheral Interface
SRR	Substitute Remote Request
xCP	Universal Measurement and Calibration Protocol

1 INTRODUCTION

Nowadays involvement of intelligent machines increases demands. With new problems and solutions presented, fast response is required. Over the years robots have replaced humans in many workplaces with repetitive tasks.

According to Oxford English Dictionary, [1] human machines known under the word 'robot' have been first time introduced by the Czechoslovakian dramatist Karel Capek in his play R.U.R. ('Rossum's Universal Robots'), published in 1920.

There are many different types of robots used for various purposes in different environments. They can be divided into two main categories, namely stationary, and mobile types of robots. Stationary (*i.e.*, fixed base) robots are widely used in an industrial environment. Mobile robots can be further divided into three sub categories: wheeled, legged robots, and others (e.g., snake-like, swimming, and flying robots). For instance, wheeled robots are used in non-industrial (*e.g.*, robotic vacuum cleaners, hobby robots, military robots) and as well in industrial environments (*e.g.*, automated guided vehicles - AGVs).

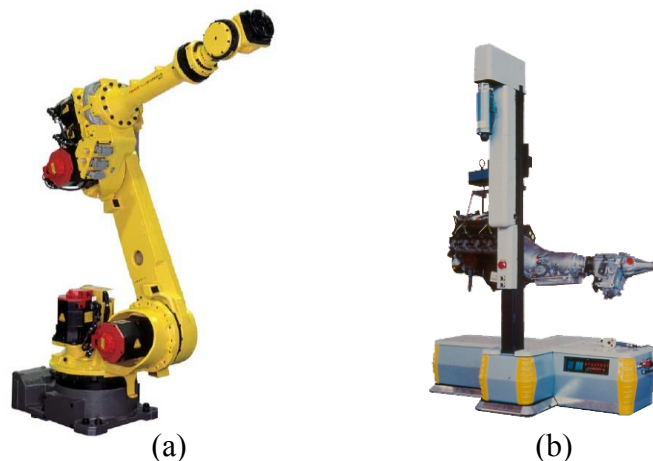


Figure 1-1: (a) Stationary robot arm [2], (b) mobile AGV robot [3]

Industrial robots operate in precisely structured environments, which use preprogrammed control.

Some robots are controlled remotely by humans (called teleoperators) for performing work in hazardous areas and especially in medicine (surgical robotics). With a combination of human control (teleoperation) and automation-assisted control, more complex task can be performed. Those robots, called telerobots have the ability to perform autonomous work and are useful for semi-structured (or unstructured) environments.

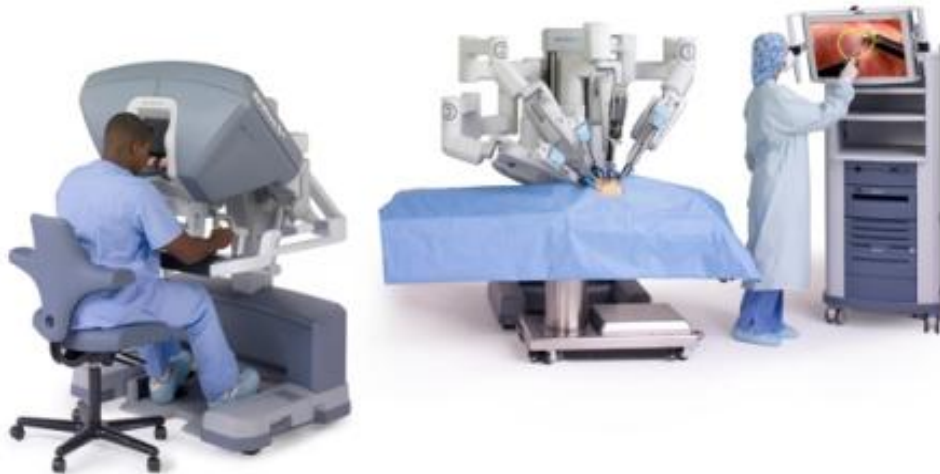


Figure 1-2: Robot surgery system [4]

Complex robots with human like characteristics, better known as humanoid robots, require a computer-based control system or powerful control unit for the effective utilization of the advanced technology. Humanoid robots differ from any other kinds of robots that can be used in a wide variety of industrial applications (*i.e.*, industrial robots).



Figure 1-3: Humanoid robot [5]

Humanoid robotics is focusing in the field of artificial intelligence where robots could perform non-repetitive tasks in semi-structured (or unstructured) environments. Such robots are neither so rare nor so common to rapid development as predicted years ago as they are more difficult to design and control.

In Springer's Handbook of Robotics [6], some definitions of humanoid robots can be spotted. For instance: *“Compared to other robots, a humanoid robot is a robot with its overall appearance based on the human body. These machines are capable of bipedal (meaning two feet) movement and manipulation. Two legs (biped) robots are more rare than a four to six legs (multiped) robots are. A humanoid robot can take advantage of their entire body (e.g., lifting objects while moving the body to compensate for the load). It allows interaction with made-for-human tools or environment. To deal with (nonlinear) stability problem, a robot walking on two legs requires external sensors and control systems that are more complex.”*

Humanoid robots are still a long way from being totally autonomous and independent, with human characteristics and intelligence during walking over different surfaces. In fact, it is not so easy and it is a technical challenge to obtaining a stable walking motion similar to human gait. Human walking is complex task, which requires the coordination of different parts of human body. With a humanoid robot, the regulation of whole body represents a mechanical problem that requires the coordination of a number of different joints. Humanoid robots require high level of intelligence in dynamic environments and complex electronics, which is prone to failure. Developing humanoid robots with more sophisticated human-like physical features requires a specific hardware design and introduces more innovative approaches during the development process.

1.1 Purpose of a Thesis

The purpose of this project is to design a motor driver for a Humanoid robot. The electronic circuit boards have been specifically designed for an existing humanoid robot and are based to run brushed and brushless DC motors. The communication between motor driver and the central controller in this project is designed based on a standard Control Area Network (CAN).

1.2 Archie

This work is part of a research project done through academic year 2009/2010, focused on electronic design prototyping for a humanoid robot called Archie. Archie is the flagship project [7] between Technical University of Vienna (Austria) and University of Manitoba (Canada) since 2004, based on an aluminum skeleton. It has thirty Degrees of Freedom (DOF), including seven DOFs at each leg (one DOF in the ankle, one in the knee, toe-and-heel, and three in each hip). Actuation is achieved by a combination of brushed and brushless DC and RC servomotors.

Table 1-1: Data sheet of Archie

DOFs		30
Size (Height without head)		1.2 m
Weight (without head)		Approximately 20 kg
Power supply		Lithium Ion Battery
Sensors		Gyroscope, Accelerometer, and Encoders
Actuation	Upper part	RC servo motors
	Lower part	- Brushed DC motors combined with planetary gear drive - Brushless DC motors combined with harmonic gear drive
Walking speed		Dynamically at a minimum of 0.5 m/s
Material of the body		Aluminum

In related work [7] Byagowi describes the control system implemented on Archie humanoid robot. He presents simulation and experimental results that demonstrate the effectiveness of the robot and the control algorithms used on it. In context of the motor driver development, his prior work experiences with the robot are used for improvements and redesign of an electronic design.

1.3 Control System

The control system of Archie consists of two levels, a central controller and joint controllers (divided in three types of motor controllers, a servomotor, brushed and brushless DC motor controller). Joint controllers are connected between each other and controlled from the central controller unit using high-speed network where each joint in a robot uses his own individual motor driver (see Figure 3-3). All the data from the joints are further processed and the calculations with appropriate command are sent back to the joint controllers that are used for the balancing robot [7]. Joints are synchronized with the central controller (refresh rate 1 millisecond), which handles the high-level control tasks. Each joint (driven by electric motor) in a robot has ability to follow complex paths. Motion of the joint (*e.g.*, torque, velocity and the position) is monitored with motor driver.

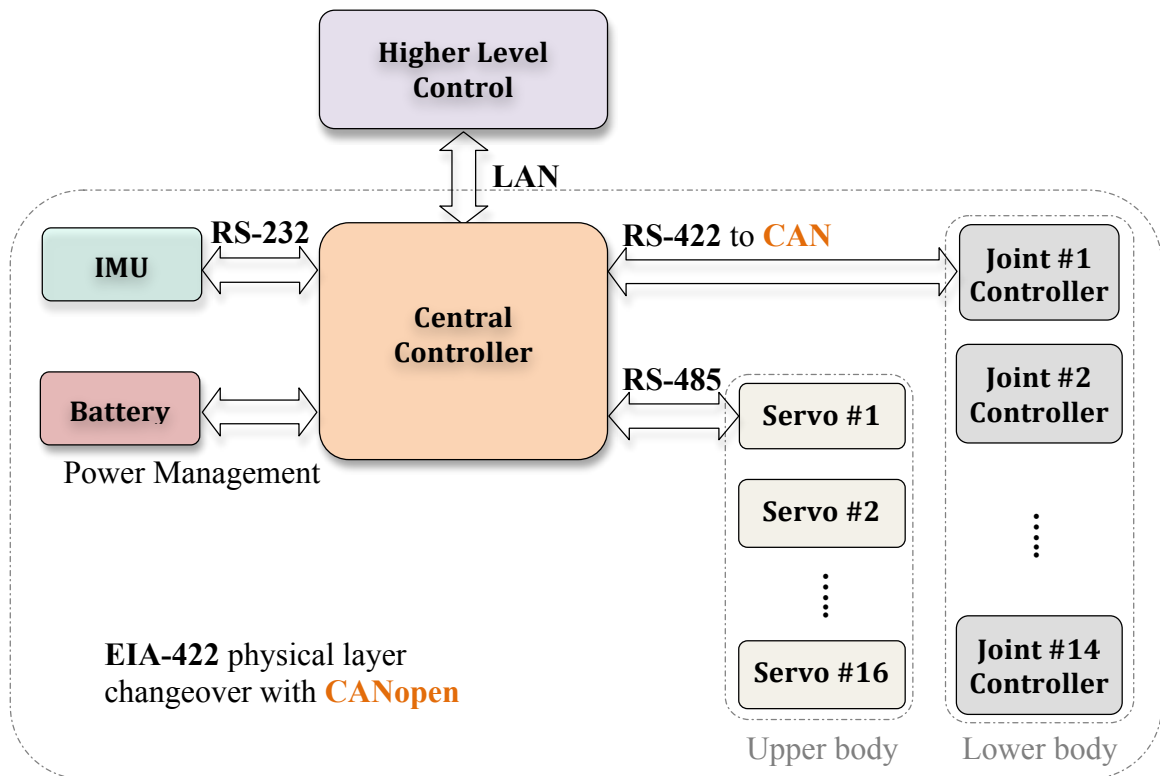


Figure 1-4: Control system block diagram, based on [7]

The Inertial Measuring Unit (IMU) senses 3D motion and sends feedback data to central controller. Central controller processes data (position of the motor) from each individual joint controller and tries to balance a humanoid robot during a walking gait.

Hierarchical design of decentralized control structure is implemented in embedded system. Where the network stays free of unnecessary messages, and each joint controller is assigned for a specific task.

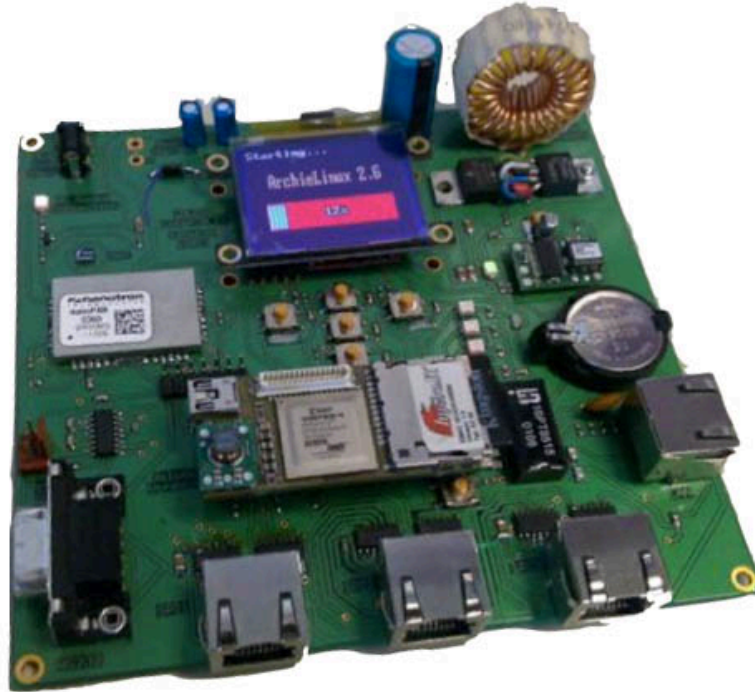


Figure 1-5: Central controller [7]

The central controller is based on the Xilinx Virtex 4™ family of FPGAs^I using the XC4VFX12 (FX-series) package. The FPGA contains hardcore 32-bit embedded PowerPC 405 processor, running the Linux operating system to control the entire robot. Three Master serial peripheral interface (SPI) units are implemented for the data bus on a custom-made printed circuit board (developed by A. Byagowi) with an RS-422 physical layer (EIA-422). On each of these three communication buses a maximum of 7 clients can be connected due to limitations^{II} from the communication protocol. For sending parameters to the central controller, for evaluation of robot behavior and for simulation needs, the board is equipped with a Local-Area Network (LAN) connection.

^I - Acronym of Fundamental Programmable Gate Array

^{II} - The RS-422 protocol supports up to 10 clients, where the actual number depends on a predetermined bit rate, which is calculated from the number of clients (C), packet size (P), the amount of data packets required to exchange data between each client (A), and desired refresh rate (R). Therefore: (bit rate) = (C x P x A) / R

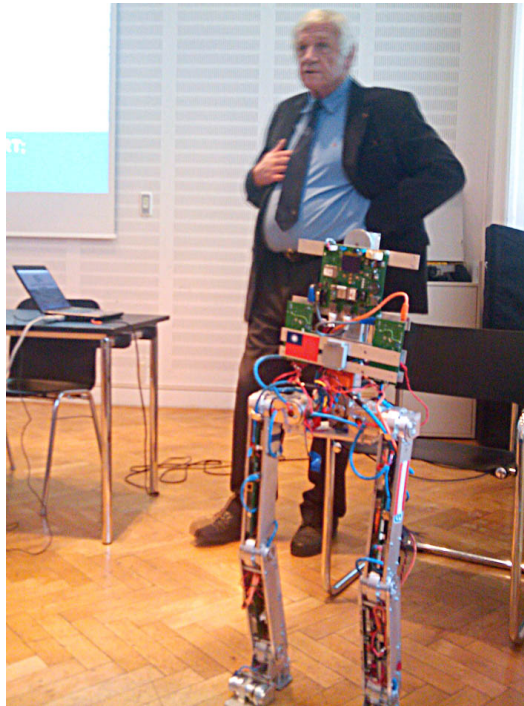


Figure 1-6: Archie presented at the Austrian workshop of autonomous mobile robots

1.4 Chapter Outline

The remainder of thesis is divided as follows. Chapter 2 presents several communication protocols and continues by explaining CAN protocol. In Chapter 3 the mathematical model of the DC motor is described. It then presents the relevant information about motors used in the joints of a humanoid robot. Chapter 4 describes the electronic circuit design implementation and guides us with step-by-step explanation through the design of a motor driver. It then discusses approaches to deal with the noise that could affect the operation of the circuit. Chapter 5 gives a presentation of the motor driver laboratory testing. Finally, Chapter 6 sums up and looks into some ideas for the future developments on the motor driver.

2 LITERATURE REVIEW

The following section shows an overview of network protocols and bus systems, which can be used for electronic networking in distributed systems. Communication itself, between the control systems can slow things down (and communication errors can occur), so it is important to select most appropriate one in the context of robotic manipulation tasks.

2.1 Data Bus Communication Networks

Although Archie had been fully operational in the past, requirement during the course of project have been to redesign motor drivers and made them suitable for a new communication protocol. As stated in Chapter 1.3, the central controller uses three SPI buses for connecting the joints in lower body together.

SPI – is a clocked serial link [8] where devices communicate in a master/slave mode. The SPI communication allows simultaneous transmission and reception of data messages (full duplex). Protocol does not specify a maximum bit rate and is not intended for the distant communication. The SPI is an unbalanced system, and as such, is more prone to ground potential problems.

The objective is to replace the RS-422 physical layer protocol implemented in lower body of the robot to change the whole network to a multi-master mode, so that other joint controllers can directly communicate with each other.

RS-422 (EIA-422) – specifies the physical layer (electrical characteristics) with data transmission up to 10 Mbit/s. Devices using RS-422 protocol communicate in a master/slave mode. RS-422 is driven with balanced differential signal and as such is immune to electrical interference (EMI).

The modern vehicles electrical systems use advanced and resistant to interferences communication protocols, which are relatively close to requirements required in this project. In automotive industry there are a variety of communication networks used for controlling different hardware across the system.

The most popular wired networks that are relevant to the project are described below [9], [10], [11].

CAN – originally developed for the serial data exchange between electronic control units in automobiles with a very high level of security. Devices are driven with balanced differential signal and communicate in a multi-master mode. It is used in industrial automation systems with microcontroller networks where it can maintain data transmissions of 1 Mbit/s.

LIN^{III} – is a low cost serial communication system, developed especially for the cost-efficient communication of intelligent sensors and actuators in automobiles. Devices communicate in a master/slave mode. LIN may be used as a sub-network for a CAN bus. Protocol itself is slow with transmissions up to 40 kbit/s.

FlexCAN – is based on the CAN protocol, to support deterministic and security relevant applications. With data transmission up to 10 Mbit/s and improved reliability, safety and security, FlexCAN is suitable for embedded and industrial applications as the communication protocol.

EtherNet/IP^{IV} – is used in industrial automation systems where the main focus is on transmission of periodic signals. It uses Ethernet for physical layer and can handle large amount of data (1 Mbit/s to 10 Gbit/s), where devices communicate in a master/slave mode. In automation technology applications is less suitable due to lack of the real-time capabilities and very short error recovery times of CAN.

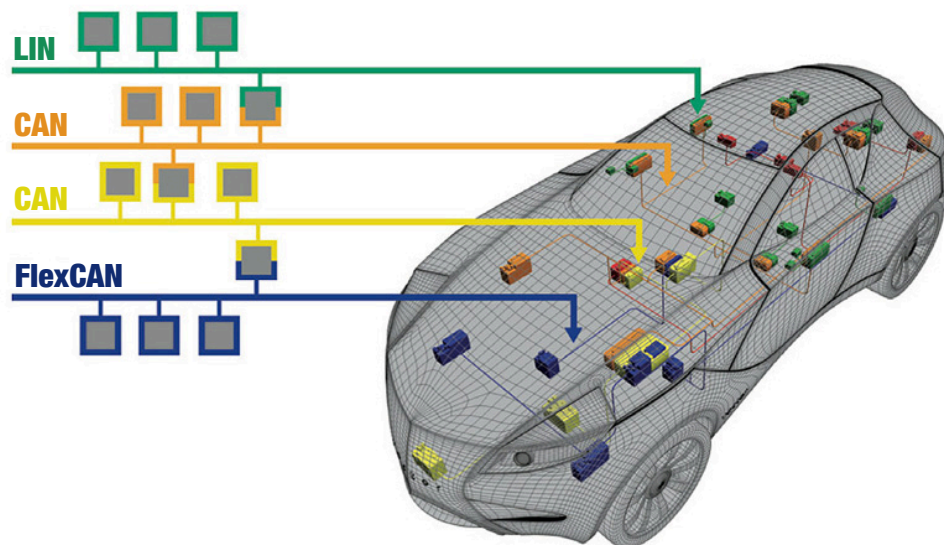


Figure 2-1: Integration of different communication protocols in automotive industry [12]

^{III} - Acronym of Local Interconnect Network

^{IV} - Acronym of Ethernet Industrial Protocol

Overall, CAN has been chosen as it fits in best for the implementation in an existing hardware since it requires only minor changes of hardware in a particular robot configuration (further explained in Chapter 3.2).

2.2 CAN (Controller Area Network)

CAN bus [9] is a serial communication bus for real-time control applications with a network of independent controllers, first introduced in 1986 by Bosch and Intel. The version of the current standard (i.e., CAN 2.0) has been in use since 1990.

The most common applications [9] where CAN is in use are related to automotive and avionics industry, robotics and industrial automation, motor and process control, and medical equipment. Primarily is used in embedded systems, where robustness, high reliability and error resistant of the protocol are required.

2.2.1 OSI Model

CAN is based on *OSI model* [10]. The OSI, or Open System Interconnection, seven-layer model defines how messages are transferred and performed by each layer in a network.

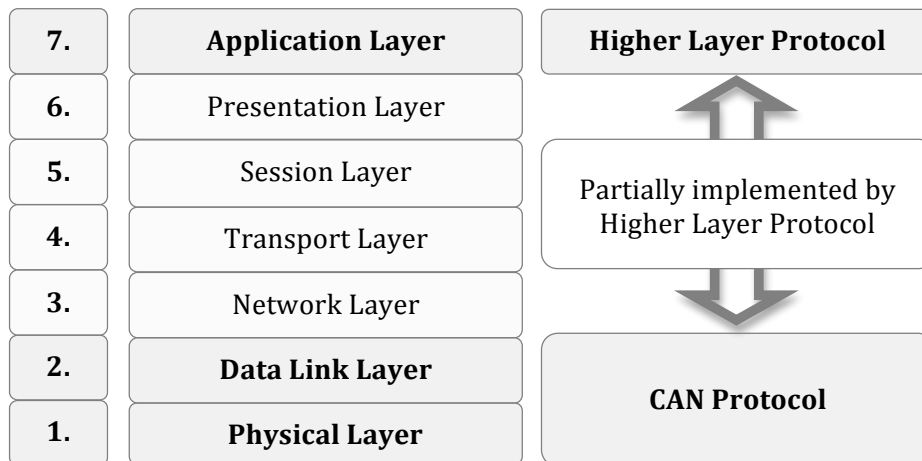


Figure 2-2: OSI reference model

Not full seven-layer communication is required. An *Application Layer*, *Data Link Layer* and *Physical Layer* are all that is needed. The CAN standard defines the *hardware* (i.e., physical layer) and the *communication* on a basic level (i.e., data link layer).

In the *Physical Layer* an interface for transmission of information is provided, where all aspects of physical communication are covered (*i.e.*, mechanical, electrical, functional and procedural interface).

The *Data Link Layer* provides reliable communication over the physical layer interface (*i.e.*, error handling, physical addressing, network topology, line discipline, and flow control).

The *Application layer*, known as service layer is the highest level of OSI model. It contains management functions to support distributed applications (*e.g.*, authentication of message, file transfer, remote control).

2.2.2 Electrical Topology

The robust design allows operating in extremely harsh environments with the extensive error checking mechanisms (*i.e.*, due to transmission errors). The most common and cheapest physical layer implementation of CAN bus communication is with a twisted wire pair, driven with balanced differential signal (see Figure 2-3), where bus lines are called ‘*CAN high*’ and ‘*CAN low*’. Differential voltage between *CAN high* and *CAN low* is around 0 V for a *recessive bit* (logic 1) and about 2 V for a *dominant bit* (logic 0).

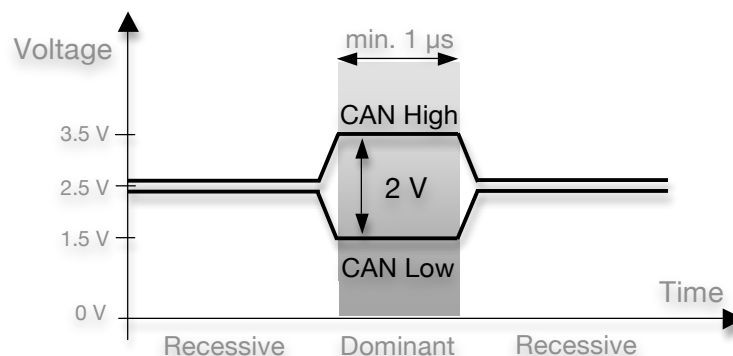


Figure 2-3: CAN dominant and recessive bus states, modified from [9]

CAN is insensitive to electromagnetic interference (EMI) or inductive spikes because both bus lines (*CAN high* and *CAN low*) are affected in the same way (with a constant differential voltage). The each end of the CAN bus line is terminated with a $120 \Omega^V$ resistor to suppress reflections [13].

^V - The basic unit of electrical resistance.

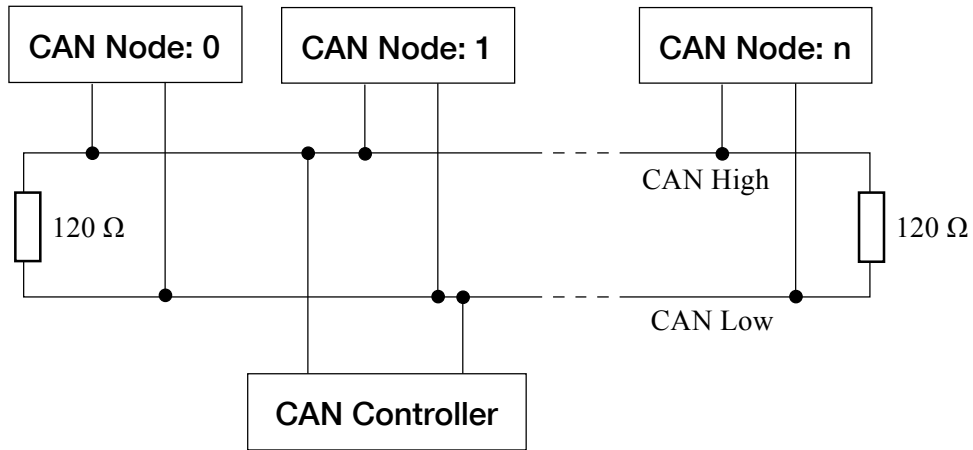


Figure 2-4: CAN bus standard termination

To further reduce emissions, standard bus termination can be replaced with split or biased split termination (see Figure 2-5). In split termination unwanted high frequency noise is blocked by the added coupling capacitor between two equal ($60\ \Omega$) termination resistors that ground potential. The best performance is achieved with biased split termination, where additional voltage divider circuit is used to stabilize the recessive common mode voltage (to set the voltage between resistors to $V_{cc}/2$).

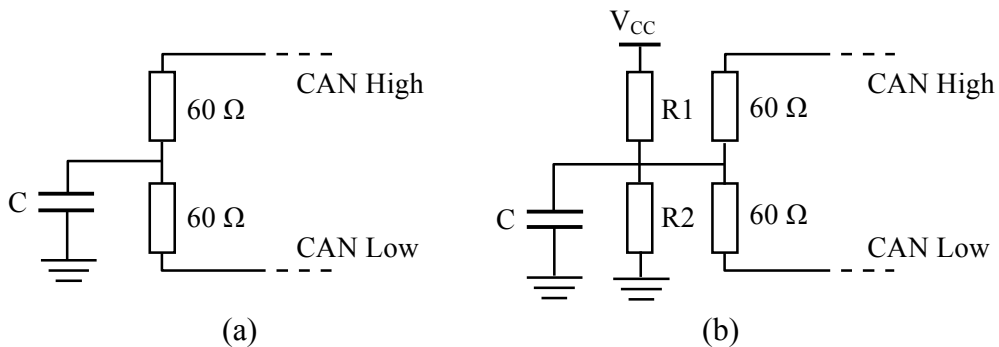


Figure 2-5: Split (a) and biased split (b) bus termination

To reduce the electromagnetic emission on the bus itself when operating at high baud rates, twisted wire pairs can be additionally shielded.

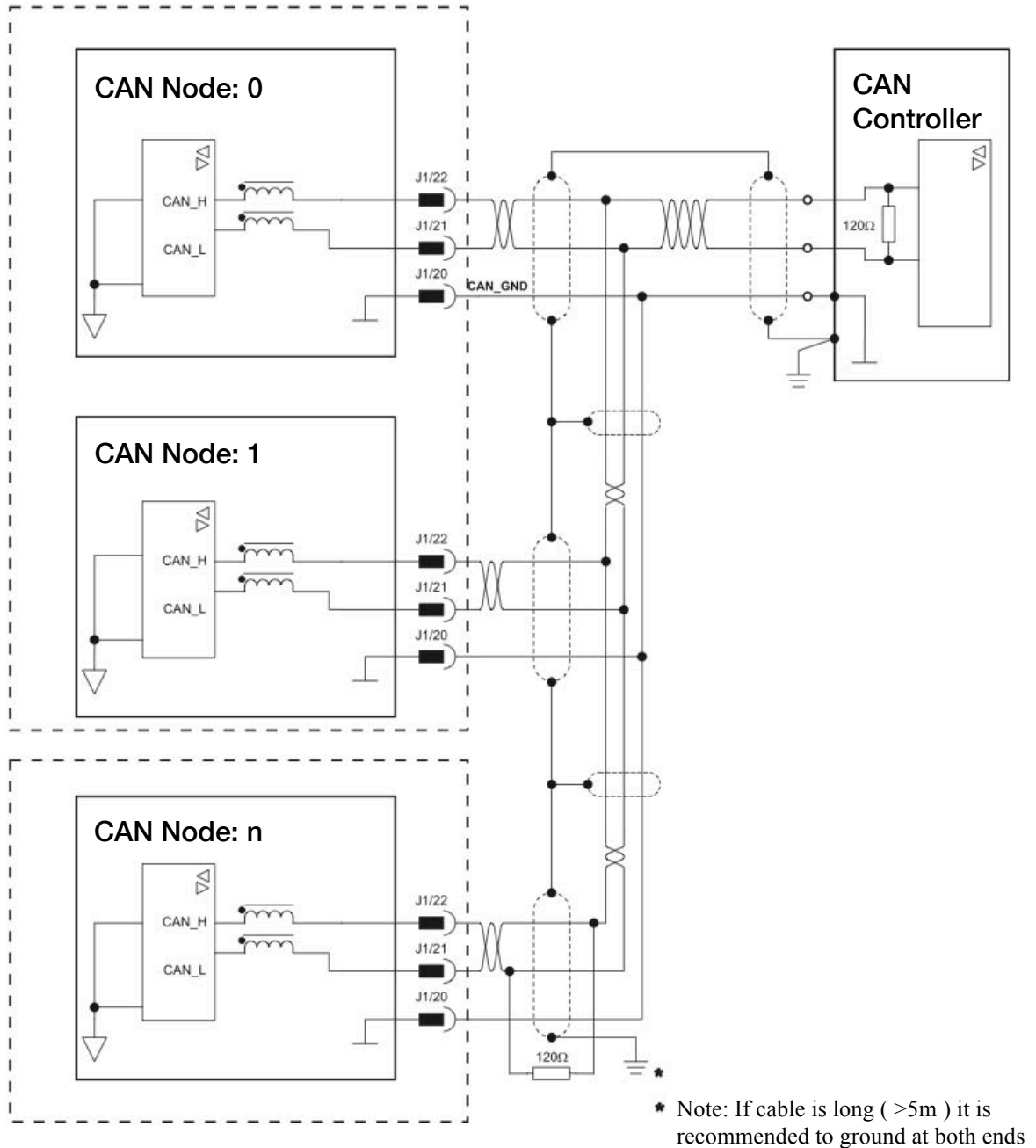


Figure 2-6: CAN bus electrical topology prone to EMI [14]

All messages are sent and delivered to all nodes on the bus. Maximal number of nodes is up to sixty-four, where specific nodes have to be addressed during the configuration of a system for each module (*i.e.*, at least one CAN priority/identifier has to be reserved).

Data that are transmitted and received in a CAN system (data from any transmitting node to single or multiple receiving nodes) are carried in the message frame [9], [10].

2.2.3 CAN Message Frame Format

The CAN uses a specific message frame format for data transmission. CAN protocol supports two message frame formats:

- ▶ Standard CAN (using 11-bit identifiers) - *Low-Speed CAN & Version 2.0A*
- ▶ Extended CAN frame format (using 29-bit identifiers) - *Version 2.0B*

Extended frame format is backwards compatible with standard format and can be used on the same bus. For this reason two bit identifiers are in the Arbitration Field of an extended frame format (see Figure 2-7). The first is 11-bits long (*i.e.*, base identifier) and the second is 18-bits long (*i.e.*, identifier extension). This gives a total length of *29-bits*. In case when those standard and extended frame formats are used on the same network, messages in standard format always have priority over an extended format message (*i.e.*, when both of them have same base 11-bit identifier).

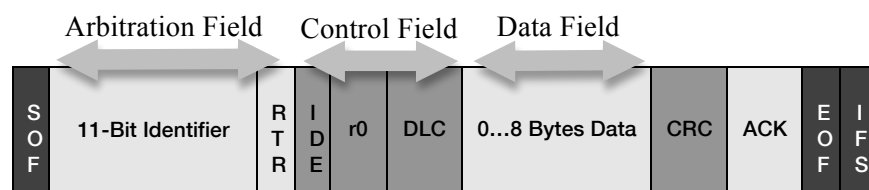


Figure 2-7: Standard CAN frame format

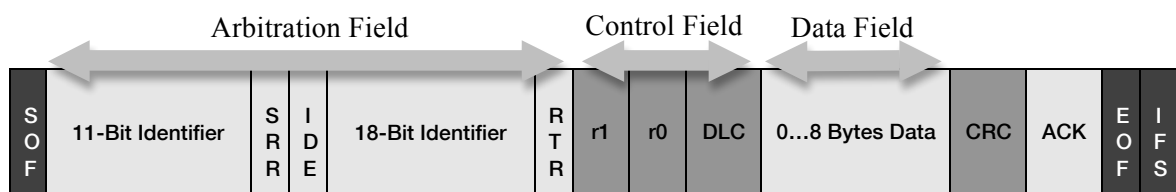


Figure 2-8: Extended CAN frame format

Each field in message frame defines a specific data within each message sent. Namely,

SOF (Start Of Frame) – specifies the start of frame transmission.

Arbitration Field – includes the message Identifier (11-bit identifier for standard and 29-bit identifier for extended frame format) and the *RTR bit* (Remote Transfer Request, to differentiate from Remote frames). *SRR bit* (Substitute Remote Request) replaces the RTR bit in the standard message location as a placeholder in the extended format.

Control Field – contains the 4-bit DLC (Data Length Code), which indicates the number of bytes of data being transmitted in the following Data Field. The additional bits r0 and r1 are reserved for future extensions of CAN.

Data Field – holds up to 8 data bytes.

CRC (Cyclic Redundancy Check) – the 15-bit CRC detects whether any bits in the transmission have changed (error detection).

ACK (Acknowledgement) – the 2-bit field indicates if an error-free message has been sent. Message is retransmitted if ACK is not detected.

EOF (End Of Frame) – the 7-bit field marks the end of a CAN frame (message)

IFS (Intermission Frame Space) – this 3-bit frame contains the information about amount of time required by the controller to move a correctly received frame to its proper position in a message buffer area.

IDE (Identifier Extension) – distinguishes between frame types (standard or extended frame format).

Because CAN alone is not suitable for any complex automation process (*i.e.*, limitation of 8-bytes per message), CAN network uses a *Higher Layer Protocol* (HLP).

2.2.4 CAN-based Higher Layer Protocol

The HLP manages the communication inside the CAN bus system and is used to implement the upper five layers of the OSI model in CAN.

Inside the HLP is typically specified start-up behavior, as well as message identifiers distribution among the different nodes in a system, the translated data frame content, and the system status report [9]. HLP allows a master/slave configuration (CAN is a multi-master system and itself does not support a master/slave configuration) with unlimited length of messages, since CAN is limited to only 8-bytes per message.

There are several higher layer protocols available for the CAN bus, the most common ones are listed below [9], [10].

DeviceNet is a low-level industrial application layer for industrial automation systems that requires robust and efficient data handling. Larger data packages require segmentation. DeviceNet network supports message prioritization and can communicate with up to 63 nodes on the network and a bus speed of 125 / 250 / 500 kbit/s.

CANopen is used in cases, which require highly flexible and configurable embedded networks. Originally designed for motion-oriented machines control networks (e.g., assembling and handling machines), and used in various applications, such as industrial machines, medical devices, vehicles, building automation. Supports up to 127 nodes on the network with 10 kbit/s and up to 1 Mbit/s of data throughput.

CAN Kingdom is designed to achieve maximum machine and process performance from the controller. Unlike other CAN based higher layer protocols, CAN Kingdom is not an application layer that tries to follow OSI model.

SDS stands for Smart Distributed System, like DeviceNet and CANopen, is based on the CAN protocol. One of the main uses is for machine control application where it serves as an advanced bus system for intelligent sensors and actuators.

J1939 is a serial data bus communication standard for communication, and diagnostics among vehicle components and microprocessor systems (also called Electronic Control Units – ECU). The J1939 uses higher-layer protocol built on CAN. It supports up to 30 nodes and a bus speed of 250 kbit/s.

CCP/xCP is a protocol for calibration and measurement data acquisition. The CAN Calibration Protocol (CCP) or The Universal Measurement and Calibration Protocol (xCP) is an *application layer* for CAN 2.0B. Besides supporting the CAN bus it also supports other communication interfaces (e.g., FlexCAN, EtherNet/IP).

Due to technical limitations [14] from a hardware implemented in this project, CANopen should be used as a communicational protocol between central controller and joint controllers. In the upcoming chapter, electrical motors that are used inside the robot are presented.

3 MOTOR ACTUATORS

In Archie each joint is actuated by electrical motor. Some electrical motors as actuators are not as efficient at top speed and might not be safe to operate at higher speeds. Advantage of using DC motors as actuators in joints is that they can provide a high starting torque, with smooth performance and quiet operation. In the following subsection a mathematical model of the DC motor is described.

3.1 Mathematical Model of the DC Motor

Mathematical model gives us a better understanding how position and velocity of a DC motor are controlled, considering the electrical and mechanical characteristics of the system.

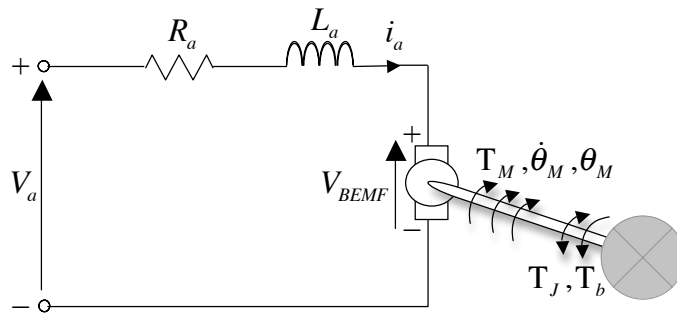


Figure 3-1: Schematic model of a DC motor

Kirchhoff's 2nd voltage law states that sum of all voltages within the loop must equal zero,

$$V_a - V_{Ra} - V_{La} - V_{BEMF} = 0 \quad (3-1)$$

where V_a is the applied voltage and V_{Ra} defines the rotor electrical resistance, given by the resistance of the motor armature (R_a), and inductor current in the motor armature (i_a).

$$V_{Ra} = R_a \cdot i_a \quad (3-2)$$

The motor armature voltage (V_{La}) is proportional to change in inductance of the coil (L_a) with respect to change in position of rotor, and the armature inductor current (i_a).

$$V_{La} = L_a \frac{d}{dt} i_a \quad (3-3)$$

The effective voltage of the motor is reduced by back electro-motive force (BEMF) voltage (V_{BEMF}) that is calculated based on the motor angular velocity ($\dot{\theta}_M$) and the BEMF constant (k_b). The BEMF voltage increases with increased angular velocity.

$$V_{BEMF} = \dot{\theta}_M \cdot k_b = \frac{d}{dt} \theta_M \cdot k_b \quad (3-4)$$

From Newton's 3rd law of motion, sum off all torques acting on a body is equal to zero.

$$T_M - T_J - T_b - T_L = 0 \quad (3-5)$$

where T_M is the motor torque, T_J is the torque from the angular acceleration of the rotor, T_b is the friction torque, and T_L is the load torque. T_L is not considered at this point and will be introduced later. The motor torque is calculated from the resulted current through the armature winding (i_a), where k_t is the torque constant. Therefore,

$$T_M = i_a \cdot k_t \quad (3-6)$$

The torque from the angular acceleration is given by:

$$T_J = J \frac{d}{dt} \dot{\theta}_M = J \frac{d^2}{dt^2} \theta_M \quad (3-7)$$

where J is the shaft^{VI} inertia, and θ_M is the motor angular position. Friction torque (T_b) is presented in Equation (3-8) below, where b is the damping factor for the shaft friction losses:

$$T_b = b \frac{d}{dt} \theta_M \quad (3-8)$$

To formulate the state space model, Equations (3-1) and (3-5) are rearranged for solving derivatives, written as:

$$\frac{d}{dt} i_a = -\frac{R_a}{L_a} \cdot i_a - \frac{k_b}{L_a} \cdot \dot{\theta}_M + \frac{1}{L_a} \cdot V_a \quad (3-9)$$

$$\frac{d}{dt} \dot{\theta}_M = \frac{k_t}{J} \cdot i_a - \frac{b}{J} \cdot \dot{\theta}_M \quad (3-10)$$

$$\frac{d}{dt} \theta_M = \dot{\theta}_M \quad (3-11)$$

^{VI} - Shaft is a mechanical component for transmitting torque and rotation on the output of the motor.

Differential equations given in previous page are written into state space form, expressed as:

$$\dot{X} = \begin{bmatrix} \frac{d}{dt} i_a \\ \frac{d}{dt} \dot{\theta}_M \\ \frac{d}{dt} \theta_M \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_b}{L_a} & 0 \\ \frac{k_t}{J} & -\frac{b}{J} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ \dot{\theta}_M \\ \theta_M \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} V_a = Ax + Bu \quad (3-12)$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \dot{\theta}_M \\ \theta_M \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} V_a = Cx + Du \quad (3-13)$$

Using the Laplace transformation, Equations (3-9) and (3-10) are modified in the following form:

$$I_a(s) = \frac{V_a(s) - s \cdot k_b \cdot \theta_M(s)}{(L_a \cdot s + R_a)} \quad (3-14)$$

$$\theta_M(s) = \frac{k_t \cdot I_a(s)}{(J \cdot s^2 + b \cdot s)} \quad (3-15)$$

where $I_a(s)$ from Equation (3-14) is substituted into Equation (3-15) to obtain:

$$(J \cdot s^2 + b \cdot s) \cdot \theta_M(s) = k_t \cdot \frac{V_a(s) - s \cdot k_b \cdot \theta_M(s)}{(L_a \cdot s + R_a)} \quad (3-16)$$

which is used for the transfer function, $G_P(s)$, between the input voltage $V_a(s)$ and the motor angular position $\theta_M(s)$:

$$G_P(s) = \frac{\theta_M(s)}{V_a(s)} = \frac{k_t}{s \cdot ((L_a \cdot s + R_a) \cdot (J \cdot s + b) + k_b \cdot k_t)} \quad (3-17)$$

Therefore, the transfer function, $G_V(s)$, between the input voltage $V_a(s)$ and the motor angular velocity $\dot{\theta}_M(s)$:

$$G_V(s) = \frac{\dot{\theta}_M(s)}{V_a(s)} = \frac{k_t}{((L_a \cdot s + R_a) \cdot (J \cdot s + b) + k_b \cdot k_t)} \quad (3-18)$$

The Equation (3-17) is shown in the block diagram (Figure 3-2).

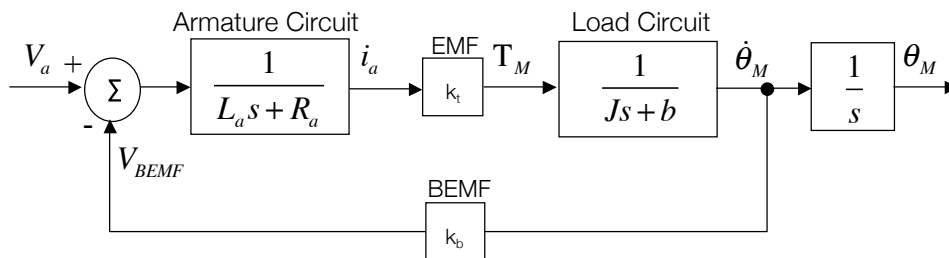


Figure 3-2: Block diagram of the DC motor, formed from given equations above

3.2 Motor Based Joints

Inside of the robot there are three different types of motors used: brushed, brushless DC motors and the rest (but not seen in the picture) in the upper (unassembled) part of the body that are standard RC servomotors. To save cost and due to the mechanical space restrictions in lower body [7], the brushed DC motors are used in hip (*i.e.*, transversal movement), and heel-and-toe.

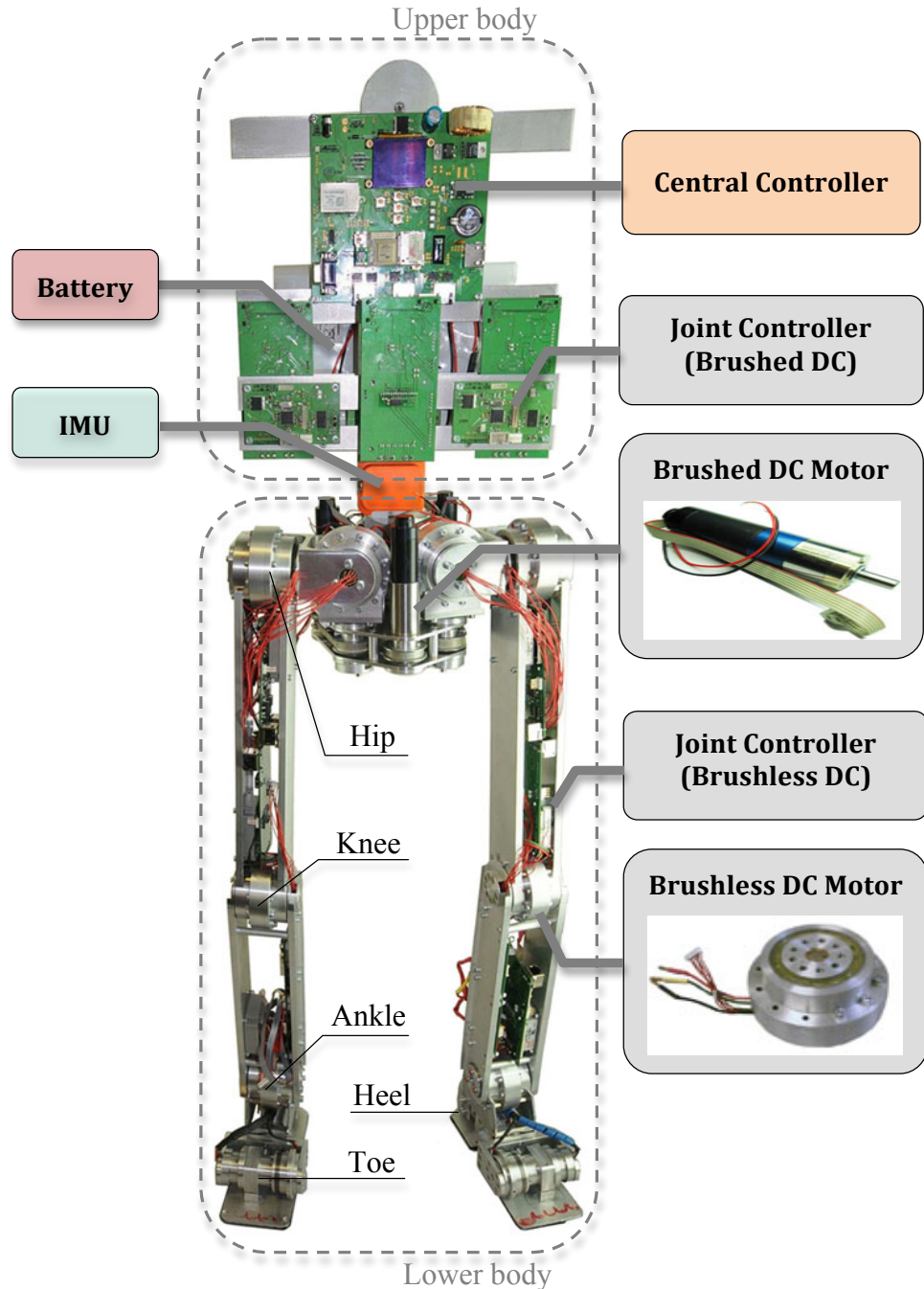


Figure 3-3: Frontal view from Archie structure

In general Archie's joint consists of three parts, a motor, a gear drive, and an encoder. Those motor joints are fixed in a suitable place on pre-selected areas across the skeletal structure. The speed of rotation of DC motor is usually too fast to be directly applied in a robot and by lowering the speed they may become less reliable. In this case, it is necessary to use gear reduction, to slow down the speed of the motor shaft. Decreased output velocity has the positive side effect of increasing torque. Higher output torque out of the motor shaft in Archie's joint is achieved with, a harmonic and planetary gear drive.

Two types^{VII} of motors used for joints in a lower body of a robot are presented, a brushed and a brushless DC motor combined with a gear drive.

BRUSHED DC MOTOR

The hip joint from the robot consists of a 24 V **brushed DC motor** (i.e., Faulhaber, 3257 series), which is combined with a planetary gear drive to achieve higher output torque (up to 10 Nm of torque at 12 rpm). An incremental shaft encoder that has resolution of 2048 lines per revolution is placed onto a rotating part of a motor. Generated count pulses from an encoder are used for positioning, and for the indication and control of both shaft velocity, acceleration, and direction of rotation.

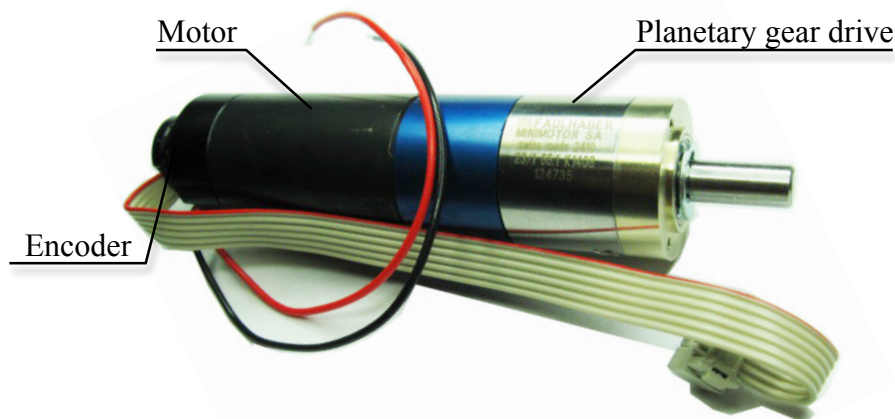


Figure 3-4: The brushed DC motor combined with planetary gear drive and encoder

^{VII} - The side-by-side comparison can be found in Appendix A-2.

Brushed motor is connected to the **planetary gear drive** (Faulhaber, 38/1) with a gear ratio 1:415. The planetary gear drive mechanism contains four main parts [15]:

- ▶ *Sun gear* fixed on the motor shaft, which rotate the planet gear.
- ▶ *Planet gear* rotated from the sun gear. Planet gear rotates the planet carrier in the same direction as the sun gear but with reduced speed (due to gear reduction).
- ▶ *Planet carrier* with attached planet gear pins on a plate connected to the output shaft, which is driven by the planet gear.
- ▶ *Internal gear* fixed to the gear drive chassis with the inner tooth ring gear to accommodate the planet gear shaft spline.

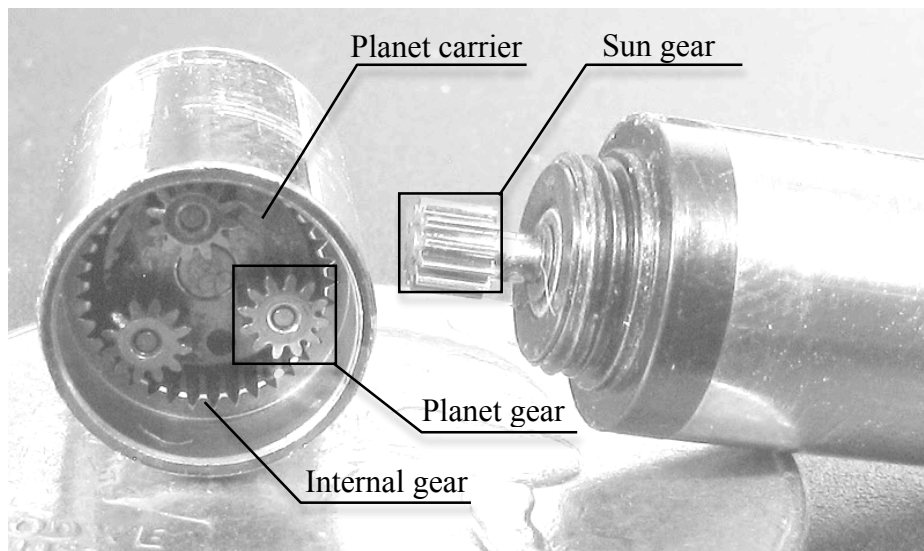


Figure 3-5: Planetary gear drive [16]

BRUSHLESS DC MOTOR

The 3-phase low voltage 24V **brushless DC motors** made by Maxon-Motors (type EC45) are used for joint movement inside a humanoid robot (see Figure 3-3). This is a kind of DC motor that has no brushes, where the commutation is controlled electronically. The major advantage is that this type of a motor does not require any maintenance, as it is required for brushed motors (i.e., brush dust). Overall, this motor offers high dynamic performance (i.e., smaller moment of inertia), higher ratio of torque to weight, and a higher efficiency [10]. It produces smaller amount of electrical noise and electromagnetic interference when compared to brushed motors.

The **harmonic gear drive** (*i.e.*, Harmonic Drives Systems Inc., 20-160-874405-6) is connected to the brushless motors with a gear ratio 1:160. Usage of harmonic gear drive decreases the size of the robot joints and increases its efficiency (e.g., low vibration and high positional accuracy). By using harmonic drive, the motor have a compact design with almost zero backlash^{VIII}. In combination with brushless DC motor the output shaft of gear drive can produce up to 13 Nm^{IX} of torque at 42.5 rpm^X.

Harmonic gear drive [15] consists of three components:

- ▶ *Circular spline*^{XI} fixed on a motor case with a solid circular shape steel ring and internal teeth inside the diameter.
- ▶ *Flexible spline* placed on the harmonic gear drives mechanism with a flexible steel cylinder that reveals an outer element with external teeth.
- ▶ *Wave generator* fitted onto an elliptical plug that represents a solid disk with a small ball bearing.

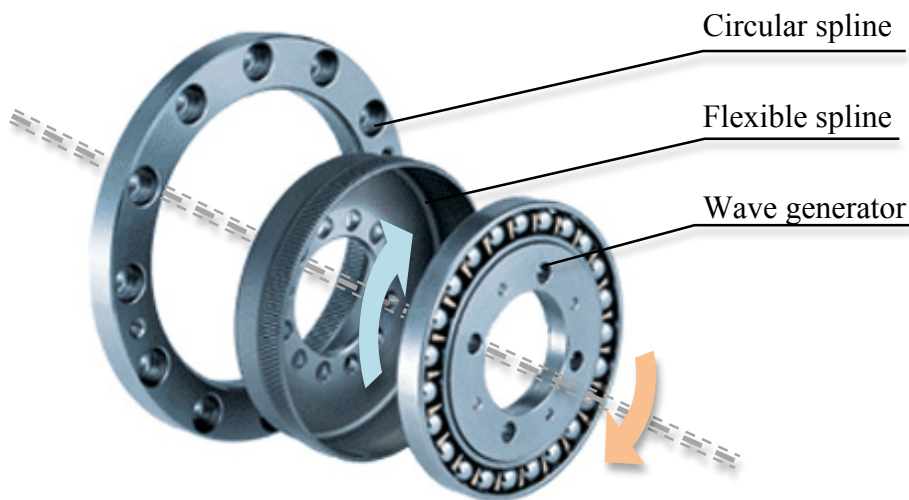


Figure 3-6: Exploded view of a harmonic gear drive [17]

^{VIII} - Backlash is the relative motion of mechanical parts caused by looseness.

^{IX} - Stands for Newton metre, a unit of torque.

^X - Abbreviation for revolutions per minute.

^{XI} - Spline or 'gear teeth', is used for mechanical torque transmission along the axis of the shaft.

The Figure 3-7 below shows casing of the motor with an integrated encoder sensor and a harmonic gear drive used for joint actuation.

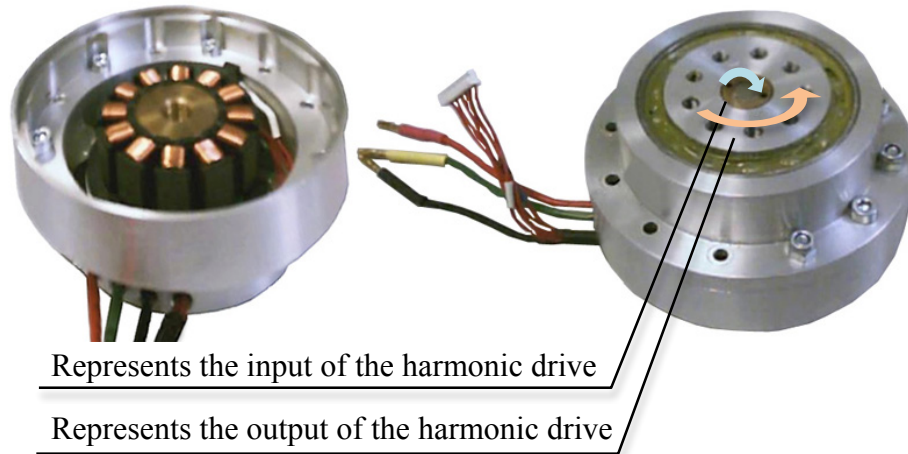


Figure 3-7: The brushless motor combined with a harmonic gear drive.

The motor has integrated 360 lines per revolution Hall effect magnetic encoder AS5134 made by Austrian Micro Systems for measuring position and velocity detection as feedback information for the control algorithm. Encoder senses pulses from a small magnet attached on a rotor and send them back into the motor controller.

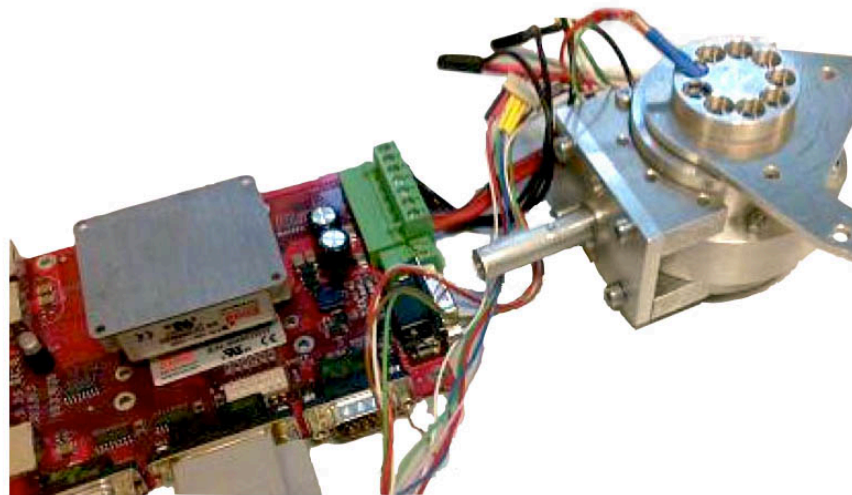


Figure 3-8: Motor controller connected to the joint [7]

For the purpose of controlling the above mentioned motors, two different circuit boards have been developed. Each board is fitted to joint actuators, according to specifications given in technical data sheets.

4 IMPLEMENTATION

Computer based ECAD^{XII} programs have significantly contributed to rapid design success in fast-growing market of electronic applications since pre-computer days. As part of the project, circuit board has been created with ECAD-software. Today's ECAD systems available on the market help us to design complex PCB layouts with ease. Over the years, technology has progressed so much that electronic circuits can be tested before leaving the virtual environment (depends on the program used). Those virtual prototyping simulations can be further embedded directly into the 3D modeling MCAD^{XIII} systems. Whether designing single-sided, double-sided or multilayer (*i.e.*, trace layer inside the PCB) boards, the understanding of the circuit operation is critical in the design process.

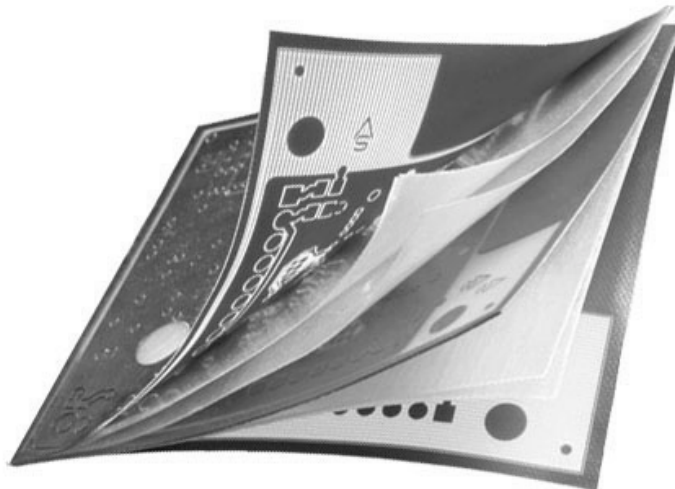


Figure 4-1: Multilayer PCB [18]

A PCB consists on one or more layers of insulating material made from woven glass fiber material impregnated with epoxy (FR-4, Flame Retardant laminate). Use of heat sinks can be reduced/eliminated when using metal backed PCBs with materials like aluminum and copper, which convey heat more effectively.

^{XII} - Acronym of Electronic Computer Aided Design, a type of software for designing electronic systems.

^{XIII} - Acronym of Mechanical Computer Aided Design, computer system used to design products.

4.1 PCB Design Process Overview

Decisions which components, material types and manufacturing methods will be used in PCB fabrication process, should be clear from the whole beginning of designing a PCB prototype. Knowing the requirements definition is the first step to building a prototype. No matter how complicated or simple is the PCB; design flow process normally follows the sequence described below.

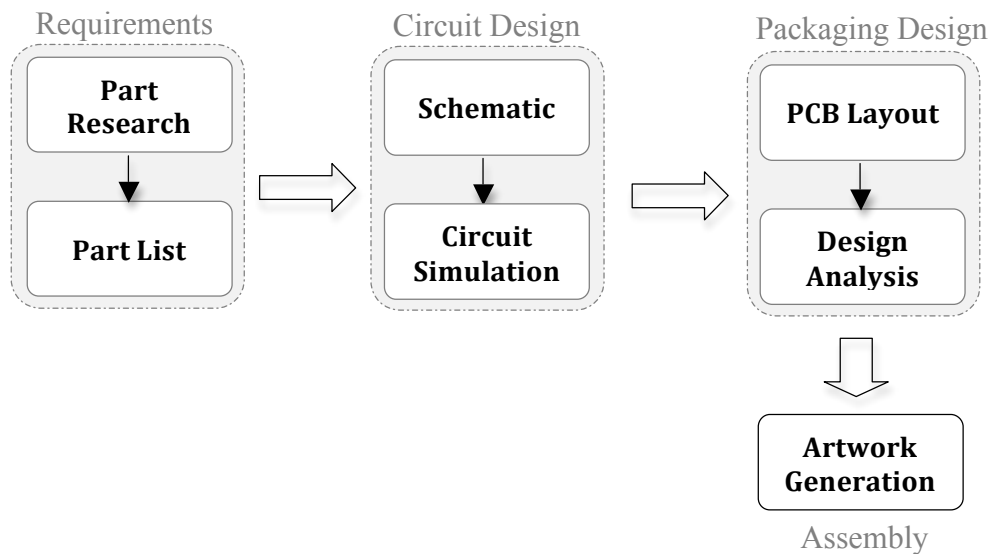


Figure 4-2: Simplified PCB design flow

In the early stages of the PCB design individual parts are matched to meet the design specifications. The circuit is wired (drawn) together into a visual diagram (schematic) from pre-selected compatible parts. The schematic represents relevant information's of an electronic circuit that are easily understandable and readable.

After adequate schematic has been drawn, circuit can be simulated^{XIV} in a virtual environment to evaluate the behavior of real-world parts. Thereby, analysis helps to eliminate signal integrity issues, eliminate crosstalk and detect electromagnetic compatibility (EMC) related problems at the earliest stage of prototyping. The schematic is then transformed into board layout where each part is placed and routed to maintain signal, power, and thermal integrity.

^{XIV} - Depends on which ECAD-software is used.

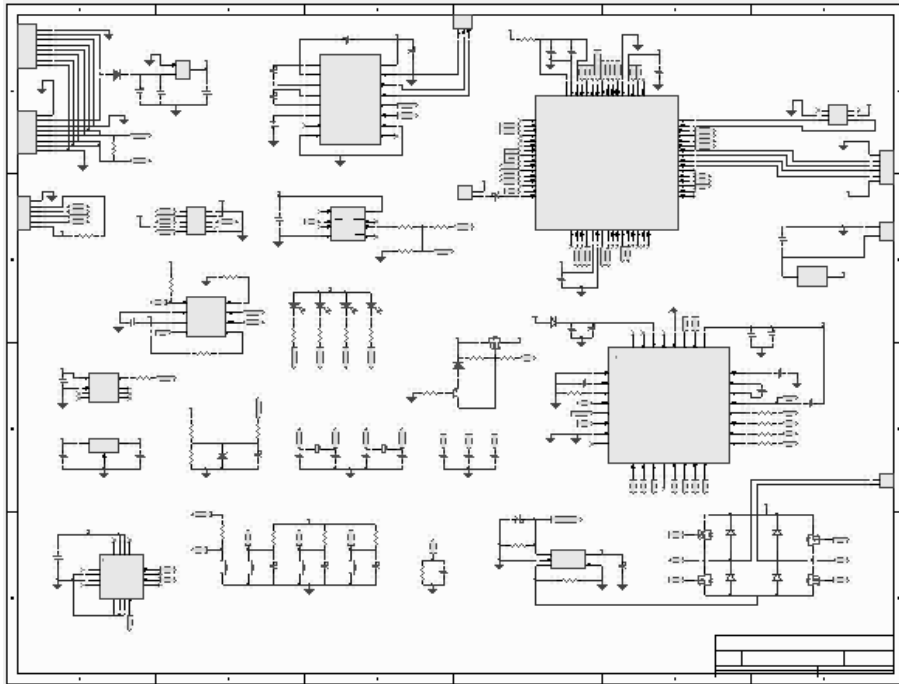


Figure 4-3: The schematic document

Before beginning actual assembly, using design software analysis tools, any potential electrical and design violations can be discovered and corrected if needed. If there are no further errors, artwork generation is ready for the first circuit prototype board fabrication.

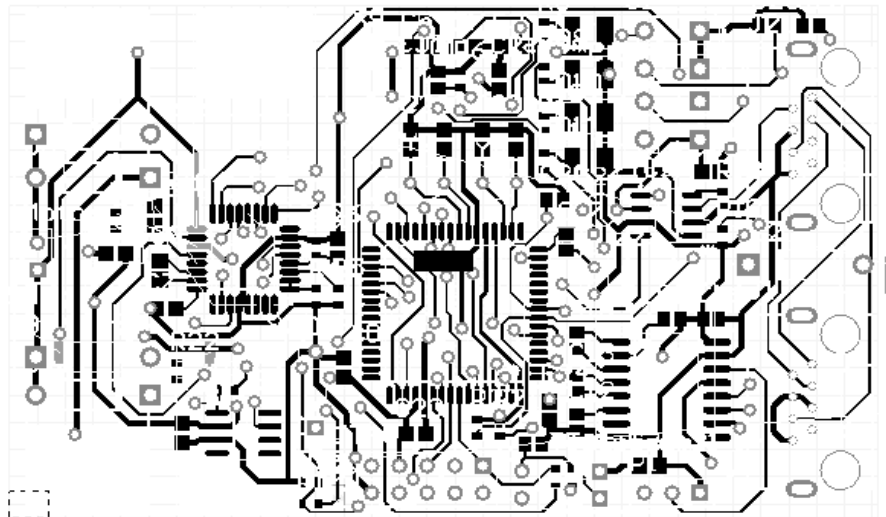


Figure 4-4: PCB layout during optimization

On the following pages, the process described above is presented with short description of components used for running robot joints.

4.2 Motor Controller's Requirements

Each driver circuit needs to support position and velocity control. Since joints of the robot are dealing with different load properties regarding to a movement direction, the motor driver should be able to detect position (encoder) and provide appropriate data's to the central controller via CAN connection. New motor driver should physically fit on the robot's chassis and should support most of the existing hardware.

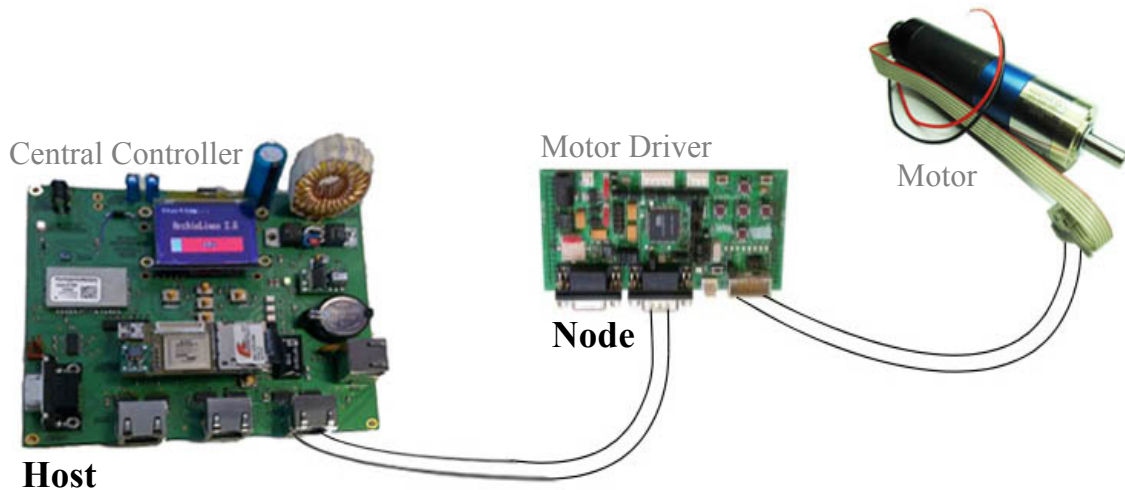


Figure 4-5: Block diagram of CAN based motor driver

Therefore, some parts of the custom designed motor drivers are described in the following subsections with a description of components. Both motor drivers have been designed for small size and maximum stability to fit inside the robot skeleton chassis.



Figure 4-6: Aluminum skeleton chassis used to convey the heat away from the controller

4.3 Choosing the Components

In the sake of finding the best choice for the component's brand using the distributor's selling catalog might ease the process of selection. With a small research and compatibility check of manufactures component characteristics, a wide range of hardware problems can be prevented and problems identified.

Some of the distributor companies (such as Farnell) are presenting online catalogues to enable the costumer to compare the components. Using the electronic components distributor website, a giant database of a large variety of available components on the market are technically described and specified. Moreover, information about the price is available, beside the technical specifications. An appropriate switches (*i.e.*, transistors), and strong enough elements (*e.g.*, resistors, capacitors, diodes) have been determined by the maximum current flows, and a voltage carried by each conductor in a circuit.

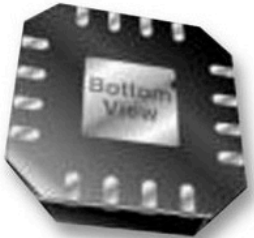


Image is for illustrative purposes only. Please refer to product description

Manufacturer: ANALOG DEVICES
Order Code: 1274125
Manufacturer Part No: ADXL330KCPZ
RoHS: ● Yes

Description

- ACCELEROMETER LOW POWER, ADXL330
- Acceleration Range:± 3.6g
- No. of Axes:3
- Sensor Case Style:LFCSP
- No. of Pins:16
- Supply Voltage Range:1.8V to 3.6V
- Operating Temperature Range:-25°C to +70°C
- SVHC:No SVHC (18-Jun-2010)
- Operating Temperature Max:70°C
- Package / Case:LFCSP
- Temperature Operating Min:-25°C
- Base Number:330
- IC Generic Number:ADXL330
- Logic Function Number:ADXL330
- Packaging Type:Peel Pack
- Sensitivity:300mV/g
- Supply Current:320µA
- Supply Voltage Max:3.6V
- Supply Voltage Min:1.8V
- Termination Type:SMD
- No. of Bits:330

[▶ Show Alternatives](#)
[▶ Show Accessories](#)

Availability	
Availability:	861
Price For:	1 Each
Minimum Order Quantity:	1
Order Multiple:	1
Unit Price:	£14.08
Qty	<input type="text" value="1"/> BUY

Price	
Qty	List Price
1 - 9	£14.08
10 - 99	£10.80
100 - 249	£9.16
250 - 499	£8.78
500 - 999	£8.22
1000+	£7.73

Figure 4-7: Component search result from online catalog [19]

By knowing which communication protocol, and what type of motors are intended for use in the joints, the next task was to design a suitable motor controller. Motor drivers have been developed to supply motors with power. With feedback received to a controller, it can control position and velocity of the motor with a good deal of precision.

4.4 Brushless DC Motor Driver

The control solution of the brushless DC motor driver is based upon a commercial solution using *Elmo Whistle* (Elmo Motion Control, 2006) to run brushless DC motors.

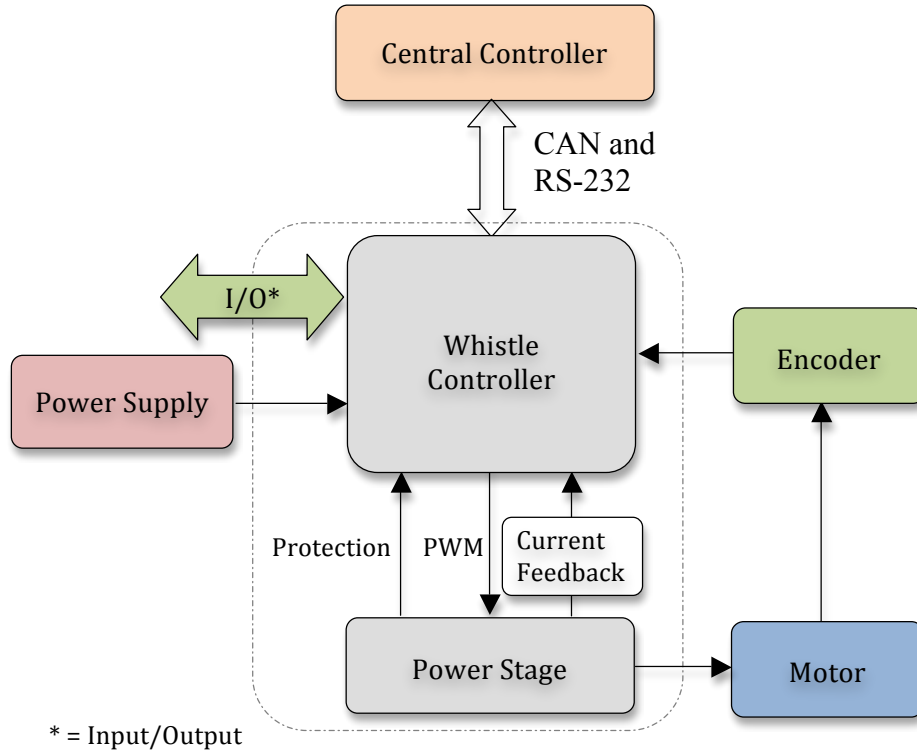


Figure 4-8: Brushless DC motor driver block diagram, based on [14]

These drives are ready for PCB mounting and designed to be embedded into existing applications with minor modifications. Each brushless motor driver is combined with Whistle motor drive controller that is mounted into PCB.

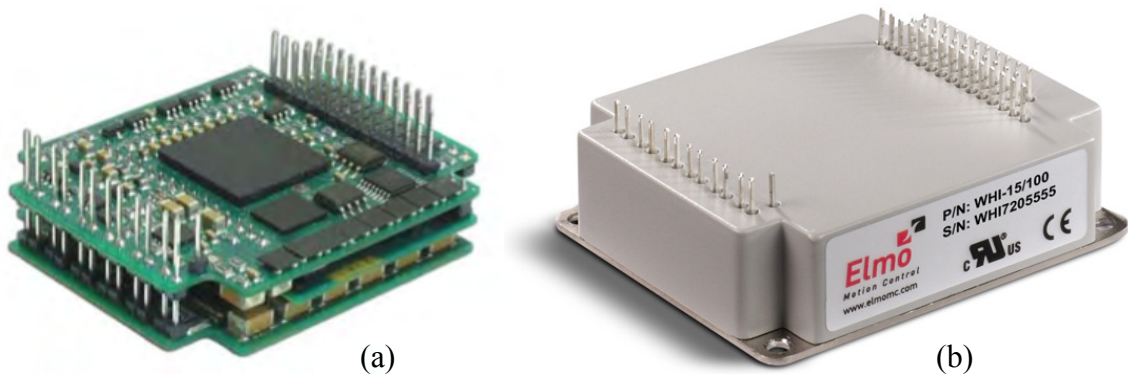


Figure 4-9: Elmo Whistle drive (a), and with heat sink mounted (b), taken from [14]

Elmo's Whistle [14] drives are used in a variety of industrial machinery equipment (e.g., robotics, semiconductor production, lab automation, avionics). As stated in datasheet of Whistle: 'They operate from a DC power source in current, velocity, position and advanced position modes, in conjunction with a permanent-magnet synchronous brushless motor, DC brush motor, linear motor or voice coil.'

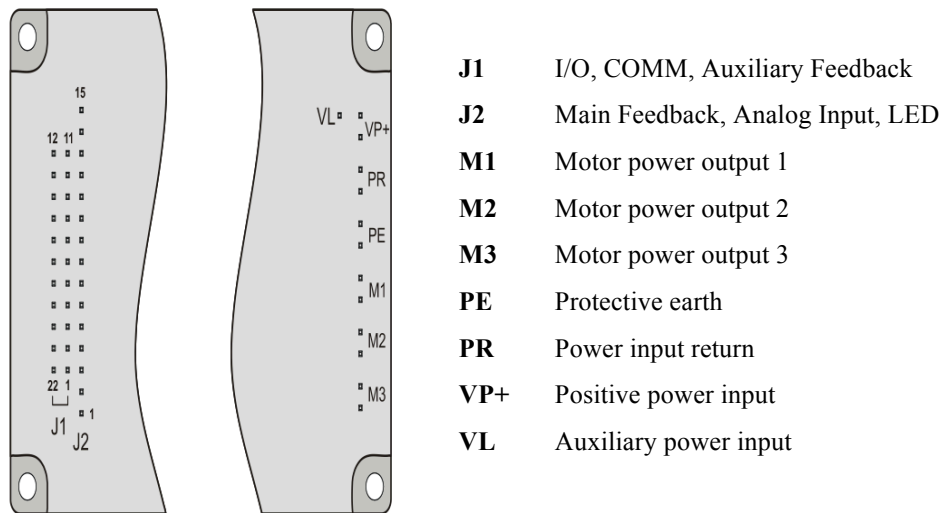


Figure 4-10: Pins configuration of Elmo Whistle [14]

As Elmo motion controllers have already integrated support for CAN and RS-232 serial communication, just small tweaks and changes were required on the circuit board.

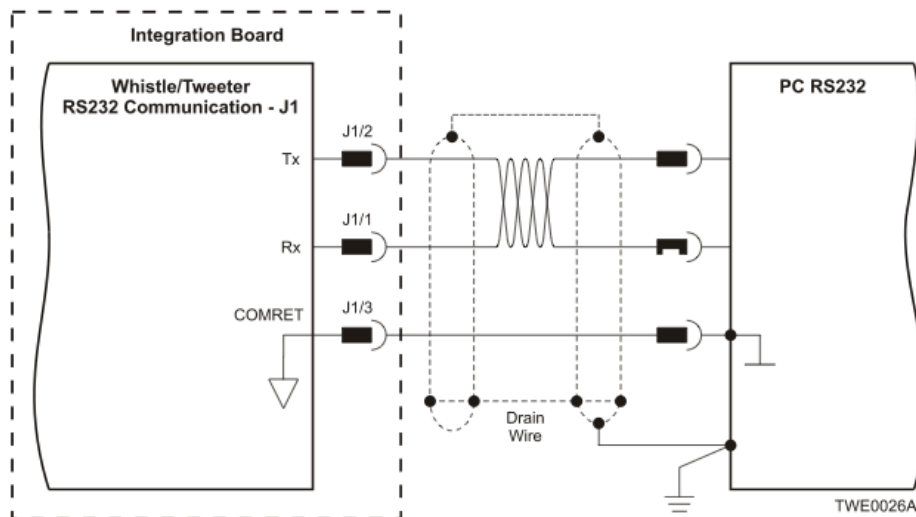


Figure 4-11: RS-232 connection diagram of Elmo Whistle [14]

4.5 Brushed DC Motor Driver

For a brushed DC motor driver, the Elmo Whistle has been replaced with the microcontroller and H-bridge. The new design^{XV} of the brushed DC motor driver is based upon experiences with Whistle during laboratory testing (explained in Chapter 5).

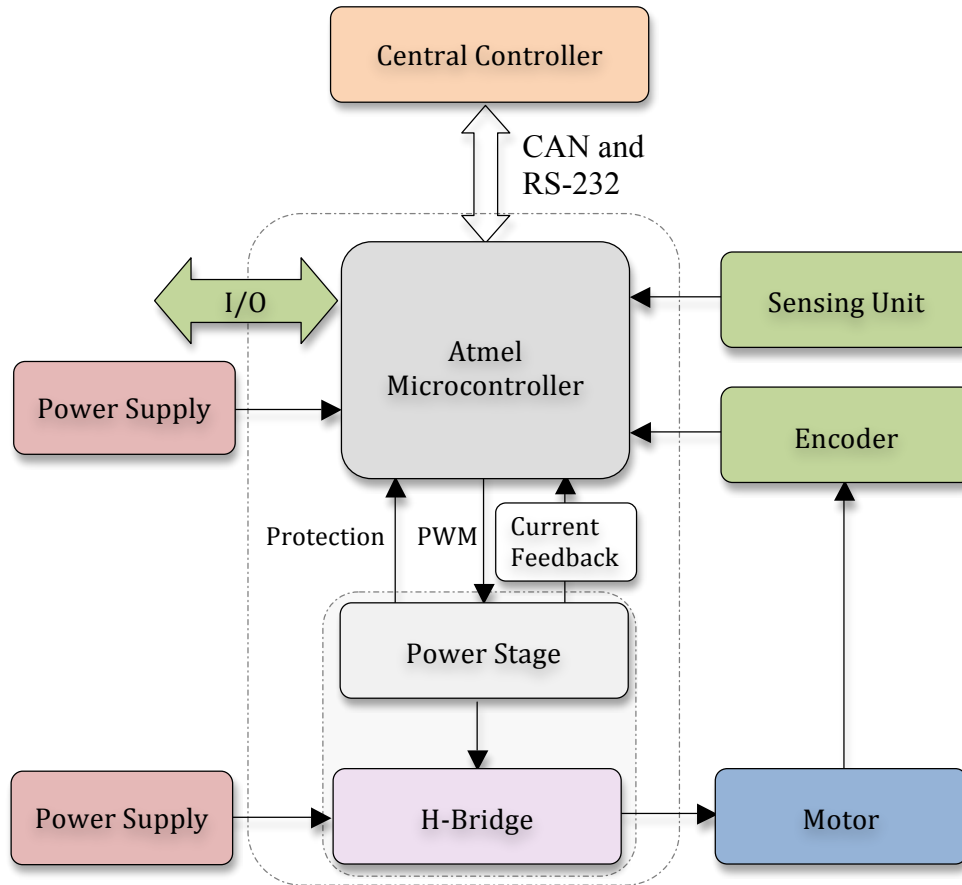


Figure 4-12: Brushed DC motor driver block diagram, based on [14]

The selected microcontroller family AT90CAN128 from the Atmel is used to implement controlling algorithm. The AT90CAN128's main function is to create Pulse Width Modulation (PWM) signals to control the motors. Since the motor cannot be driven directly from the microcontroller's output, there must be some interface circuitry so that the motor power is supplied from another power source. The power stage is based on the ATA6824 H-bridge driver from Atmel.

^{XV} - The schematic, layout and bill of materials are included in Appendix B-2, C-2, and D-2.

Because the motion control functionality is already implemented on-chip, only the direction command and the PWM signal for the velocity information needs to be provided by the microcontroller [20]. The circuit used in this project is called H-bridge. It consists of four switches IRLML0030 from International Rectifier that are controlled by a microcontroller and driven by H-bridge DC motor driver.

The sensing consists of the axis electronic accelerometer ADXL330 made by Analog Devices, temperature sensor LM35DM from National Semiconductor, and motor current sensor INA139 made by Texas Instruments that are connected into Analog-to-Digital Converter (ADC). Joint position is obtained from implemented encoder on the motor shaft, which is connected to a microcontroller.

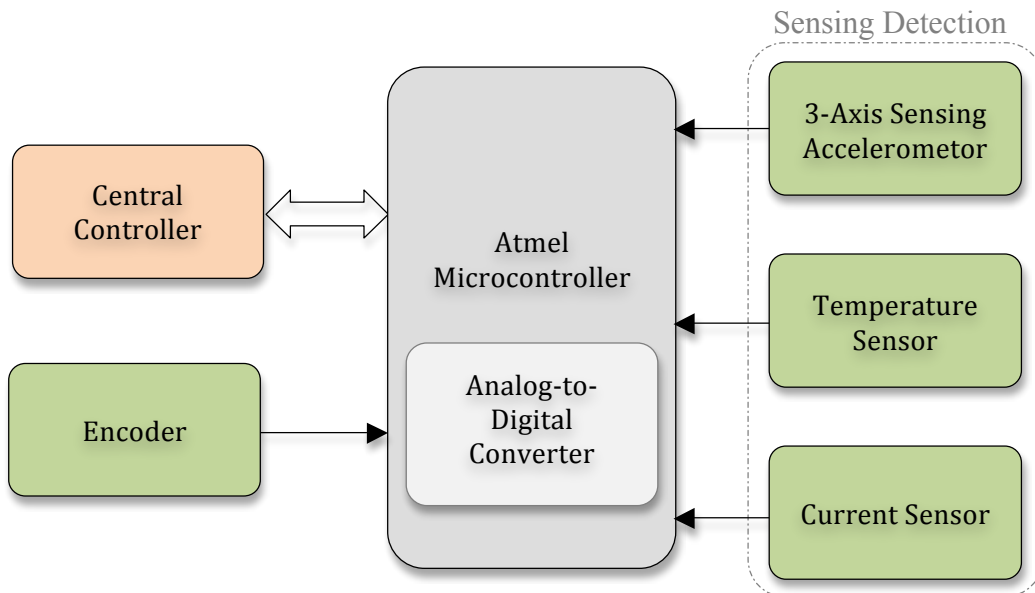


Figure 4-13: Architecture of sensing detection

Detailed discussions about the components used in the brushed DC motor driver are addressed below:

1. AT90CAN128
2. ATA6824 and IRLML0030 for H-bridge
3. LM35DM, INA139 and ADXL330 for sensing detection
4. Communication interfaces

4.5.1 Microcontroller AT90CAN128

The microcontroller is a minimum system with a Central Processing Unit (CPU), memory, clock timer, and I/O ports on a single Integrated Circuit (IC) that control electronic devices.

In brushed DC motor driver, AT90CAN128 is selected. It is a low-power Atmel AVR CMOS^{xvi} 8-bit microcontroller based on Reduced Instruction Set Architecture (RISC). It has read-while-write capabilities with 128 K bytes of In-System Programmable (ISP) flash. It has 4 K bytes of Electrically Erasable Programmable Read-Only Memory (EEPROM) and 4K bytes of Static Random-Access Memory (SRAM), with boot loader size of 8 K bytes [21].

Table 4-1: Comparison between different microcontrollers [21], [22].

<i>Device</i>	<i>Flash</i>	<i>EEPROM</i>	<i>RAM</i>	<i>Speed</i>	<i>ADC Inputs</i>	<i>Max I/O</i>
AT90CAN32	32 K Bytes	1 K Bytes	8 K Bytes	16 MHz	8	53
AT90CAN64	64 K Bytes	2 K Bytes	8 K Bytes	16 MHz	8	53
AT90CAN128	128 K Bytes	4 K Bytes	8 K Bytes	16 MHz	8	53
ATmega16	16 K Bytes	512 Bytes	4 K Bytes	16 MHz	11	27
ATmega32	32 K Bytes	1 K Bytes	4 K Bytes	16 MHz	11	27
ATmega64	64 K Bytes	2 K Bytes	8 K Bytes	16 MHz	11	27

Table 4-1 presents comparison between different types of microcontrollers with integrated CAN controller. The AT90CAN32, AT90CAN64 and AT90CAN128 differ only in memory sizes, and boot loader support. Selected microcontroller family AT90CAN32/64/128 differs with the one from family ATmega16/32/64 in number of ADC inputs and pin configuration. Both compared microcontrollers have the same maximum clock speed (16 MHz).

^{xvi} - Acronym of Complementary Metal Oxide Semiconductor

Figure 4-14 shows pins configuration from family AT90CAN128. This microcontroller has 7 ports, namely PA₇ – PA₀, PB₇ – PB₀, PC₇ – PC₀, PD₇ – PD₀, PE₇ – PE₀, PF₇ – PF₀, and PG₄ – PG₀. Where pins from port F (PF₇ – PF₀) can be initialized as analog inputs to the ADC.

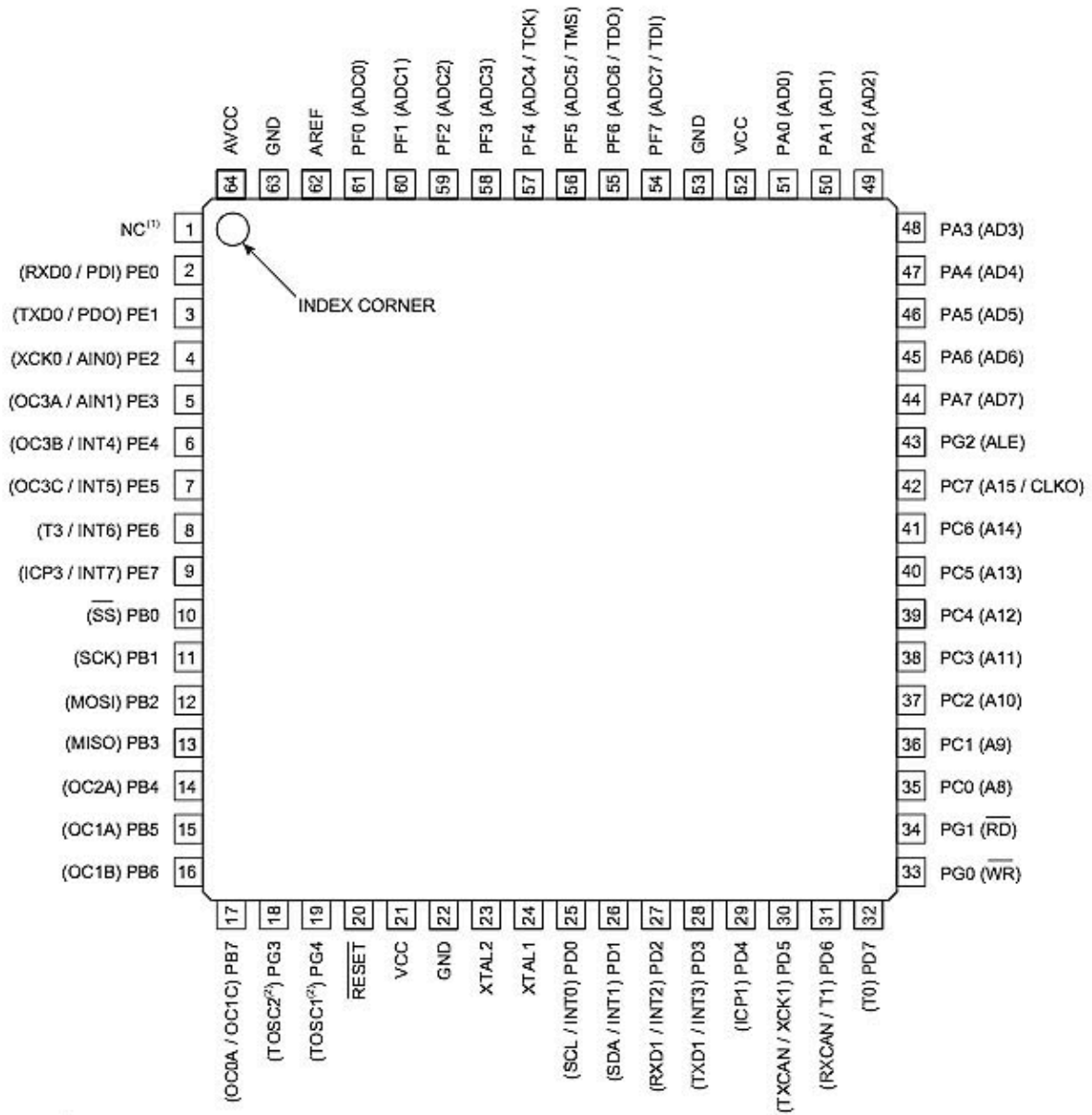


Figure 4-14: Pins configuration of AT90CAN128 [21]

Two quartz oscillators are placed nearby microcontroller. Quartz oscillator (16 MHz) with two parallel capacitors is connected to AT90CAN128 pin 23 (XTAL2) and pin 24 (XTAL1), and second quartz oscillator (32 kHz) for the internal asynchronous timer/counter that is connected to pin 18 (TOSC2) and pin 19 (TOSC1).

The TLE6250G differential CAN bus transceiver from Infineon is used to work as an interface between the AT90CAN128 pin 30 (TXCAN) and pin 31 (RXCAN), and the physical differential bus (CAN high and CAN low).

Circuit is powered by two different voltage sources, one that outputs 8V and the other that outputs 24V. An 8V supply is distributed over CAN network cable (RJ-45 connector, pin 7), while 24V power supply for the H-bridge is connected separately. A 5V regulator (LM7085) is needed to drop the supply voltage from 8 to 5V for the microcontroller.

Diode in front of the LM7085 is used to protect the circuit from wrong polarity supply, while for the input voltage stabilization and from the output side of the LM7085 voltage supply capacitors are used. Additional 3.3V regulator (LM3480) is need for the acceleration sensor.

4.5.2 H-Bridge

There are many choices, which transistor to use for H-bridge. The chosen transistor IRLML0030 from International Rectifier is an n-channel power MOSFETs^{XVII} with a drain source voltage up to 30V. It has a good efficiency at low voltages with fast switching. MOSFET IRLML0030 needs approximately 5V at its gate, and fully opens the gate at 10V. An IRLML0030 has a maximum static drain-to-source resistance of 0.027 Ω , which behave very much like an ordinary switch when turned on.

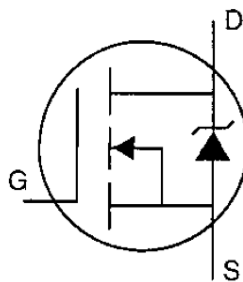


Figure 4-15: Schematic symbol of an n-channel MOSFET

^{XVII} - Acronym of Metal Oxide Semiconductor Field Effect Transistor

When H-bridge is operating, from four transistors (Q1 to Q4), two diagonal transistors are switched ‘on’ from the PWM signal in order to control the motor direction of rotation, while other two are turned ‘off’. In the circuit are four additional protection diodes (D1 to D4) that provide a safe path for the BEMF energy from the motor to dissipate and protect the integrated circuit (IC). At the same time they reduce power loss during the non-overlap time when neither of the high side or low side transistors is on [23]. The circuit can provide up to 4A output of continuous drain current.

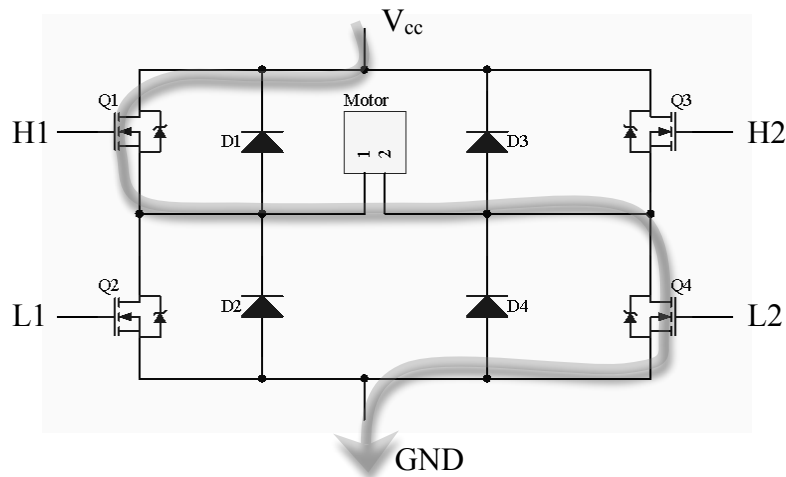


Figure 4-16: Motor in H-bridge rotates in a clockwise direction

When selecting another pair of transistors to be active, rotation changes in opposite direction (see Table 4-2).

Table 4-2: H-bridge truth table

Q1	Q2	Q3	Q4	Description
0	0	0	0	Standby mode
1	0	0	1	Motor clockwise
0	1	1	0	Motor counter clockwise
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes

Dedicated chip ATA6824 from Atmel is used to drive MOSFET’s, which is optimized for H-bridge configuration. A resistor in the gate line (see Appendix B-2) of a MOSFET protects ATA6824 from causing high voltage ($L \frac{d}{dt} i_a$) spikes. A side effect of using a resistor is that it slows down turn ‘off’ switching.

Figure 4-17 shows pins configuration of ATA6824. Generated PWM signal from AT90CAN is connected to the PWM port (pin number 11) of ATA6824. Pins 14, 15 and 16 provide diagnostic output information's for microcontroller. If there are any errors detected, such as short circuits, voltage failures, and over-temperature, the ATA6824 switches the outputs off. Pin numbers 18 and 20, of ATA6824 provide gate voltage for high-side switches (Q1 and Q3), and corresponding pins 26 and 27 for low-side switches (Q2 and Q4) used in H-bridge (see Appendix B-2).

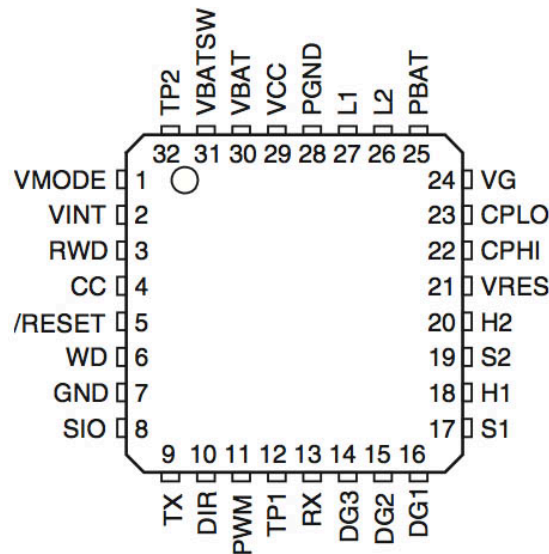


Figure 4-17: Pins configuration of ATA6824 [20]

The particular ATA6824 used for MOSFET control includes additional pins for a microcontroller supply, a window watchdog, and a serial interface [20].

4.5.3 Sensing Detection

Electronic sensors are used to detect conditions of the physical environment. Therefore, they generate digital (*i.e.*, a binary output signal) or analog electrical signals (*i.e.*, a signal proportional to the quantity being measured).

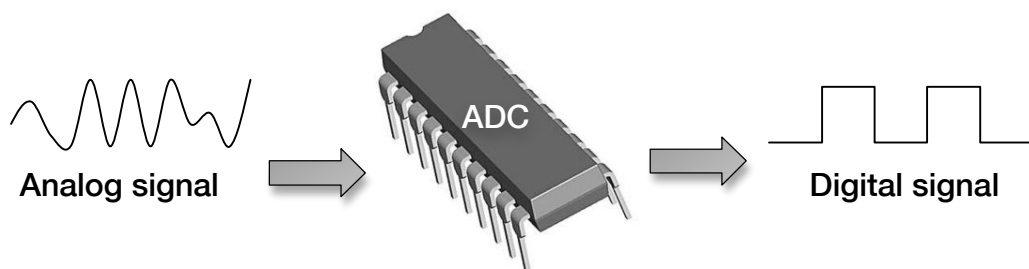


Figure 4-18: Analog to digital conversion

Analog electrical signal values are connected into ADC input, which converts this measurement into a 10-bit digital number (*i.e.*, using AT90CAN's ADC inputs) so that it can be processed by the microcontroller. On the brushed DC motor driver, temperature, motor (*i.e.*, current detection), and motion (*i.e.*, 3-axis detection) sensors are used. As well as external inputs for encoders that are fitted on the motor to determine the position of the motor shaft. These sensors have found to be quite useful in the robot for more advanced control algorithms.

LM35DM

A high precision integrated-circuit temperature sensor (*i.e.*, LM35DM, National Semiconductor) with ± 0.5 Celsius degree sensing precision is added on the board for measuring temperature. As the LM35DM has a voltage output that is linearly proportional to temperature it can be directly connected to the analog input of the microcontroller.

INA139

The added current shunt monitor chip INA139 made by Texas Instruments serves as the current sensor. The sensor measures the current passing through the shunt resistor. Because the current passing through the shunt resistor is most probably affected by PWM noise from the motor, signal output is filtered (*i.e.*, low-pass filter) before the measurement.

ADXL330

3-axis accelerometer sensor (*i.e.*, ADXL330, Analog Devices) that is used in the motor driver can measure the static acceleration of gravity, as well as dynamic acceleration resulting from motion, shock, or vibration [24]. Proportional to change of acceleration, tilting accelerometer around any axis (X, Y, Z) will change the output signals (*i.e.*, analog values), which are transmitted into the microcontroller.

ENCODER

To determine position and velocity from the motor, two connectors with different inputs are available. One connector can receive signals from the incremental encoder that generates square wave (an A and a B outputs) pulses that are 90 degrees out of phase, and second connector that can receive signals (with a unique code for each position) from the absolute position encoder.

4.5.4 Communication Interfaces

On both motor driver boards, a few different connectors can be found. They are used for connecting subsystems together with other circuit boards and communication between them. The central controller sends appropriate data to the motor driver through CAN or RS-232 communication bus. Driver circuits are wired in a daisy-chain arrangement via CAN-bus. In this arrangement multiple joint controllers are connected in a linear topology.

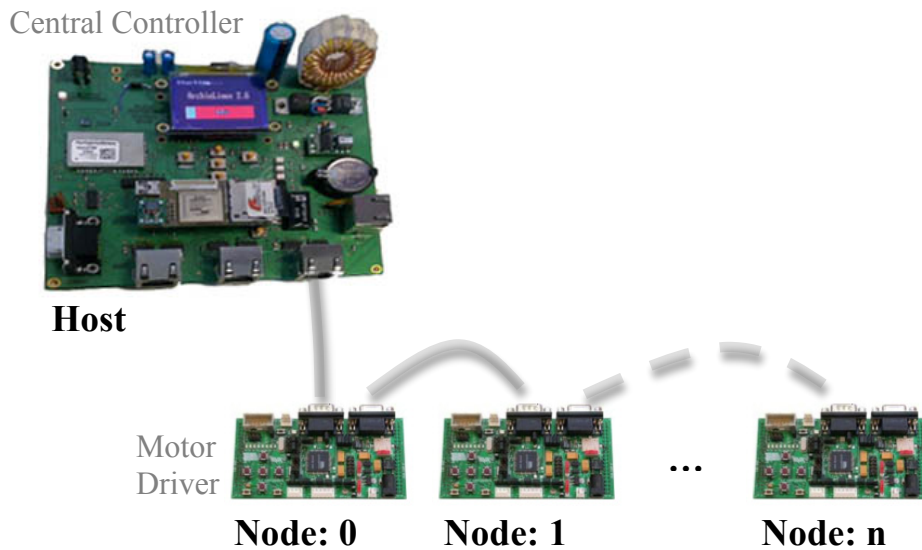


Figure 4-19: Daisy-chain network connection (linear bus topology)

Linear bus topology (see Figure 4-19) requires less cabling than a star topology (a point-to-point connection). Nodes are connected with standard RJ-45 Ethernet connectors and wired with unshielded twisted-pair (UTP) Ethernet cabling. Two RJ-45 connectors are placed on each board for daisy chain wiring. Located on the opposite side from motor power connectors, to not interfere with power pins.

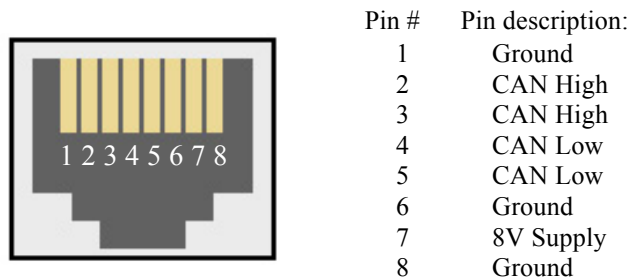


Figure 4-20: RJ-45 wiring connection pin-out for CAN communication

For debugging and testing purpose the motor driver can be connected to an external computer (higher level control) through a RS-232 serial cable, using a 3-pin connector.

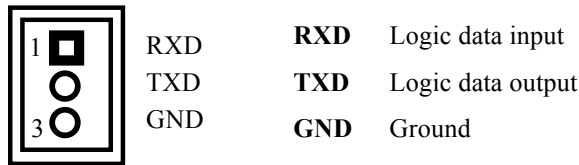


Figure 4-21: RS-232 three pin connector description

RS-232 uses a single-ended, bipolar voltage for data transmission, which differs from the one that is used in the circuit. RS-232 line driver/receiver SP232ACN from Sipex has been placed to the circuit to handle CMOS logic gate inputs from the controller and output RS-232 signals to the input logic levels, consistent with the RS-232 standard specifications. In addition to CAN and RS-232 connectors, the AT90CAN128 has both ISP and JTAG (Joint Test Action Group) connectors for programming the microcontroller.

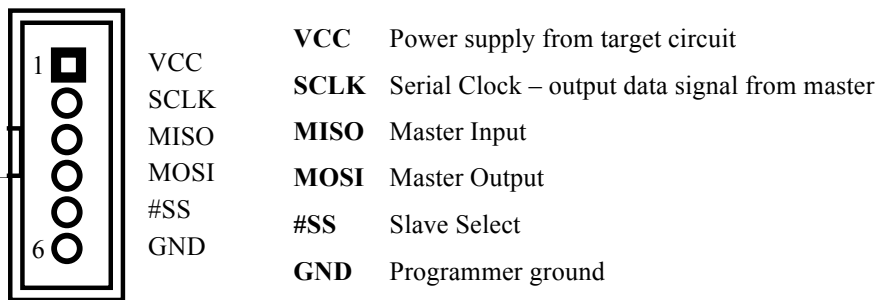


Figure 4-22: ISP pin description

ISP and JTAG buses provide access to all kinds of information (*i.e.*, fuse and lockbits settings), when diagnosis of hardware (*i.e.*, ATtiny24 and AT90CAN128) is needed (*i.e.*, boundary scan diagnostics) and for the purpose of debugging microcontroller.

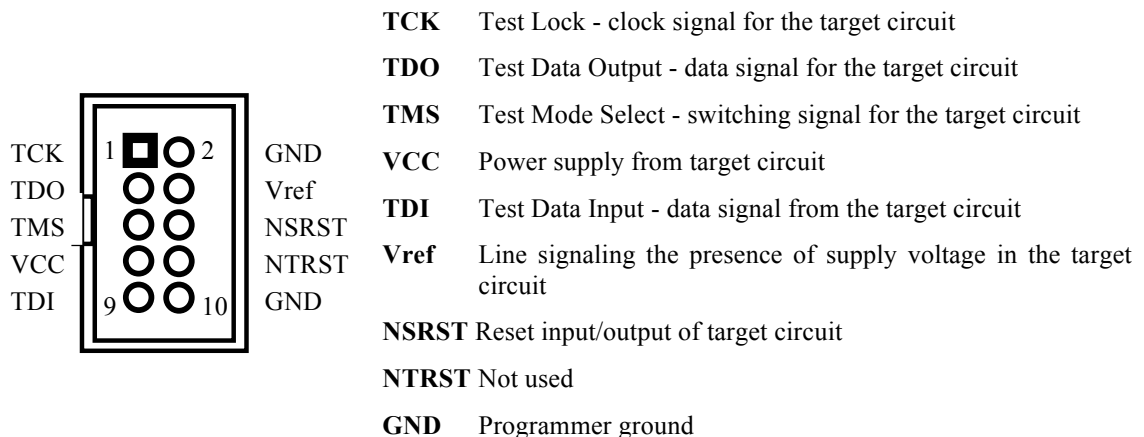


Figure 4-23: JTAG pin description

4.6 Design Evaluation and Discussion

Often ECAD-software^{XVIII} comes equipped with a huge library of components required to draw a schematic. In a schematic library can be found footprints needed for a PCB layout. Then again, sometimes there is a need to build own custom library of footprints, as they may not be found.

Components in the schematic library can be redesigned (*i.e.*, symbols and footprints) and directly linked to the schematic design document. Some of the components in this project have been created from scratch, while others may be just modified or taken from an existing library of footprints. For ease of use, components and footprints have been stored into a new schematic library document called ARCHI(v1) to be shared among team members interested in getting involved in circuit design and further improvements.

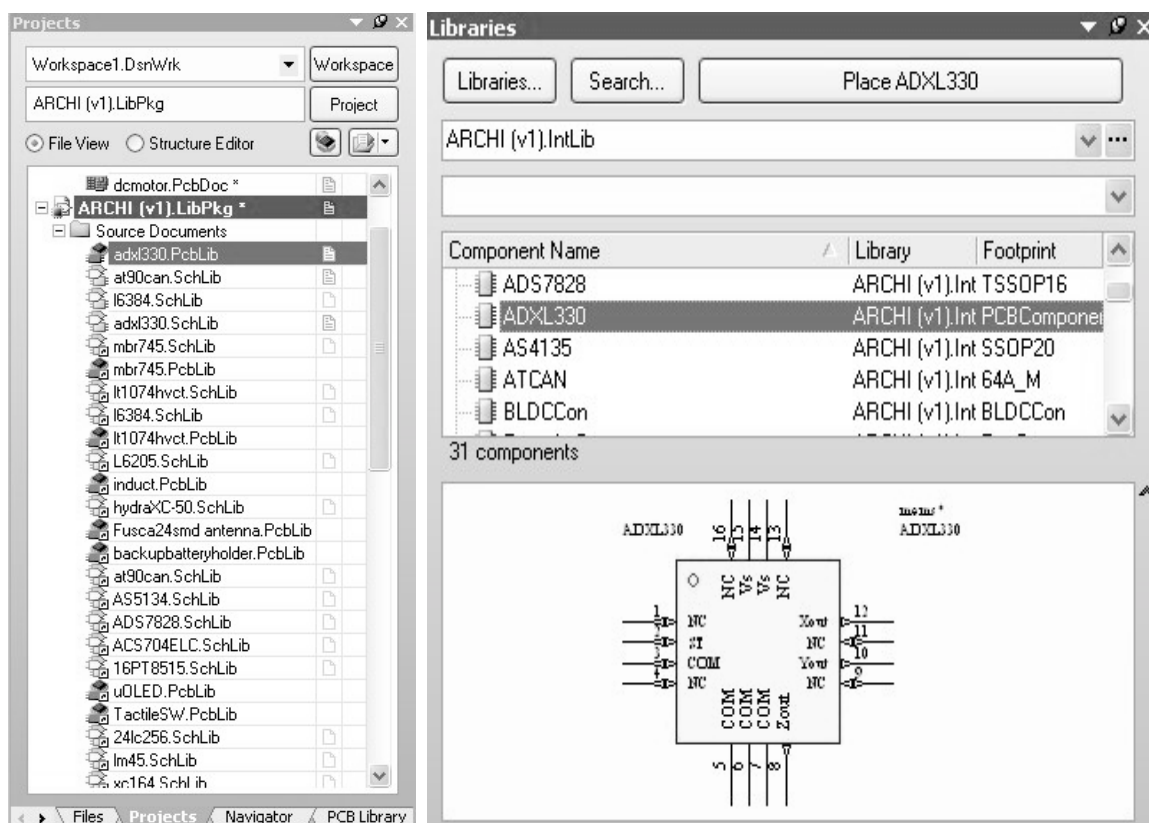


Figure 4-24: Custom-made component stored in a ARCHI(v1) schematic library

^{XVIII} - A list with a common ECAD-software can be found in Appendix A-1.

All new components added to the project are labeled with a distinct name (see Figure 4-24) along with indicated value (*i.e.*, pin numbers, part values, polarities, and attributes). Each electronic component stored in a schematic library has a footprint based on physical characteristics from technical datasheets.

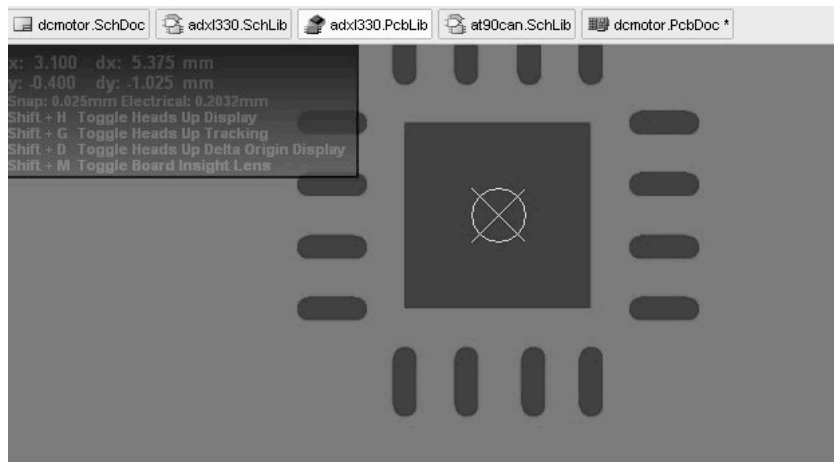


Figure 4-25: Custom-made footprint stored in a ARCHI(v1) footprint library document

To prevent future problems with wrong pin assignment from customized electronic components, signals are named in a way that makes it easy to understand the schematic.

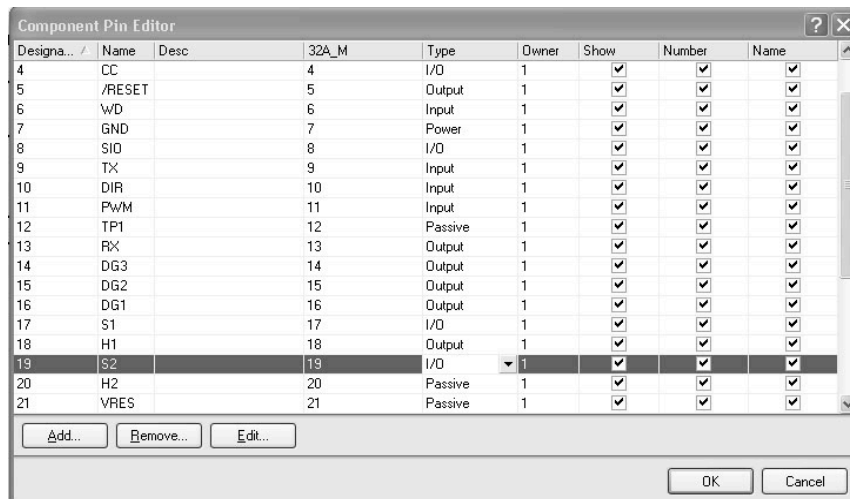


Figure 4-26: Electronic component pin inspection

Besides typical design process (briefly explained in Chapter 4.1), from drawing schematic, creating customized libraries up to decisions made for adequate placements of components on the PCB there are a number of considerations to take into account when designing a prototype.

Obviously, the knowledge required is specific for each step of creating a prototype. Throughout design process of the brushed DC motor driver, some of below-mentioned constraints shown in Figure 4-27 have been considered in order to make the circuit more robust.

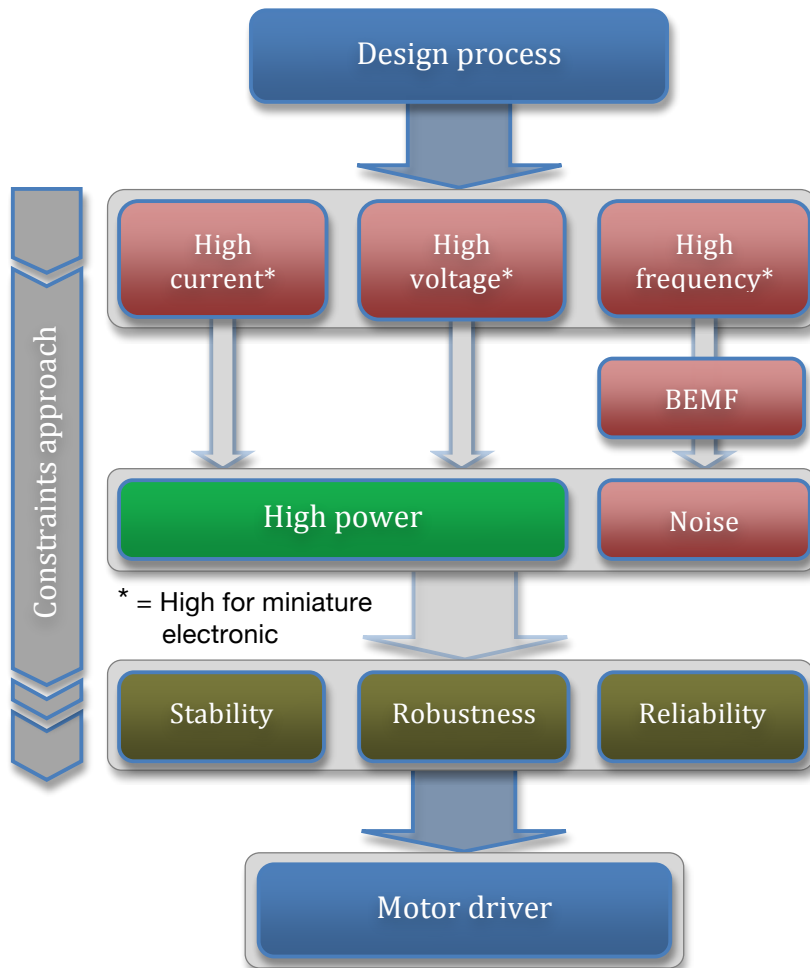


Figure 4-27: Constraints approach block diagram for designing a motor driver

As brushed DC motors are known for causing a wide range of electrical and magnetic noise in the circuit [25], there are situations in which noise can disturb signal transmission. Once inside the circuit, it can affect more sensitive signal traces. Generated electrical noise in the circuit can cause equipment shutdowns, false reading and could cause unreliable and unpredictable behavior.

In general, by adding a *low pass filter* where the signal has been converted to digital values can reduce noise that gets to the controller [26]. The damaging effect from an electrical noise in the power supply lines or other high impedance component of a circuit and ground loops (i.e., difference in voltage) are reduced or eliminated across the frequency band with *bypass capacitor* (i.e., monolithic ceramic capacitor). Bypass capacitor is wired in parallel setup between the power line (+V) and the ground. The same capacitor wired in series can be used to decouple one part of a circle from another (known as a *decoupling capacitor*).

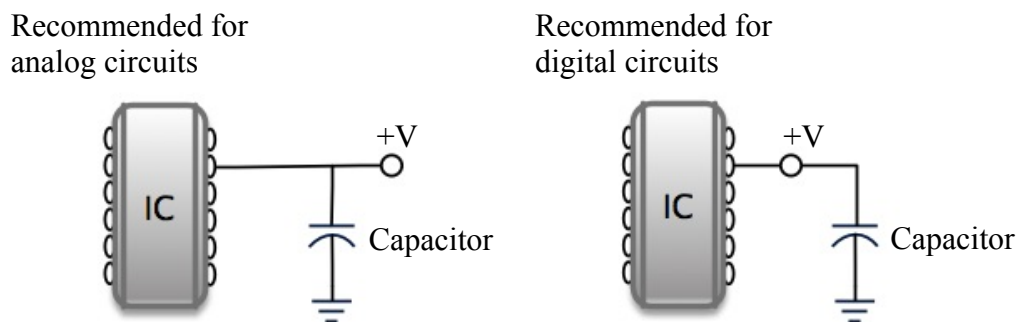


Figure 4-28: Bypass capacitors used to filter out the electrical noise. Based on [26].

To prevent unreliable and unpredictable behavior while the microcontroller *powers up* and *powers down*, ‘*pull-up*’ or ‘*pull-down*’ resistors are used. In this way, the input always has a ‘default’ state, even if nothing is connected to it.



Figure 4-29: Protection against random unreliable and unpredictable reading.

Because of the inertia in a running motor, a motor can act like a generator on deceleration, thus the BEMF should be minded. *Schottky diodes* protect the controller circuit from a motor BEMF voltage. They provide a low resistance path for collapse current to the motor, whenever inductive loads are switched off [27]. Additional *choke coil* reduces the PWM switching noise (i.e., EMI emission) caused by DC brush-type motors.

When creating a PCB layout in the ECAD-software it is possible to choose if routing is done with automatic and/or manual routing process. The best results are achieved in combination with automatic and manual routing - because our judgment of how to arrange circuitry is better.

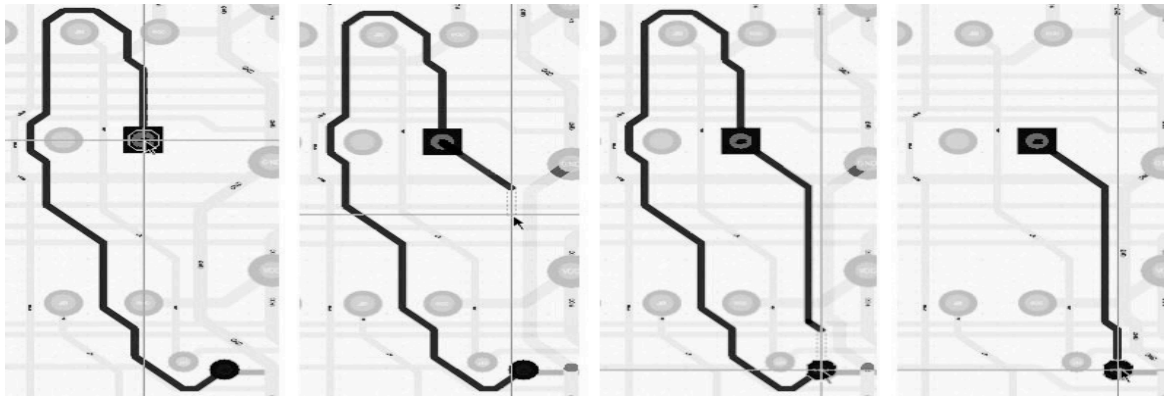


Figure 4-30: Trace manipulation. Images from [28]

Placing power traces away from a traces carrying signal (feedback, control and communication traces) minimizes parasitic effect and prevents that a magnetic field around will cause a false signals transmissions. With short signal traces and component leads, the motor driver can be immune on interference from other systems and minimizes the heat generated by the conductors [26]. Current loops can be reduced to a minimum and a proper ground system can be designed, when traces are manually manipulated.

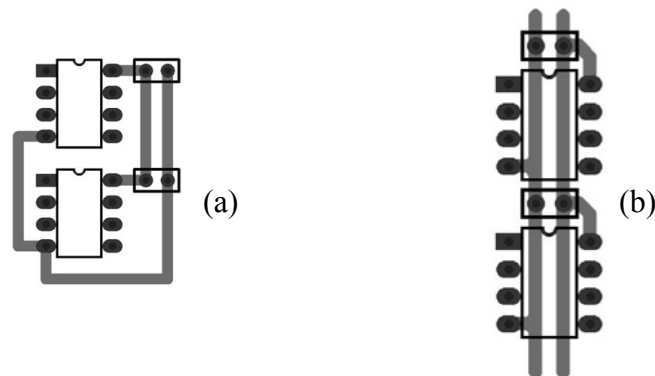


Figure 4-31: Trace manipulation with (a) good, and (b) bad routing [29]

In this project double-sided boards are used (*i.e.*, two layers of copper tracks, one on each side of the board). Production of single and double-sided boards is cheaper and if traces suffer damage it is easier to repair single/double-sided board than when using a multilayer board.

As this work has focused on PCB prototyping, the first design had only few components, with the challenge to learn using ECAD-software

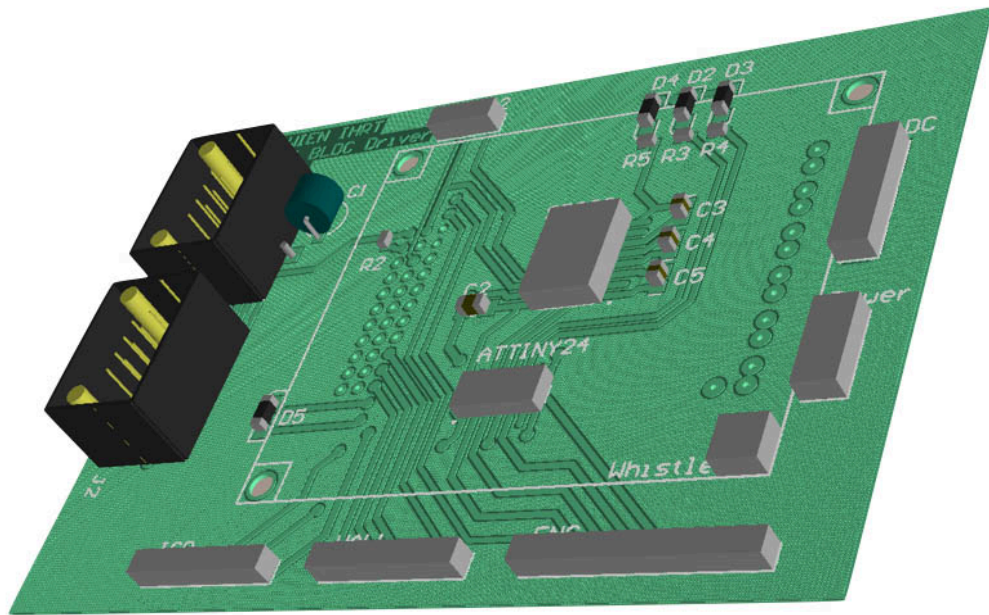


Figure 4-32: Front side 3-D view of the brushless motor driver during designing process

Light-emitting diodes (LEDs) are placed on circuit board to indicate the status of the controller (i.e., power status, bus traffic and errors).

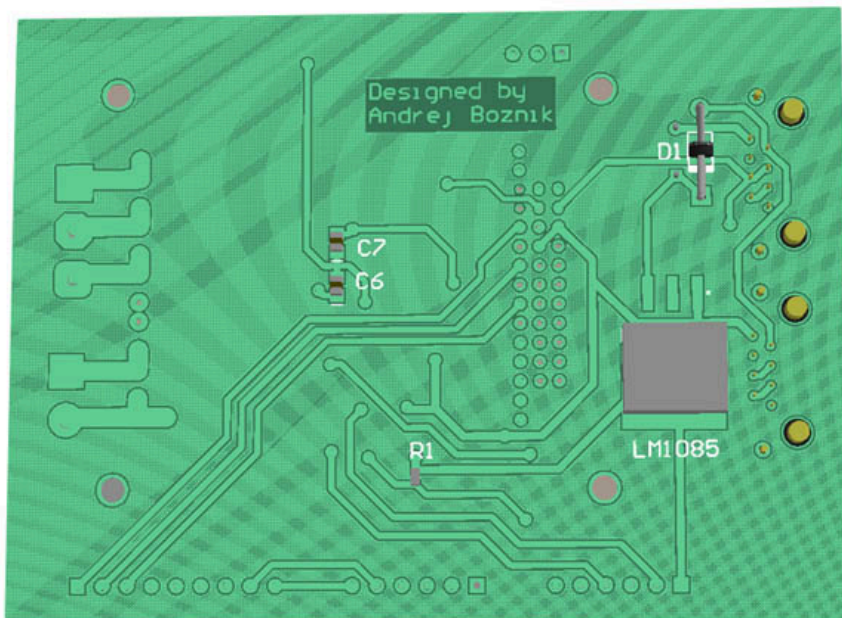


Figure 4-33: Backside 3-D view of the brushless DC motor driver

The second motor driver PCB (see Appendix D-2) differs from the first PCB meant to run brushless DC motors (see Appendix D-1). The main difference between them is that one uses integrated motor controller (Elmo Whistle), while second PCB uses its own microcontroller (AT90CAN) for the purpose of controlling brushed DC motor with H-bridge.

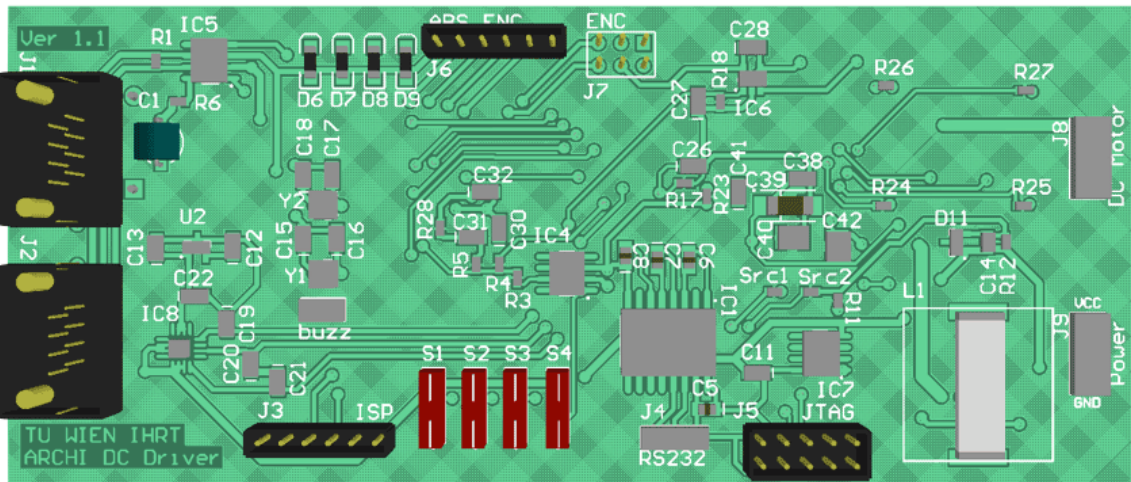


Figure 4-34: Front 3-D view of the brushed motor driver, with incomplete 3D bodies

The overall layout of the motor driver has been created with size still able to fit on the humanoid robot construction. All the components have been arranged as close to one another as possible with heat dissipation in mind. Heat from MOSFETs is dissipated via the aluminum skeleton chassis.

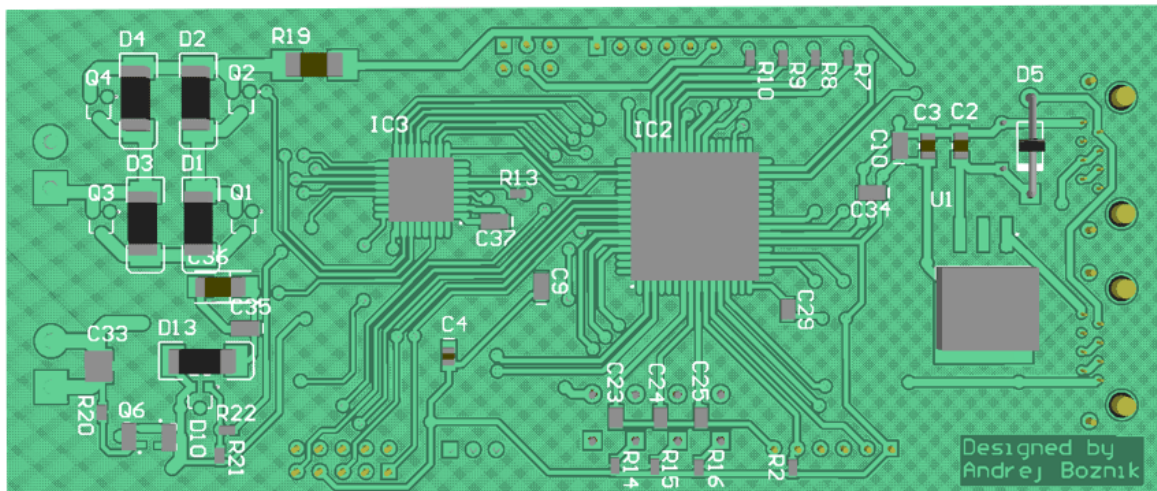


Figure 4-35: Backside 3-D view of the brushed motor driver

5 SIMULATION AND EXPERIMENTAL RESULTS

5.1 Experimental Setup

Laboratory tests have been carried out on the brushed DC motor controller (A. Byagowi, Archie ver. 1.7), and DC motor (*i.e.*, Faulhaber, 3257 series) combined with the planetary gear drive (Faulhaber, 38/1) described in Chapter 3.1.

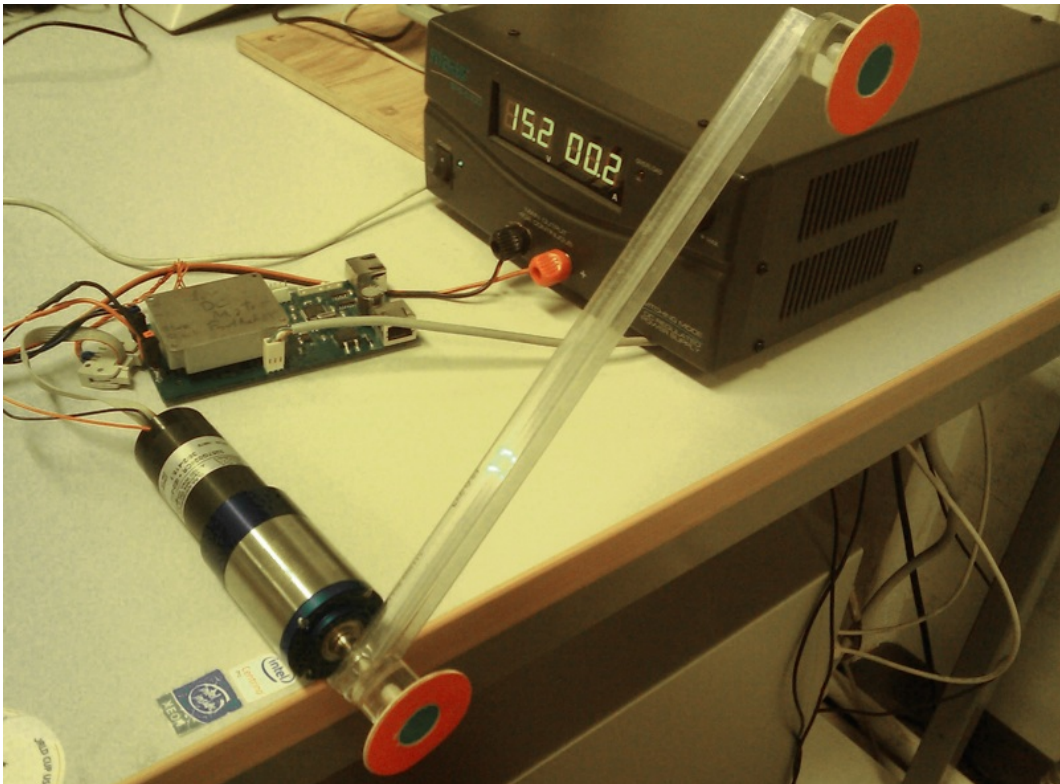


Figure 5-1: Picture of an actual experiment

Angular position of swinging link mounted on the output shaft of the motor gear drive in this experiment is aimed to rotate 0 to 0.5π radians and depends on given velocity during an observation period of 4seconds.

Encoder with 2048 lines/rev attached to the shaft of the DC motor is used to measure the angular position of the shaft. The motor controller received each parameter values from a computer via the serial (RS-232) port.

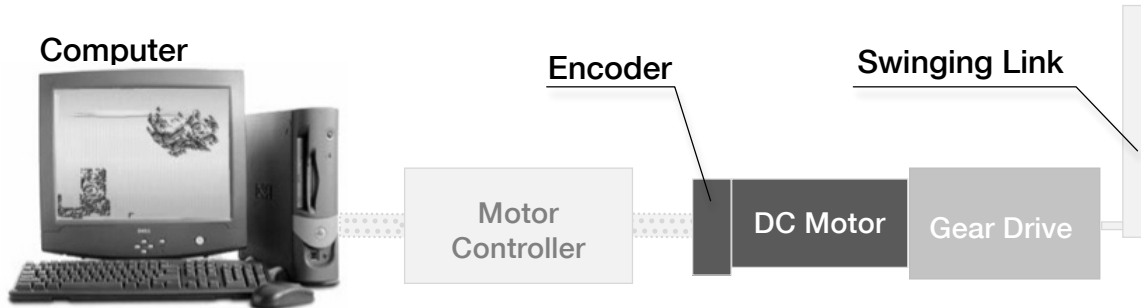


Figure 5-2: Motor controller connected to a DC motor with a gear drive and encoder mounted on the motor.

As the motor controller interface, the standard Motion Monitor Toolset (Elmo MC Composer, ver. 2.19) has been used. The parameters were recorded in the textbook and converted into values that are presented in graphs. The motor controller was tested with the debug consoles shown in the below Figure 5-3.

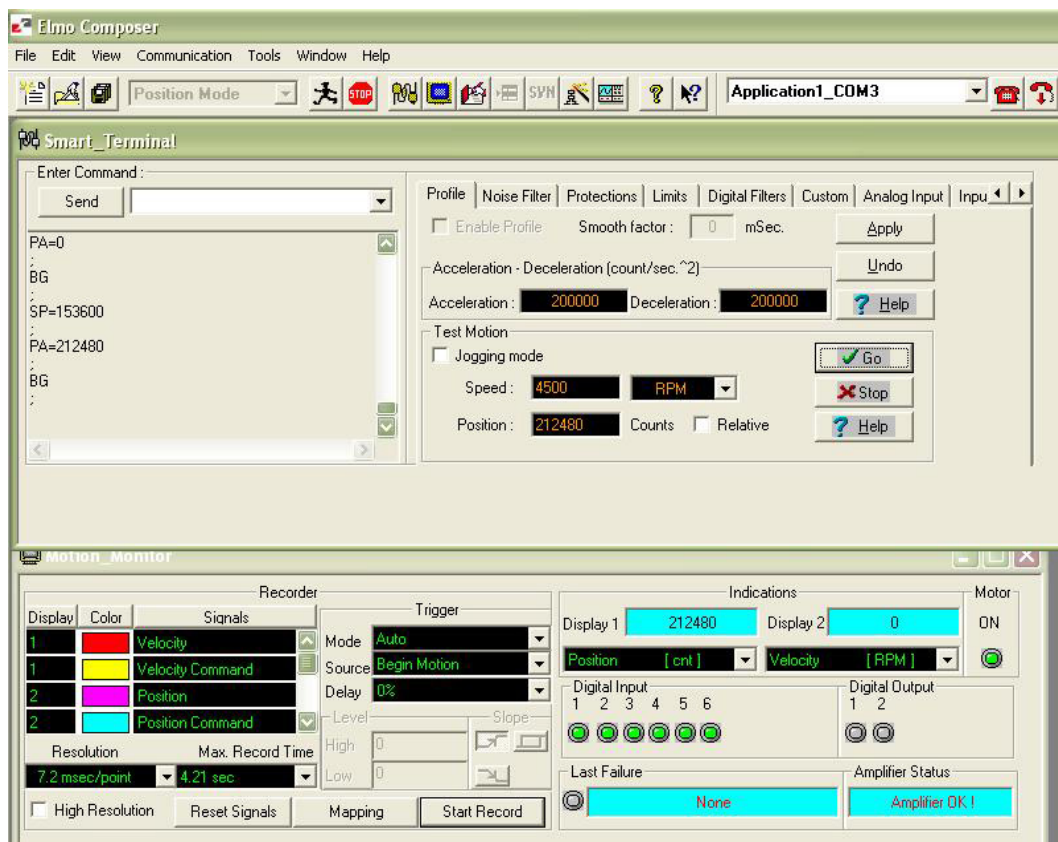


Figure 5-3: Elmo MC Composer application interface

This motor controller is based on the *Elmo Whistle* (Elmo Motion Control, 2006) controller. Elmo Whistle has a three-cascaded control loop to control the position, velocity, and the torque of the motor. The proportional (P) compensator is used for position control.

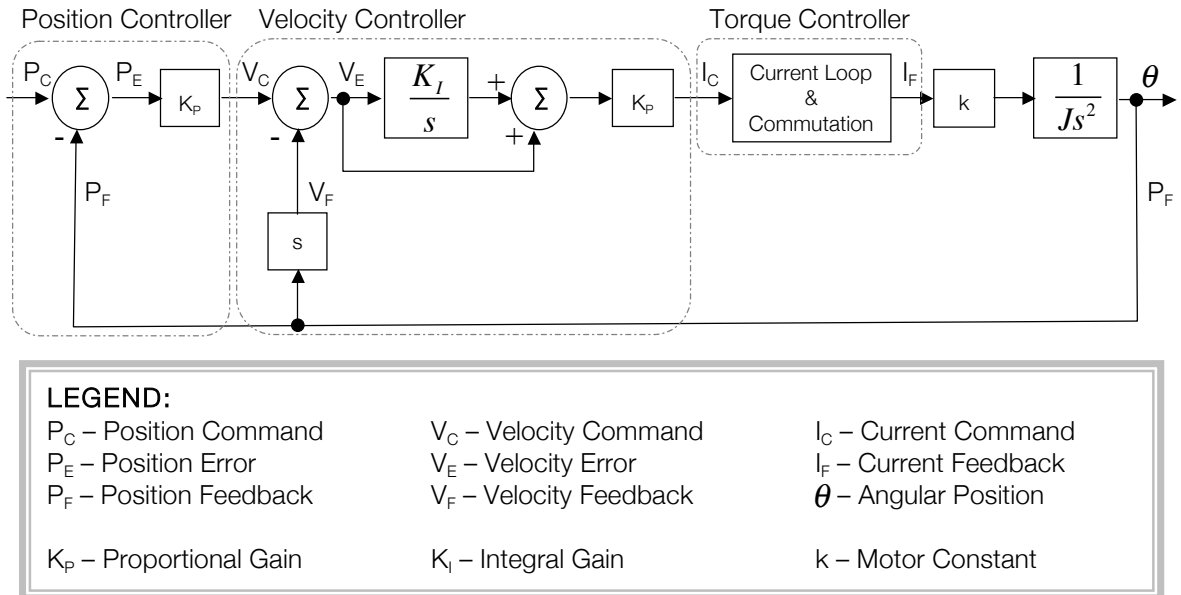


Figure 5-4: Three cascade control loop (simplified block diagram) [30]

For velocity control (closed loop) the proportional-integral (PI) compensator is employed with a feed forward cancelation for maintaining a constant velocity value and minimizing the output from PI compensator.

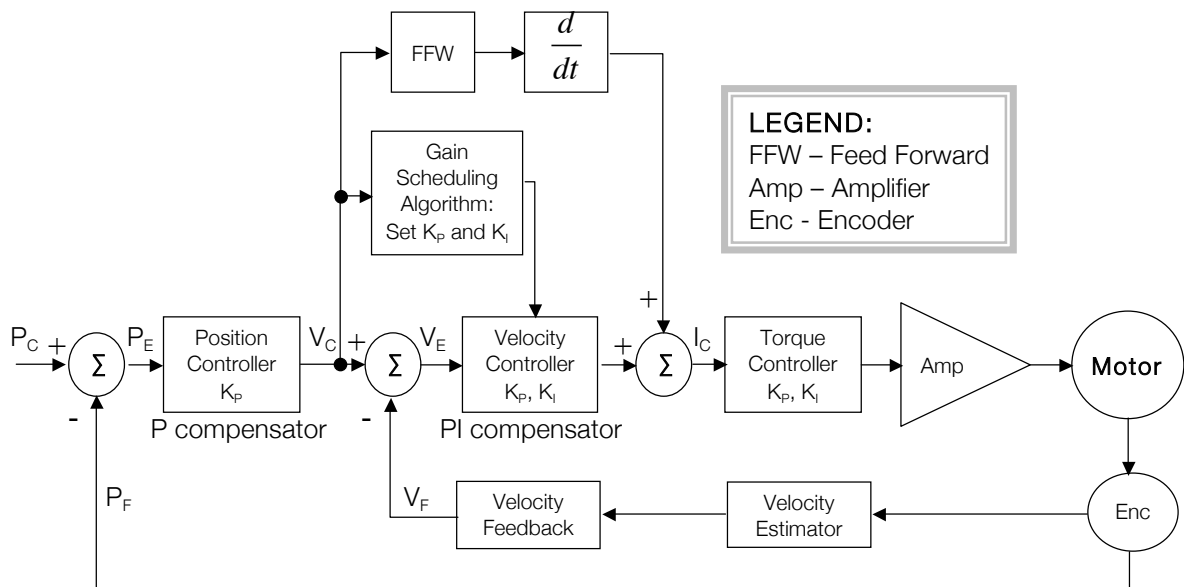


Figure 5-5: Simplified block diagram of Elmo Whistle motion controller, based on [30]

5.2 Experimental Results

The purpose of testing a control system is to find suitable parameters for a stable control. The simulation of a digital controller is designed with transfer function, Equation (3-18), between the input voltage $V_a(s)$ and the motor angular velocity $\dot{\theta}_M(s)$:

$$G_v(s) = \frac{\dot{\theta}_M(s)}{V_a(s)} = \frac{k_t}{((L_a \cdot s + R_a) \cdot (J \cdot s + b) + k_b \cdot k_t)} \quad (5-1)$$

Feed forward cancelation is utilized by feeding the reference signal into the inverse model of the DC motor using Equation (5-1) and is expressed as follows:

$$G_{v,inv}(s) = \frac{V_a(s)}{\dot{\theta}_M(s)} = \frac{((L_a \cdot s + R_a) \cdot (J \cdot s + b) + k_b \cdot k_t)}{k_t} \quad (5-2)$$

where:

- inductance of the coil (L_a) = $0.27 \cdot 10^{-3}$ H
- resistance of the motor armature (R_a) = 1.63Ω
- torque constant (k_t) = $37.7 \cdot 10^{-3}$ Nm/A
- BEMF constant (k_b) = $37.7 \cdot 10^{-3}$ Vs/rad
- moment of inertia (J) = 41 gcm^2
- damping factor (b) = $26 \cdot 10^{-5}$ Nms/rad
- settling time for velocity controller ($T_s = \frac{1}{350 \text{ Hz}}$) = 0.00285 s
- settling time for position controller ($T_s = \frac{1}{80 \text{ Hz}}$) = 0.00125 s

From Equation (5-1), a transfer function can be expressed as:

$$G_v(s) = \frac{\dot{\theta}_M(s)}{V_a(s)} = \frac{37.7 \cdot 10^{-3}}{(1.107 \cdot 10^{-8})s^2 + (6.69 \cdot 10^{-5})s + 1.845 \cdot 10^{-3}} \quad (5-3)$$

Therefore, the continuous transfer function from Equation (5-3) needs to be converted into a discrete transfer function to discretise the continuous model:

$$G_v(z) = \frac{\dot{\theta}_M(z)}{V_a(z)} = \frac{1.464z + 87.36 \cdot 10^{-3}}{z^2 - 0.924z + 3.31 \cdot 10^{-3}} \quad (5-4)$$

Closed loop response of a transfer function from Equation (5-4) from the velocity controller is shown in Figure 5-6.

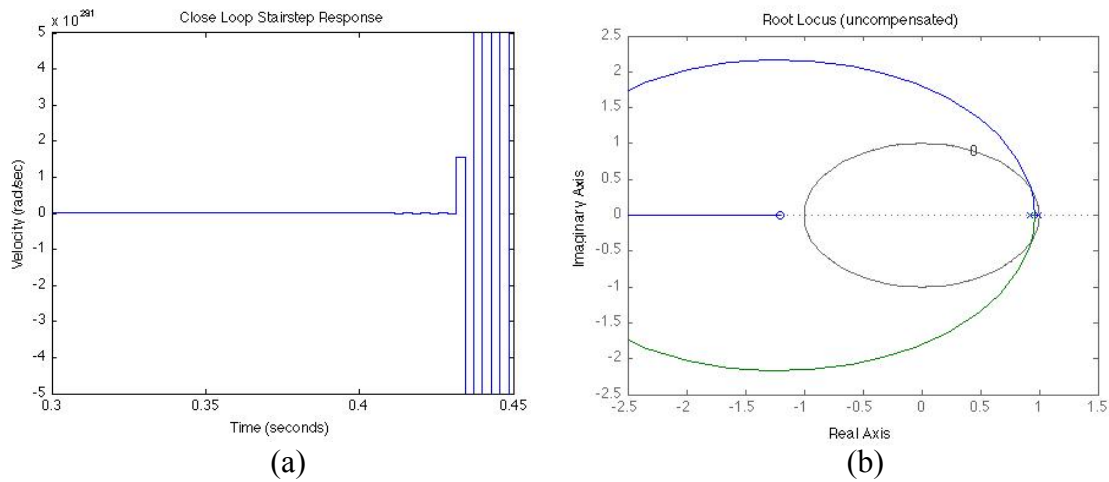


Figure 5-6: Stair step response (a), and Root Locus (b) of the closed loop without added compensator (simulation)

From Figure 5-6 (a) it can be seen that the close loop response of the system is unstable. Therefore, there must be something wrong with the compensated system. Using the root locus graphical method, the system stability can be defined by determining the position of system poles and zeroes in the s-plane of a closed loop control system. The system is unstable because one pole lies outside the unit circle, see Figure 5-6 (b). To make the system stable, the pole location needs to be changed by changing the compensator design.

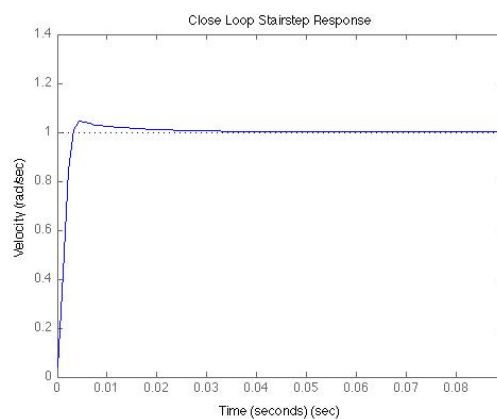


Figure 5-7: Stair step response of the compensated closed loop control system (simulation)

To improve the control algorithm, the PI compensator requires some tuning of K_P and K_I gain parameters. By modifying the PI controller gain parameters, characteristics of the motor response can be influenced (rise time, steady-state error, peak overshoot and settling time). In velocity controller, gain parameters are set to $K_P = 53$ and $K_I = 3873$. Followed by tuned velocity controller, the gain for outer loop position controller is adjusted.

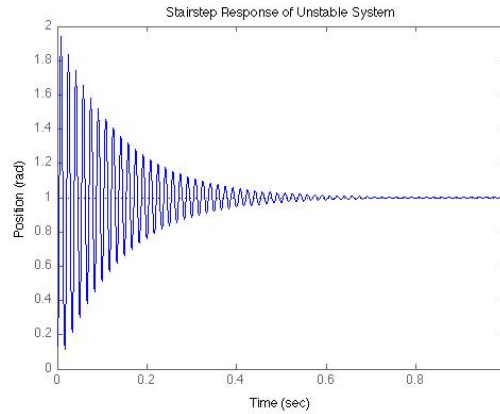


Figure 5-8: Stair step response of the open loop control system with to high K_P

Cascaded to the velocity controller in outer loop is placed the P compensator. The outer loop parameter of position controller is set to $K_P = 61$. Those parameters bring the system closer to the desired stable system, where the control algorithm compensates different loads values that occur on the motor. The stair step response of the compensated outer control loop is shown in Figure 5-8, with rise time about 0.04s, overshoot level about 12%, and the settling time 0.3s

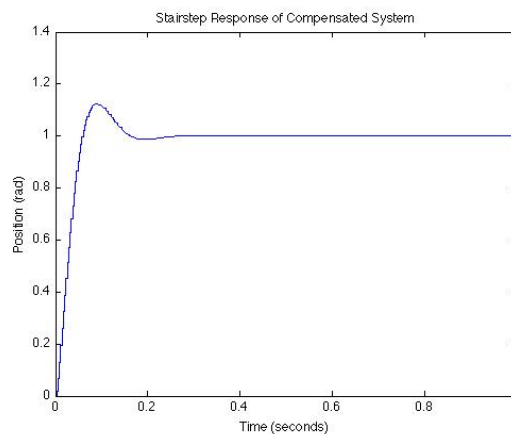


Figure 5-9: Stair step response of the compensated open loop control system (simulation)

Actual experimental results of the response captured from encoder attached on the motor shaft are illustrated graphically in following Figures 5-10 to 5-14.

Motor is limited with angular velocity of (a) 500-rpm, (b) 1000-rpm, (c) 2000-rpm, and (d) 3000-rpm. Figure 5-10 compares the given command response with the ideal integral compensated response of the motor controller. The relationship between real position and position command is simply that the motor controller is tested to verify the similarity between the given command and the real motor controller response.

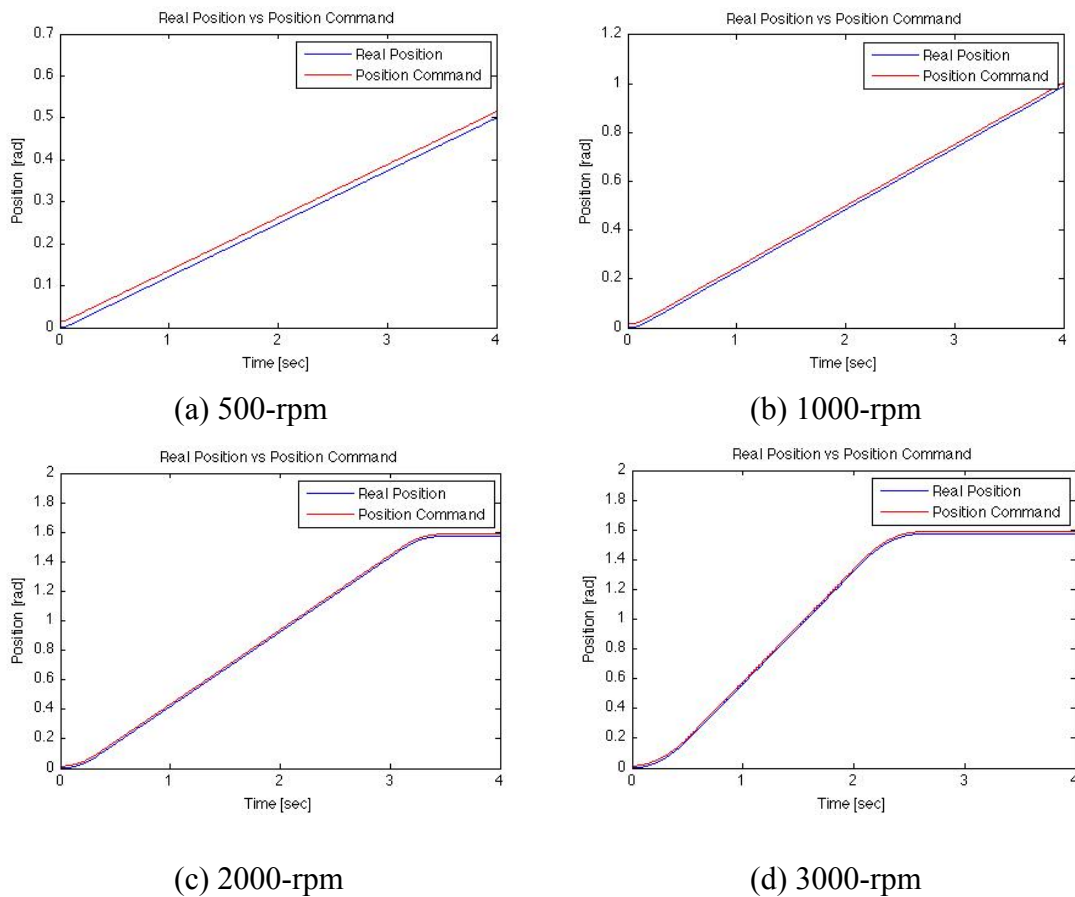
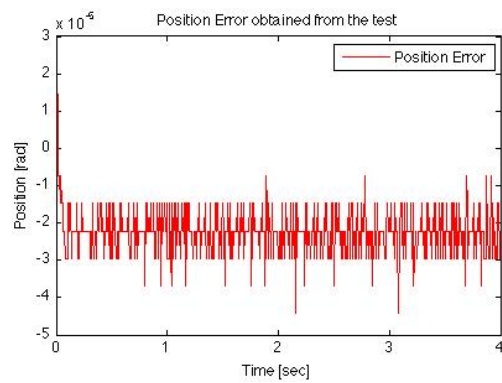


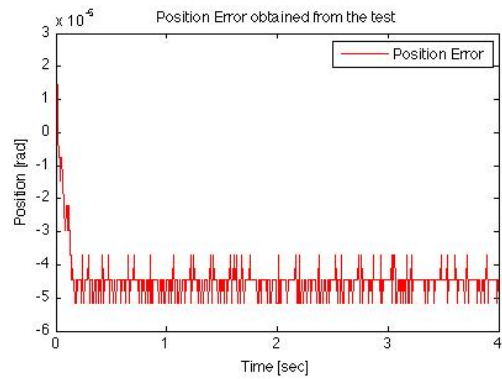
Figure 5-10: Given position command from debug consol vs. real position obtained by encoder mounted to the motor of the compensated system

One third of the expected end position is visible in Figure 5-10 (a) as the recording time has not been long enough. In addition to reach 0.5π radians movement, given command should be at least three times longer. The same goes for Figure 5-10 (b), where two-thirds of 0.5π radian rotation is illustrated. The transient response of both the real and the given command is the same without any oscillations.

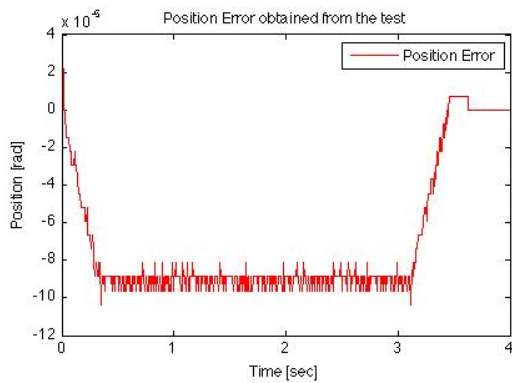
Difference between two values (given vs. real position) shown in Figure 5-10 is called position error and is illustrated in Figure 5-11. It is important that error is as minimum as possible, in order to obtain a suitable control performance.



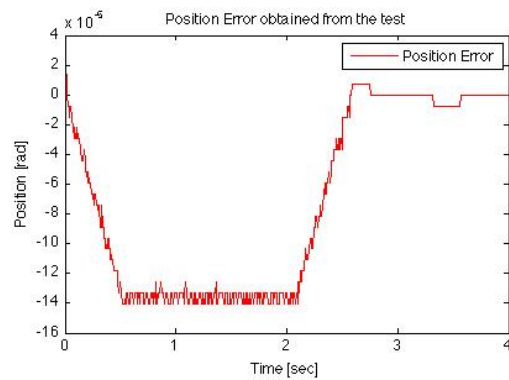
(a) 500-rpm



(b) 1000-rpm



(c) 2000-rpm

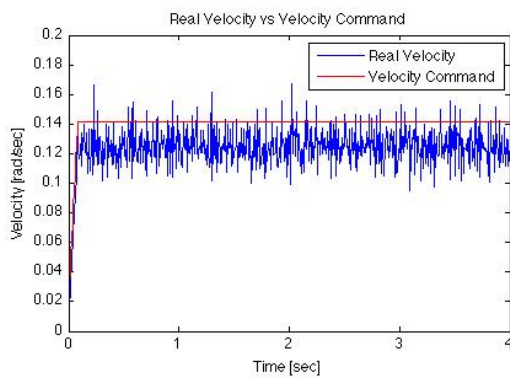


(d) 3000-rpm

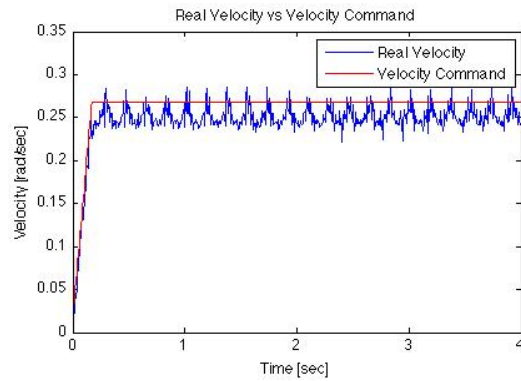
Figure 5-11: Position error of the compensated system

Significant deviation is seen in Figure 5-11 (c) and (d) position error graph, when the motor reaches 0.5π radians and becomes inactive.

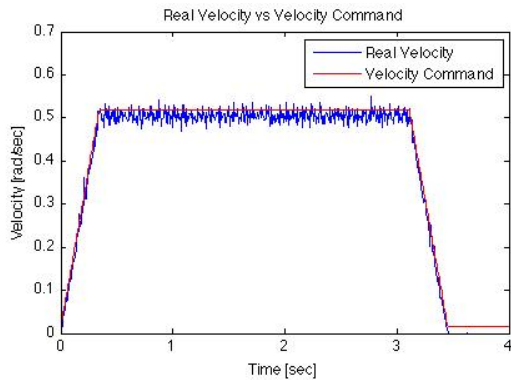
The following Figure 5-12 corresponds to Figure 5-10 and represents the relationship between velocity and time of the motor shaft at various speeds between given velocity command and real velocity response.



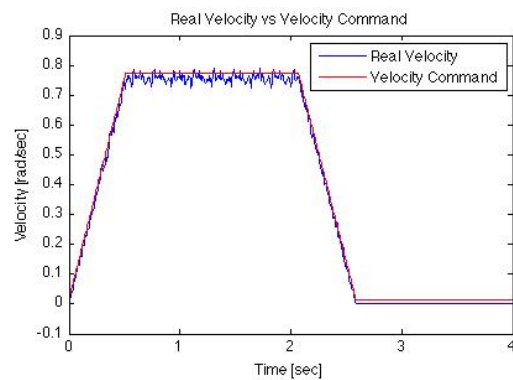
(a) 500-rpm



(b) 1000-rpm



(c) 2000-rpm

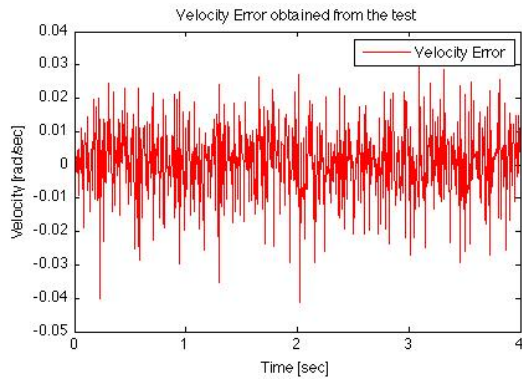


(d) 3000-rpm

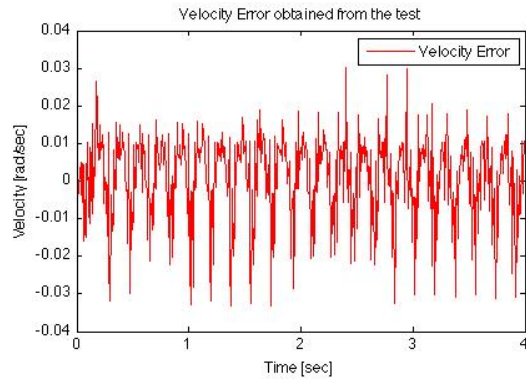
Figure 5-12: Given velocity command from debug consol vs. real velocity obtained by encoder mounted to the motor (compensated system)

Through simulation motor oscillates around desired value. This indicates that gain parameters in velocity controller could be further tuned (decreasing K_P gain). There is descending step seen in Figure 5-12 (c) and (d), meaning that swinging link has reached its final position of 0.5π radians.

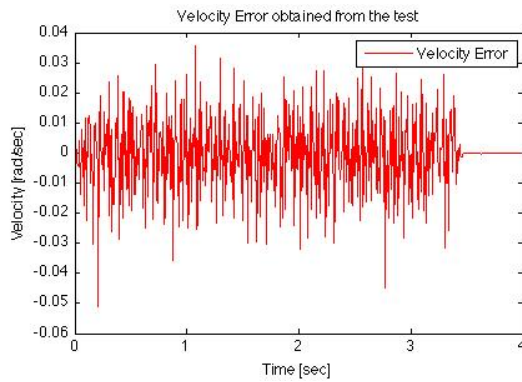
Following given information from data obtained in Figure 5-12 about actual real velocity, Figure 5-13 illustrates the velocity error (given vs. real velocity).



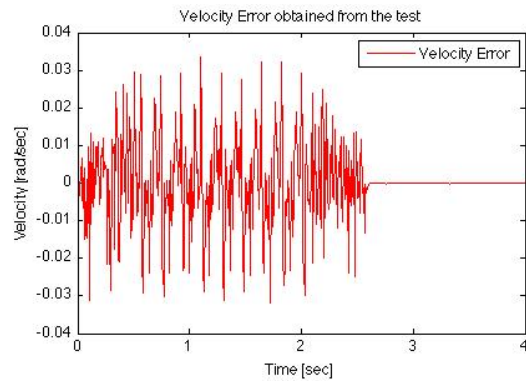
(a) 500-rpm



(b) 1000-rpm



(c) 2000-rpm



(d) 3000-rpm

Figure 5-13: Velocity error of the compensated system - continued

In Figure 5-14 we have looked at various responds of current consumption when different velocities are applied to the motor. It can be concluded that electric current consumption is increased with increasing a running speed of the motor. There are notable steady-state errors in current consumption, which is expected due to chosen parameters. In Figure 5-14 (a) and (b) ongoing current consumption trough a motor have never stopped during the period of 4 seconds, as desired position has not been reached.

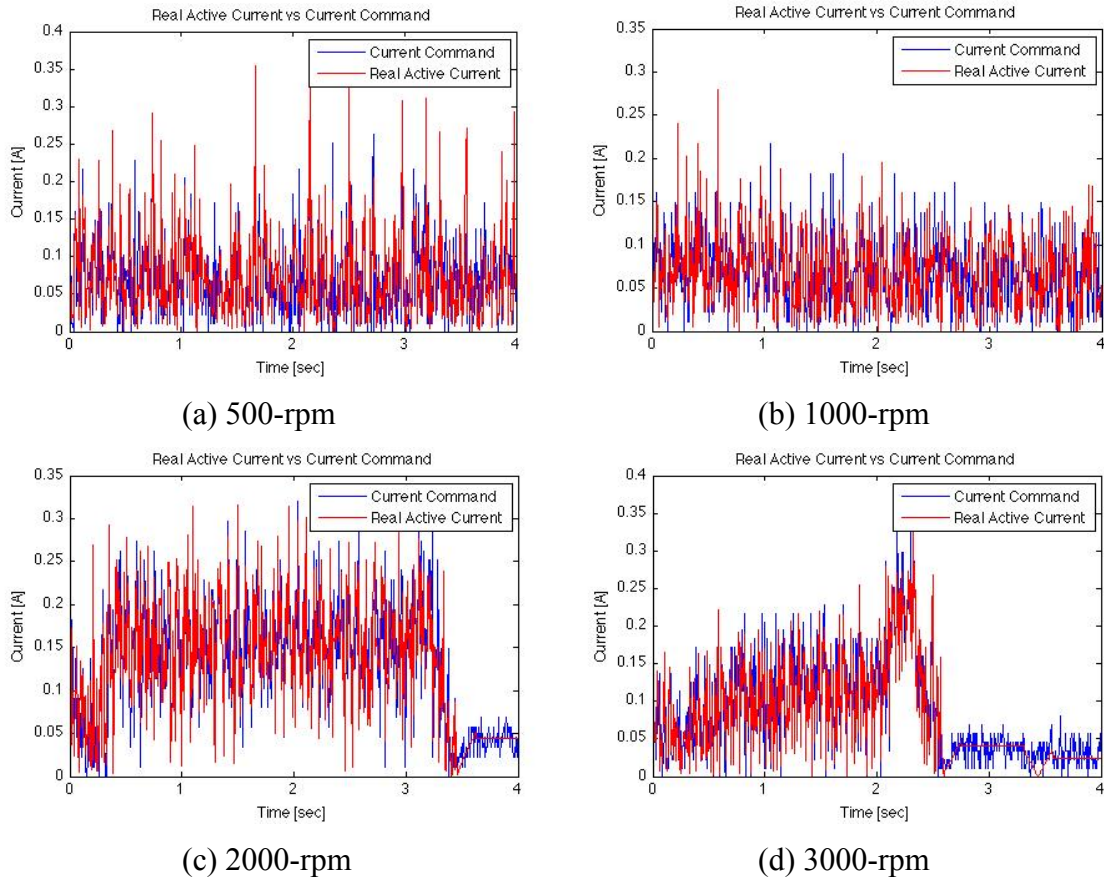


Figure 5-14: Given current command from debug consol vs. real current measurement in motor controller (compensated system) - continued

As seen in Figure 5-14 (c) and (d) it becomes inactive, after a certain amount of time. When compared with previous two Figures 5-10 and 5-12, it appears that change is associated with end of motor command. When motor reaches desired position the current consumption falls, see Figure 5-14 (c) and (d).

6 CONCLUSION

In this thesis, a motor driver design for a humanoid robot has been presented. It started with a description of the implemented communicational protocol and then followed with the main aim of this project, prototyping of the hardware that would control motors inside the robot.

During the project there were several factors that led to redesign of both motor drivers. One of them was all of a sudden need to change the platform architecture to simplify the future programming of motion controller hardware. Basic idea with brushless DC motor driver was with the reuse of components, to serve as an economic advantage. After first board design completion, and with expressed desire to migrate to less complex, and cheaper solution, another board meant to run brushed DC motors uses ATMEL microcontroller. Students with no prior knowledge could easily learn how to control joints only with knowledge of the C language. Migrating to microcontroller saves cost per each motor driver and enable a wider range of use in a classroom (easier software algorithm programming using C language).

The motor driver prototypes represent the first step of a complete redesign of electronic modules, which communicates over CAN bus. As it was meant, the next step should be to build up and assemble language program that efficiently communicate with the rest of the hardware. Due to redesign of joint controllers, the data bus in central controller needs some modifications for transition to a standard Control Area Network (CAN 2.0A) using CANopen communication protocol.

REFERENCES

- [1] *Oxford English Dictionary*. Second Edition, [CD ROM] Oxford University Press.
- [2] *RobotWorx*. [Webpage] Available from: <http://www.robots.com>
- [3] *Corecon, Inc.* [Webpage] Available from: <http://www.coreconagvs.com>
- [4] *Intuitive Surgical, Inc.*, Photo of the DaVinci device. [Webpage] Available from: <http://www.intuitivesurgical.com> [accessed February 18, 2010]
- [5] *ASIMO by Honda*. [Webpage] Available from: <http://thestockmasters.com/honda-rules-hmc-060408.html> [accessed February 18, 2010]
- [6] *Kopacek, P.*, 2009. Automation in Sports and Entertainment. In Nof, S. Y. & SpringerLink, Springer Handbook of Automation. Berlin, Heidelberg: Springer Berlin Heidelberg.
- [7] *Byagowi, A.*, 2010. Control System for a Humanoid Robot. PhD Vienna University of Technology.
- [8] *Murphy, N.*, 2009. Serial Communication Protocols: CAN vs. SPI. [Webpage]. Available from: <http://www.netrino.com> [accessed April 6, 2010]
- [9] *CAN in Automation*, 2010. [Webpage]. Available from: <http://www.can-cia.org> [accessed January 20, 2010]
- [10] *Johansson K.H. & Torngren, M. & Nielsen, L.*, 2005. Vehicle Applications of Controller Area Network. In Hristu-Varsakelis, D. & Levine, W.S., ed. Handbook of Networked and Embedded Control Systems (Control Engineering). 1st ed. Boston: Birkhauser.
- [11] List of automation protocols. [Webpage]. http://en.wikipedia.org/wiki/List_of_automation_protocols [accessed January 14, 2010]
- [12] *Vector Informatik GmbH*. [Webpage]. Available from: <http://www.vector.com> [accessed May 27, 2010]
- [13] *Microchip*, 2002. A CAN Physical Layer Discussion, [PDF]. Available from: <http://www.microchip.com>
- [14] *Elmo Motion Control*. Whistle/Tweeter Digital Servo Drives Installation Guide [PDF]. Available from: <http://www.elmomc.com> [accessed February 18, 2010]
- [15] *Mark. R.*, 1997. Precision Planetary vs. Harmonic/Cycloidal drives. Available from: <http://www.electromate.com>
- [16] *Atlas O*, 2010. [Webpage] <http://tinyurl.com/6hs98of> [accessed September 14, 2011]
- [17] *Harmonic Drives Systems Inc.* [Webpage]. Available from: <http://www.harmonicdrive.de>
- [18] *LPKF Laser & Electronic AG*. [Webpage]. Available from: <http://www.lpkf.com> [accessed April 7, 2010]

- [19] *Farnell*. [Webpage]. Available from: <http://www.farnell.com> [accessed February 4, 2010]
- [20] *Atmel*. Datasheet for ATA6824. [PDF]. Available from: <http://www.atmel.com>
- [21] *Atmel*. Datasheet for AT90CAN32/64/128. [PDF]. Available from: <http://www.atmel.com>
- [22] *Atmel*. Datasheet for ATmega16/32/64. [PDF]. Available from: <http://www.atmel.com>
- [23] *Zumbahlen, H.*, 2008. *Linear Circuit Design Handbook*. Amsterdam: Newnes.
- [24] *Analog Devices*, [Webpage]. Available from: <http://www.analog.com>
- [25] *Hughes, A.*, 2006. *Electric Motors and Drives, Third Edition: Fundamentals, Types and Applications*. Newnes.
- [26] *Mitzner, K.*, 2009. *Complete PCB Design Using OrCAD Capture and Editor*. Newnes.
- [27] *Emadi, A.*, 2005. *Handbook of Automotive Power Electronics and Motor Drives (Electrical and Computer Engineering)*. 1 ed. CRC Press.
- [28] *Altium*, Interactive and Differential Pair Routing. [PDF]. Available from: <http://www.altium.com> [accessed January 7, 2010].
- [29] Jones, D.L., 2004. *PCB Design Tutorial (revision A)*. [PDF]. Available from: www.eevblog.com [accessed November 9, 2009]
- [30] *Elmo Motion Control*. Elmo's Training Classes [Webinar].

APPENDIX A: COMPARISON LIST

A-1 List of Common ECAD-Software

In this additional appendix, I briefly overview few different ECAD applications, which I think it is worth mentioning. ECAD-software is available from a variety of software (over 50 different) vendors. There is a wide range of free software applications, while the rest are in most cases subscription-based industry software solutions for professionals. These are generally offering up-to-date footprints and symbol library bundles, with detailed component descriptions and MCAD compatibility.

A typical ECAD-software package for designing PCBs usually contains tools for:

- ▶ *schematic,*
- ▶ *circuit simulation,*
- ▶ *layout.*

While more advanced ECAD-software packages include additional tools or support add-ons for:

- ▶ *auto routing,*
- ▶ *3D board visualization,*
- ▶ *thermal analysis,*
- ▶ *exporting MCAD data.*

Open source software (without cost) is almost as good as paid one when creating relatively simple and not so complicated PCBs. Auto routing algorithm and testing functionality may be missing when compared with commercial software. Generally, free distributions are not MCAD compatible.

Open source software:

Fritzing can be considered as an entry tool for beginners (connections are wired the same way as in physical prototype). Runs under Windows and on UNIX/Linux operating system.

gEDA is the largest and oldest open-source project that runs on UNIX/Linux operating system. A perfect tool for a quick and basic PCB designs prototyping.

KiCad is a powerful and free alternative to Eagle. Runs under Windows and on UNIX/Linux operating system.

Here is a small overview of paid ECAD programs that are commonly used in the industry.

Commercial software:

Altium Designer^{XIX} offers high levels of interoperability with MCAD data. It includes package tools for printed circuit board, FPGA^{XX} and embedded software design. It runs under Windows operating system. It is pretty powerful and expensive (1 year subscription, 4,000 €), professional use. 30-days free trial version is fully functional.

Cadence^{XXI} **Allegro PCB Designer** is an industry standard system for electronic design that manages entire PCB product lifecycle. It runs under Windows and on UNIX/Linux operating system. It is very expensive (entry level starts at 10,000 €) and complex ECAD system for professionals. The most basic version (OrCAD PCB Designer) of Cadence's Allegro suite is available with starting price at 1,500 € (1 year subscription). Limitations of the demo are in the size and complexity of the design.

Creo^{XXII} is visualization MCAD design tool solution, which requires extra modules for ECAD system. It runs under Windows operating system. It is very expensive (extra modules for drawing schematics and creating PCB layouts are required).

Eagle^{XXIII} is affordable (starts at 50 €) ECAD-software, which runs under Windows and on UNIX/Linux operating system. Has no compatibility with MCAD. Freeware version with reduced functionality is available.

PADS^{XXIV} belong to the same class of performance level as Altium and Allegro. It runs under Windows operating system. It is expensive, with fully functional 30-days free trial version available.

Zuken Cadstar^{XXV} is all-in-one solution for PCB design. It is very expensive and easily compared with Altium and Allegro. It runs under Windows operating system. Available demo exists with limitations in complexity of the design (limited to 300 pins & 50 components).

^{XIX} - Altium Designer is a registered trademark of Altium Limited

^{XX} - Acronym of Field-Programmable Gate Array, a type of logic chip that can be programmed.

^{XXI} - Cadence is a trademark of Cadence Design Systems, Inc.

^{XXII} - Creo is a trademark of Parametric Technology Corporation.

^{XXIII} - Eagle is a registered trademark of Premier Farnell, USA.

^{XXIV} - PADS is a trademark of Mentor Graphics Corporation.

^{XXV} - Cadstar is a trademark of Zuken-Redac Group Limited

Using the industry's most widely used software in most cases ensures keeping library updated (from components manufacturers) with the latest packages/components. This can dramatically speed up the design process, when using the latest components available on the market, as there is no need to create a 2D footprint and creating a 3D component models in the MCAD system.

If 2D and 3D visual component models are not found inside libraries, they may be found for free at community sites between users and suppliers like 3dcontentcentral.com.

The benefits from professional tools become really obvious when more complex and multilayer prototypes are created that requires more precision. Commercial tools are way ahead of open source software, when comparing ECAD – MCAD compatibility. As there is often a clash between file formats and compatibility issues between applications it is better to stay with one, which is compatible with the rest of the ongoing project.

A-2 Motor Comparison Chart

Side by side comparison between two types of DC motors matched to gear drive

		Brushless DC Motor Maxon-Motors EC 45	Brushed DC Motors Faulhaber 3257 G (024 CR)	
<i>Parameter</i>	<i>Symbol</i>	<i>Value</i>	<i>Value</i>	<i>Unit</i>
Supply voltage	U_N	24	24	V
No-load speed, (up to)	$n_O, (n_{e \max.})$	6,800, (5260)	5,900, (5000)	rpm
No-load current	I_O	0.201	0.129	A
Torque	T_{out}	84.3	70	mNm
Maximal current	$I_{max.}$	2.36	2.3	A
Efficiency	$\eta_{max.}$	83	83	%
Output power	$P_{max.}$	50	83.2	W
Stall torque	T_H	822	547	mNm
Rotor inertia	J	135	41	gcm ²
Rotor inductance	L	573	270	μ H
Rotor resistance	R	0,978	1.63	Ω
Speed constant	k_n	285	253	rpm/V
BEMF constant	k_b	/	3.95	mV/rpm
Torque constant	k_t	33.5	37.7	mNm/A
Current constant	k_I	/	0.027	A/mNm

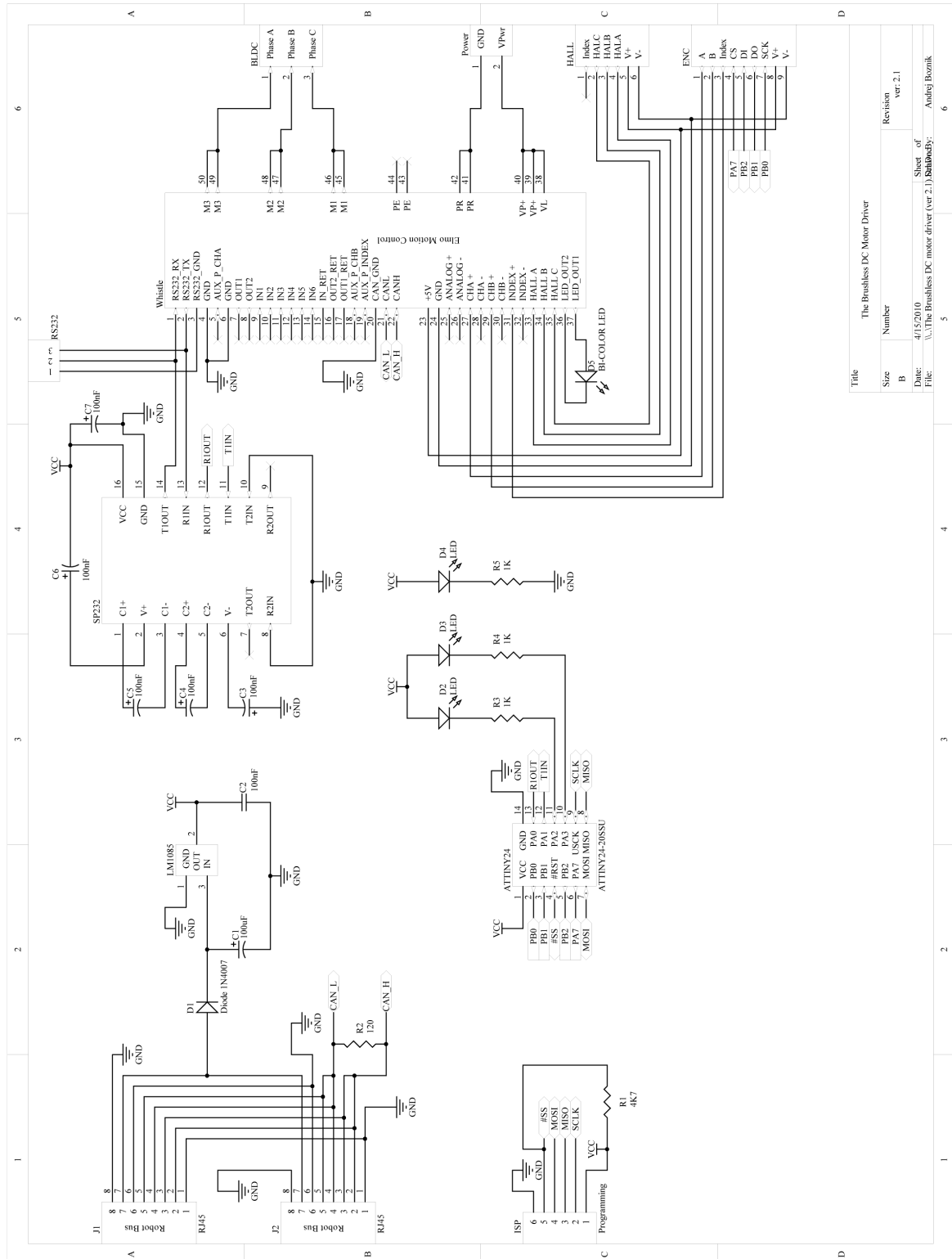
		Harmonic Gear Drive Harmonic Drives Systems Inc. 20-160-874405-6	Planetary Gear Drive Faulhaber 38/1	
<i>Parameter</i>	<i>Symbol</i>	<i>Value</i>	<i>Value</i>	<i>Unit</i>
Reduction ratio (nominal)		160 : 1	415 : 1	
Efficiency	$\eta_{PG \max.}$	77	55	%
Speed (gear output)	$n_{out \max.}^*$	42.5 rpm	12	rpm
Output torque	$T_{out+gear}$	13	10	Nm

*Note: $n_{out \max.} = (n_{e \max.}) / (\text{Reduction ratio})$

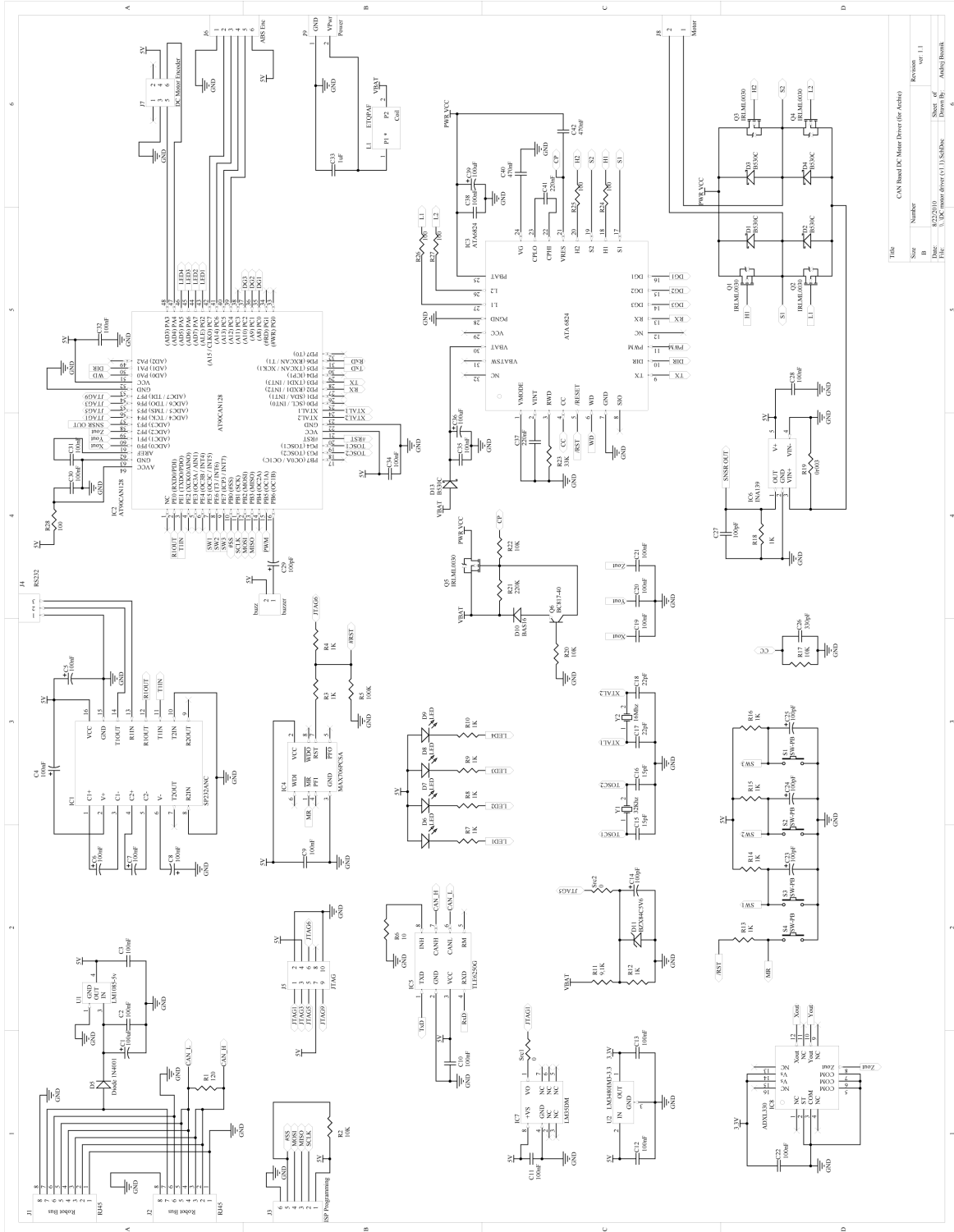
		Encoder Austrian Micro Systems	Encoder Faulhaber	
<i>Parameter</i>	<i>Symbol</i>	<i>Value</i>	<i>Value</i>	<i>Unit</i>
Lines per revolution (motor output)	N	360	2048	
Lines per revolution (gear output)		360 x 160	2048 x 415	
Supply voltage (DC)	V_{CC}	4.5 – 5.5	3.6 – 5.5	V
Current consumption	$I_{CC \max.}$	15	6	mA

APPENDIX B: CIRCUIT SCHEMATICS

B-1 Circuit schematic for the brushless DC motor driver



B-2 Circuit schematic for the brushed DC motor driver



Title			
Size	Number	Revision	Rev. 1.1
ID	4323016	Sheet of	6
File	3. DC motor driver (V1) Schematic	Drawn By	Andy Horvath

APPENDIX C: BILL OF MATERIALS

C-1 Bill of materials for the brushless DC motor driver

Designator	LibRef	Package	Description	Value	Quantity
Whistle	Elmo Whistle		ELMO Whistle (Controller)		1
ATTINY24	ATtiny24	D014_N	ATTINY24 (Microcontroller)		1
SP232	SP232ANC	WSO16_M	SP232ANC (RS-232 Line Driver)		1
C1	Cap Pol1	CAPPR5-5x5	Polarized Capacitor (Radial)	100uF	1
C2	Cap Pol3	C1206	Capacitor (Semiconductor SIM)	100nF	1
C3, C4, C5, C6, C7	Cap Pol3	C0805	Polarized Capacitor (Surface Mount)	100nF	5
D1	Diode 1N4007	DO41	1N4007		1
D2, D3, D4	LED	DSO-F2/D6.1	LED		3
D5	BI-COLOR LED	DSO-F2/D6.1	BI-COLOR LED		1
R1	Res3	RESC1608L	Resistor	4K7	1
R2	Res3	RESC1608L	Resistor	120	1
R3, R4, R5	Res3	RESC1608L	Resistor	1K	3
LM1085	LM1085-5V	TS3B_M	LM 1085 (Voltage Regulator)		1
Power	Power	Power, 2-Pin	Power Supply Connector		1
BLDC	BLDC	BLDCCon, 3-Pin	Motor Power Connector		1
HALL	Hall	HallCon, 6-Pin	Hall Sensor Connector		1
ENC	EncoderCon	EncCon, 9-Pin	Encoder Connector		1
ISP	Programming	HDR1X6, 6-Pin	ISP Connector		1
RS232	RS232	HDR1X3, 3-Pin	RS-232 Connector		1
J1, J2	RJ45	RJ45 JACK	CAN Connector		2
					29

C-2 Bill of materials for the brushed DC motor driver

Designator	LibRef	Package	Description	Value	Quantity
IC1	SP232ANC	W SO16_M	SP232ANC (RS-232 Line Driver)		1
IC2	ATCAN	64A_N	AT90CAN128 Microcontroller		1
IC3	ATA6824	32A_M	ATA6824 (H-bridge Driver)		1
IC4	MAX706PCSA	NSO8_N	MAX706PCSA (IC supervisory circuit)		1
IC5	TLE6250G	8S1_L	TLE6250G (CAN-transceiver)		1
IC6	INA139	SOT23-5AN	INA139 (Current Shunt Monitor)		1
IC7	LM35DM	M08A_N	LM35DM (Temperature Sensor)		1
IC8	ADXL330	LFCSP_LQ	ADXL330 (Accelerometer)		1
C1	Cap Pol1	CAPPR5-5x5	Polarized Capacitor (Radial)	100uF	1
C2, C3	Cap Semi	C1206	Capacitor (Semiconductor SIM)	100nF	2
C4 - C8	Cap Pol3	C0805	Polarized Capacitor (Surface Mount)	100nF	5
C9 - C13, C19 - C22, C28, C30 - C32, C34, C35, C38	Cap Semi	CAPC2512L	Capacitor (Semiconductor SIM)	100nF	16
C14, C27	Cap Pol3	CAPC2012L	Polarized Capacitor (Surface Mount)	100pF	2
C15, C16	Cap Semi	CAPC2512L	Capacitor (Semiconductor SIM)	15pF	2
C17, C18	Cap Semi	CAPC2512L	Capacitor (Semiconductor SIM)	22pF	2
C23-C25, C29,	Cap Pol3	CAPC2012L	Polarized Capacitor (Surface Mount)	100pF	4
C26	Cap Semi	CAPC2512L	Capacitor (Semiconductor SIM)	330pF	1
C33	Cap Semi	CAPC3225L	Capacitor (Semiconductor SIM)	1uF	1
C36, C39	T491C	C	Capacitor (Surface Mount)	100uF	2
C37, C41	Cap Semi	CAPC2512L	Capacitor (Semiconductor SIM)	220nF	2

C-2 Bill of materials for the brushed DC motor driver - continued

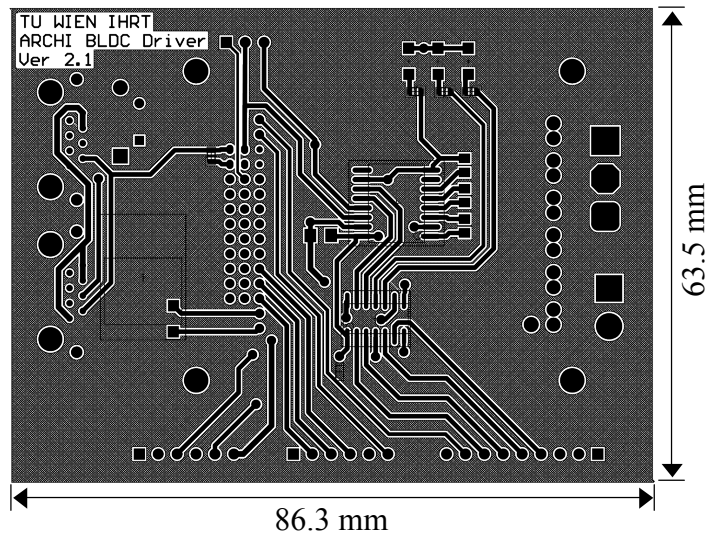
Designator	LibRef	Package	Description	Value	Quantity
C40, C42	Cap Semi	CAPC3225L	Capacitor (Semiconductor SIM)	470nF	2
D1 - D4, D13	DiodesB530C	DiodesB530C	B530C (Schottky Rectifier)		5
D5	Diode 1N4001	DIO10.46- 5.3x2.8	1N4001 (Rectifier)		1
D6 – D9	LED2	DSO-F2/D6.1	LED		4
D10	BAS16	SOT23_N	BAS16 (Switching Diode)		1
D11	D Zener	DIO7.1- 3.9x1.9	BZX84C5V6 (Zener Diode)		1
R1 – R18, R20 - R28	Res3	RESC1608L	Resistor		27
R19	Res3	PCBComponent_1	Shunt Resistor	3mΩ	1
L1	ETQPAF Coil	ETQPAF Coil			1
Src1, Src2	Res3	RESC1608L	uC ADC Input Selector		2
S1 – S4	SW-PB	SPST-2	Switch		4
U1	LM1085-5v	TS3B_M	LM 1085 (Voltage Regulator)		1
U2	LM3480IM3	MF03A_N	LM 3480 (Voltage Regulator)		1
Y1	XTAL	BCY- W 2/D3.1	32kHz Crystal Oscillator		1
Y2	XTAL	BCY- W 2/D3.1	16 MHz Crystal Oscillator		1
Q1 – Q5	IRLML0030	SOT-23	IRLML0030 (N-Channel Power MOSFET)		5
Q6	BC817-40	SOT23_N	BC817-40 (NPN Transistor)		1
buzz	Header 2	RAD-0.1	Buzzer		1
J1, J2	RJ45	RJ45 JACK	CAN Connector		2
J3	Programming	HDR1X6, 6-Pin	ISP Connector		1
J4	RS232	HDR1X3, 3-Pin	RS-232 Connector		1
J5	JTAG	HDR2X5, 10-Pin	JTAG Connector		1
J6	ABS Enc	HDR1X6, 6-Pin	ABS Position Encoder		1
J7	DCMotor Enc	HDR3X2, 6-Pin	Encoder Connector		1
J8	Motor	Power, 2-Pin	Motor Power Connector		1
J9	Power	Power, 2-Pin	Power Supply Connector		1
					117

APPENDIX D: THE PCB LAYOUT DRAWINGS

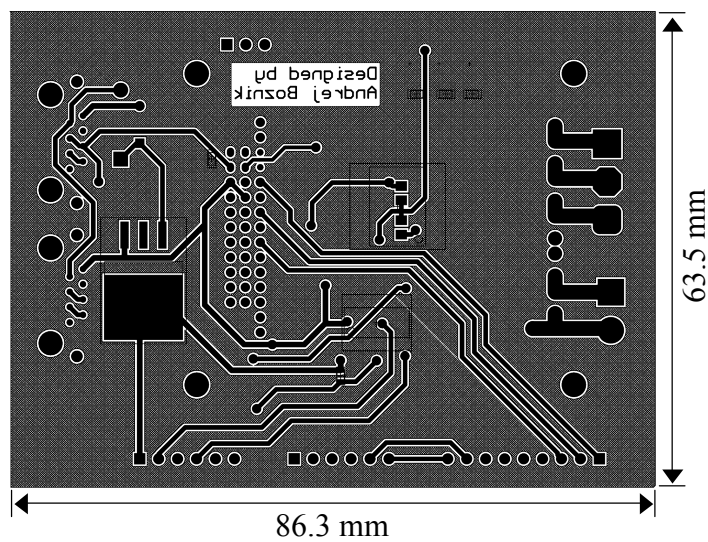
The following contain the layout required for the PCB fabrication process.

D-1 Layout drawing of the brushless DC motor driver

Top layer of the brushless DC motor driver

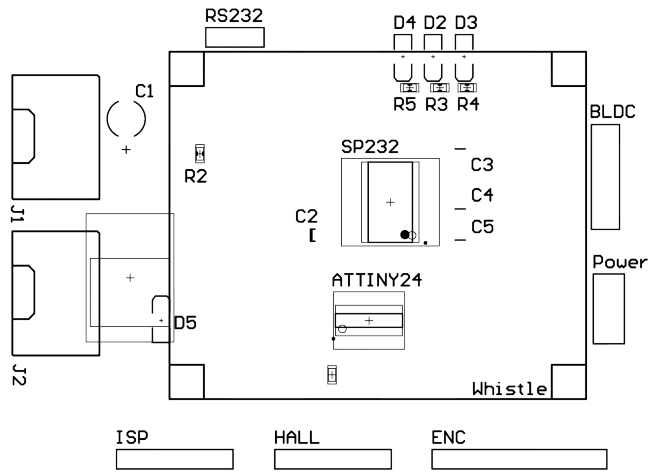


Bottom layer of the brushless DC motor driver

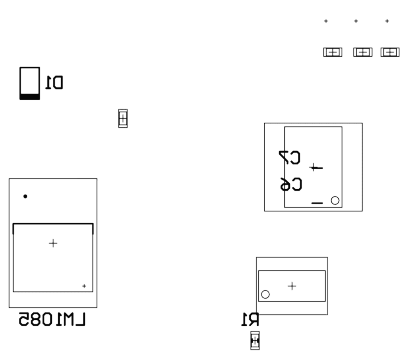


D-1 Layout drawing of the brushless DC motor driver - continued

Top silkscreen layer of the brushless DC motor driver

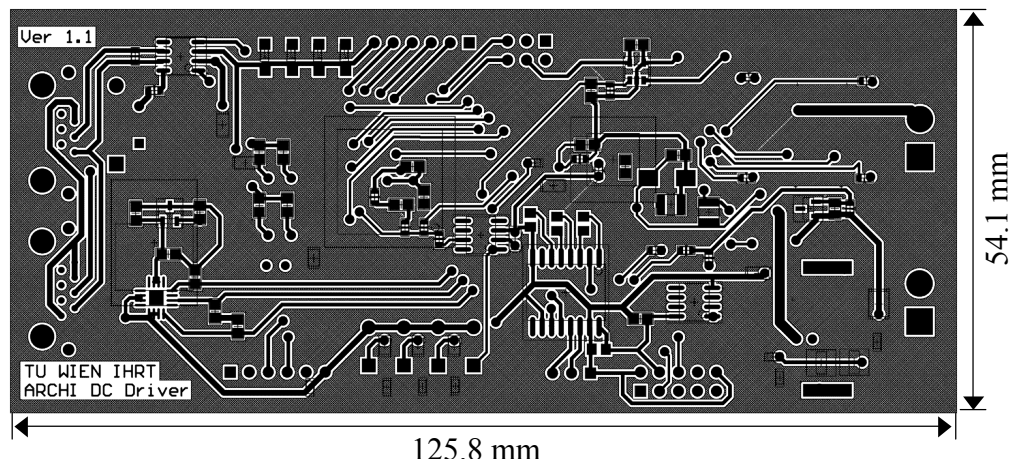


Bottom silkscreen layer of the brushless DC motor driver

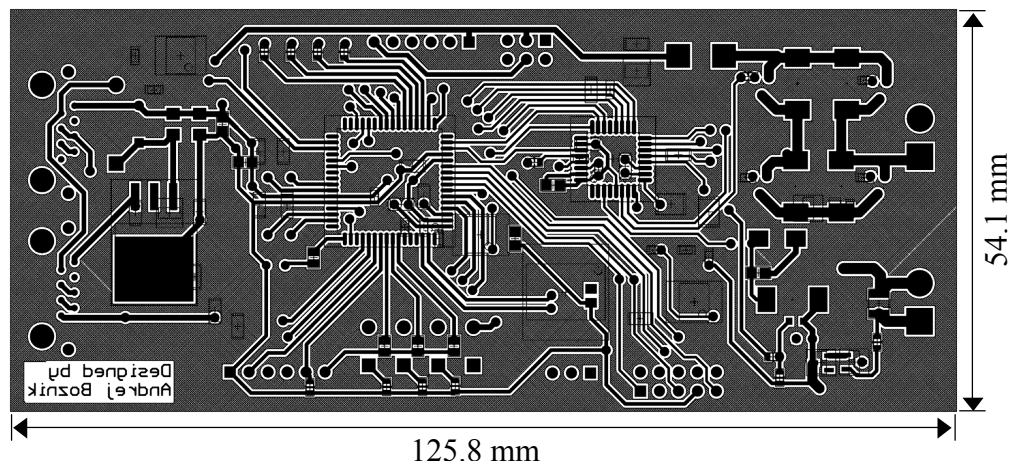


D-2 Layout drawing of the brushed DC motor driver

Top layer of the brushed DC motor driver

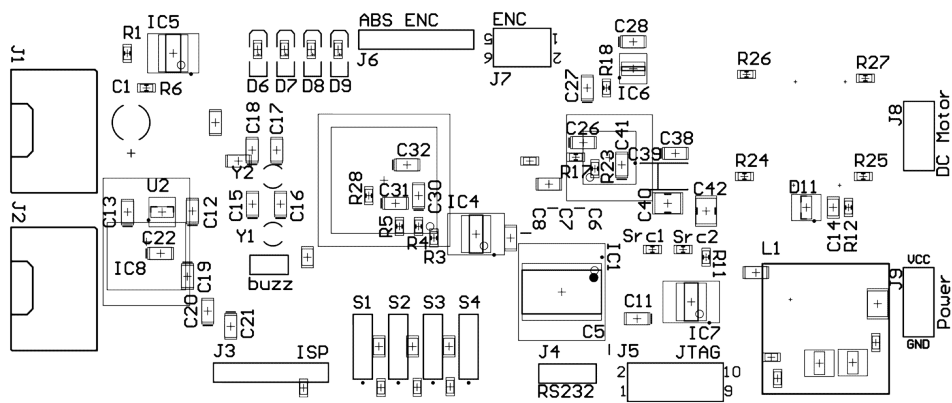


Bottom layer of the brushed DC motor driver

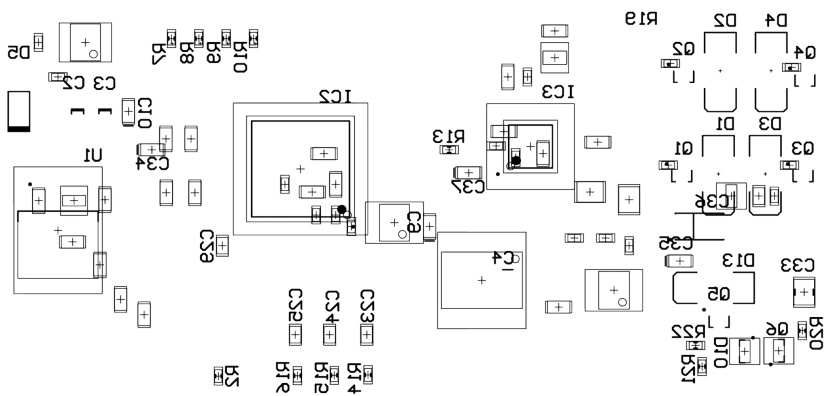


D-2 Layout drawing of the brushed DC motor driver - continued

Top silkscreen layer of the brushed DC motor driver



Bottom silkscreen layer of the brushed DC motor driver





Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

IZJAVA O USTREZNOSTI DIPLOMSKEGA DELA

Podpisani mentor doc. dr. Miran Rodič izjavljam, da je
(ime in priimek mentorja)
študent Andrej Božnik izdelal diplomsko
(ime in priimek študenta-tke)

delo z naslovom: Elektronsko vezje za pogon motorjev humanoidnega robota
z uporabo CAN Bus
(naslov diplomskega dela)

v skladu z odobreno temo diplomskega dela, Navodili o pripravi diplomskega dela in
mojimi navodili.

Datum in kraj:

07.12.2011, Maribor

Podpis mentorja:

Miran Rodič



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

IZJAVA O USTREZNOSTI DIPLOMSKEGA DELA

Podpisani mentor izr. prof. dr. Karl Gotlih izjavljam, da je
(ime in priimek mentorja)
študent Andrej Božnik izdelal diplomsko
(ime in priimek študenta-tke)

delo z naslovom: Elektronsko vezje za pogon motorjev humanoidnega robota

z uporabo CAN Bus

(naslov diplomskega dela)

v skladu z odobreno temo diplomskega dela, Navodili o pripravi diplomskega dela in
mojimi navodili,

Datum in kraj:

07.12.2011, Maribor

Podpis mentorja:

UNIVERZA V MARIBORU

Fakulteta za elektrotehniko, računalništvo in informatiko
(ime fakultete)

IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUČNEGA DELA IN
OBJAVI OSEBNIH PODATKOV AVTORJA

Ime in priimek avtorja (avtorice): Andrej Božnik
Vpisna številka: 93575550
Študijski program: UNI ELEKTROTEHNIKA - MEHATRONIKA
Naslov zaključnega dela: Elektronsko vezje za pogon motorjev humanoidnega
robota z uporabo CAN Bus

Mentor: doc. dr. Miran Rodič
Somentor: Izr. prof. dr. Karl Gotlih

Podpisani-a Andrej Božnik izjavljam, da sem za potrebe arhiviranja oddal-a elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah (Ur. l. RS, št. 16/2007) dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna elektronski verziji, ki sem jo oddal-a za objavo v Digitalno knjižnico Univerze v Mariboru. Podpisani-a izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zagovora, naslov zaključnega dela) na spletnih straneh in v publikacijah UM.

Kraj in datum: 07.12.2011, Maribor

Podpis avtorja (avtorice): Am B5



Univerza v Mariboru

*Fakulteta za elektrotehniko,
računalništvo in informatiko*

IZJAVA O AVTORSTVU diplomskega dela

Spodaj podpisani/-a Andrej Božnik,

z vpisno številko 93575550,

sem avtor/-ica diplomskega dela z naslovom:

Elektronsko vezje za pogon motorjev humanoidnega robota z uporabo CAN Bus

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)
doc. dr. Miran Rodič
- in somentorstvom (naziv, ime in priimek)
izr. prof. dr. Karl Gotlih
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v DKUM.

V Mariboru, dne 07.12.2011

Podpis avtorja/-ice: 