İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

ELECTRONIC CONTROL UNIT AND POWER ELECTRONICS DESIGN FOR AN EIGHT LEGGED MOBILE ROBOT

Master Thesis by

R. Barkan UĞURLU, Electrical Eng.

Department : INTERDISCIPLINARY

Programme: MECHATRONICS ENGINEERING

APRIL 2006

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

ELECTRONIC CONTROL UNIT AND POWER ELECTRONICS DESIGN FOR AN EIGHT LEGGED MOBILE ROBOT

Master Thesis by

R. Barkan UĞURLU, Electrical Eng.

518041017

Date of submission : 21 March 2006

Date of defence examination: 03 April 2006

Supervisor (Chairman): Assoc. Prof. Dr. Hakan TEMELTAŞ

Members of the Examining Committee Assoc. Prof.Dr. Metin GÖKAŞAN

Assoc. Prof.Dr. Ata MUĞAN

APRIL 2006

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SEKİZ BACAKLI BİR MOBİL ROBOT İÇİN ELEKKTRONİK KONTROL ÜNİTELERİ VE GÜÇ ELEKTRONİĞİ DEVRELERİ TASARIMI

Master Tezi

R. Barkan UĞURLU, Elektrik Müh.

518041017

Tezin Enstitüye Verildiği Tarih: 21 Mart 2006

Tezin Savunulduğu Tarih: 03 Nisan 2006

Tez Danışmanı: Doç. Dr. Hakan TEMELTAŞ

Diğer Jüri Üyeleri: Doç. Dr. Metin GÖKAŞAN

Doç. Dr. Ata MUĞAN

NİSAN 2006

ACKNOWLEDGEMENT

I would like to thank my supervisor Dr. Hakan TEMELTAŞ for encouraging me to work on this subject and for his fulltime support. I am also enourmously grateful for the contributions of Dr. Metin GÖKAŞAN, Bahadır ATILGAN, Hüseyin KIRMAN, C. Mete UZUNDERE and B. Çağdaş ERALTAN.

March 2006 R.Barkan UĞURLU

INDEX

ABBREVIATIONS

DC	: Direct Current
AC	: Alternative Current
DSP	: Digital Signal Processor
MCU	: Microcontroller Unit
PMBLDC	: Permanent Magnet Brushless DC Motor
PWM	: Pulse Width Modulation
SWPWM	: Sine Weighted Pulse Width Modulation
PID	: Proportional Integral Derivative
CAN	: Controller Area Network
RAM	: Random Access Memory
ROM	: Read Only Memory
EEPROM	: Electrically Erasable Read Only Memory
REF	: Reference
IC	: Integrated Circuit
DOF	: Degrees of Freedom
PCB	: Printed Circuit Board

TABLE LIST

FIGURE LIST

SYMBOL LIST

$\mathbf{T_1}$	**:** Promotion Period
$\mathbf{T_2}$	**:** Remotion Period
d	**:** Dominant bit, logic 0
r	**:** Recessige bit, logic 1
$\mathbf{V_{CC}}$	**:** Collector feed voltage
$\mathbf{V_{CE}}$	**:** Voltage drop between collector and emitter pins
$\mathbf{I_B}$	**:** Base current
$\mathbf{I_C}$	**:** Collector current
$\mathbf{V_{BE}}$	**:** Voltage drop between base and emitter pins
$\mathbf{V_{GS}}$	**:** Voltage drop between gate and source pins
$\mathbf{\beta_{DC}}$	**:** DC Gain in a transistor
$\mathbf{\beta}$	**:** Angle between stator electrical field and rotor magnetic field
$\mathbf{\theta}$	**:** Angular Position
T	: Torque
p	**:** Number of poles in a motor
φ	**:** Phase difference

ABSTRACT

ELECTRONIC CONTROL UNIT AND POWER ELECTRONICS DESIGN FOR AN EIGHT LEGGED MOBILE ROBOT

In this study, all the electronic circuits and and control units of an eight-legged mobile robot are obtained. Having created a mechanical frame, the robot has gained actuation which is controlled in different levels. Different control levels are connected to each other via CAN Bus which is a highly efficient serial communication network.Therefore, the robot is able to move around and complete tasks while the many kinds of control software is implemented to controllers.

Firstly, a living scorpion's walking sequences are examined from a biomimetric point of view. After that, CAN Bus is introduced as all the control hierarchy based on this communication network. Then mechanical design and hardware realization is fulfilled. Finally, a motor control scheme is obtained and controller test algorithms are built.

Keywords: Power Electronics, Legged Mechanism, Mobile Robots, Microprocessors, CAN Communication Protocol

Science Code : 608

ÖZET

SEKİZ BACAKLI BİR MOBİL ROBOT İÇİN ELEKTRONİK KONTROL ÜNİTELERİ VE GÜÇ ELEKTRONİĞİ DEVRELERİ TASARIMI

Bu tez çalışmasında sekiz bacaklı yürüyen bir mobil robot için gerekli olan tüm elektronik devrelerin tasarlanması amaçlanmıştır. Hali hazırda üretilmiş olan mekanik altyapının üzerine bu elektronik sistemler eklenerek robotun hareket kabiliyeti kazanması hedef olarak belirlenmiştir. Ayrıca robotun her hareketi ise çeşitli seviyelerde belirlenmiş kontrol üniteleri ile denetlenecektir. Bu farklı seviyelerde belirlenmiş olan kontrol üniteleri birbirlerine CAN ağı üzerinden bağlanmışlardır. CAN ağı, benzer uygulamalarda sıkça kullanılan verimli bir seri haberleşme protokolüdür. Böylelikle kontrol ünitelerine yüklenecek birçok farklı kontrol algoritması ile robotun istenilen görevleri başarıyla yerine getirmesi beklenmektedir.

Tez çalışmasında öncelikle yaşayan canlı bir akrebin yürüyüş biçimi biomimetrik bir açıdan analiz edilmiştir. Ardından, kontrol üniteleri arasındaki hiyerarşiyi oluşturan CAN ağının tanıtımı yapılmış; CAN üzerinden data alışverişi açıklanmıştır. Daha sonra ise mekanik ve elektriksel tasarımın gerçeklendiği dizayn kısmı gösterilmektedir. Son olarak ise motor kontrolü için önerilen kontrol yapısı ve kontrol üniteleri için gerekli olan test algoritmaları sunulmuştur.

Anahtar Kelimeler: Güç Elektroniği, Bacaklı Mekanizmalar, Mobil Robotlar, Mikroişlemciler, CAN Haberleşme Protokolü

Bilim Kodu : 608

1. INTRODUCTION

Arthropoda are one of the biggest phylums, which include numerous invertebrate organisms such as insects, spiders and cruscateans. Organisms in this phylum are characterized by a hard, segmented external covering and segmented limbs which strenghten them against negative circumstances as they are able to survive.

The field of robotics has witnessed several improvements in recent years. Generally, the most notable ones are based on biologically inspirations, widely known as biomimetry. Examples consists of many creatures on earth, from insects to human. Especially insect-like multi-segmented systems tend towards high speed locomotion even on the rough terrain. This kind of robots have fulfilled high speed and robustness over quite bad conditioned terrain.

If we look at the Robotic History in recent years, some milestone works can be seen. Following are displaying some samples :

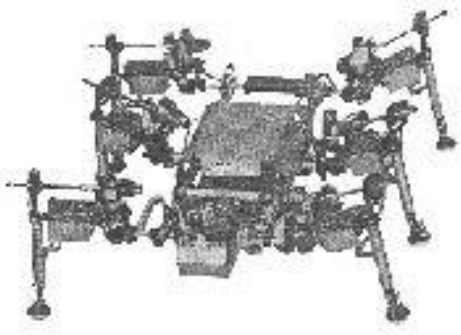

Figure 1.1 : Atilla II

In figure 1.1 Atilla II can be seen. It is a six-legged robot, firmly called hexapod. It is considered as a highly improved legged mechanism that appears in robotics up to the date, 1989

Figure 1.2 : ASIMO P3

In the above figure, figure 1.2, Honda's ASIMO is shown. Honda created P3 in 1997, the second major step in creating their ASIMO. P3 is the first completely independent bipedal humanoid walking robot.

Figure 1.3 : Sony QR Humanoid

In figure 1.3 Sony's humanoid robot is displayed. In 2001, Sony unveils humanoid robots, the Sony Dream Robots at Robodex. SDR is able to recognize 10 different faces, expresses emotion through speech and body language.

Figure 1.4 : NeCorO the Cat, OMRON

In figure 1.4, OMRON's robot cat NeCoRo can be seen. Omron released this artificial cat NeCoRo in 2002. Besides being a sophisticated robot with an artificial intelligence system, it also immitates the looking and the gestures of a real cat.

In the literature, there are huge amount of crawler robot applications such as Xiao's crawler robot[7]. However they are not able to overcome obstacles or any roughness over the trajectort. Because of that disadvantage, legged mechanisms are more capable in exploration.

As our system has many features from different disciplines, a proper literature search has been fulfilled. For motor drive applications, [1], [2] and [5] references are used. In order to struct a healty CAN network, Mitsubishi Electric's CAN manual[4] is examined. While designing an eight-legged mechanism, other realized eight legged robots, [6],[8],[9],[10],[12] and [13] are referred to our design.

In our study, we choose to design an eight-legged mobile robot which has two segments. By doing it we aimed to gain flexibility on our robots. Having configured such mechanical frame, it is our purpose that this mobile robot may be used in exploration. For instance, in pipelines there might be leakages or little cracks. Reconfiguring the sensor data on the robot, it is able find the leakage coordinates or crack.

It might also be used in some specific areas in which human access is impossible such as places after disasters. For instance, after a powerful earthquake, our robot may be used for exploration in order to receive information from the places where human cannot get in.

The main reason that movements are realized by legs is its specific adventages over wheels. As mentioned above, legged mechanisms are able to move over rough terrain. Wheeled mechanisms are not appropriate for this kind of territorries. Legged walking is also most encountered walking format in nature. Moreover legs are less likely sink into the ground. However in wheeled mechanism, soil dynamics must be included in the model. One of the most important feature of legged mechanisms is that they can give a smooth ride over a rough ground by varying the effective lenghts of their legs.

Figure 1.5: 2-Segmental robot, in Twisted Mode

Our main purpose is to design a legged mechanism that has a flexibile mobility as seen in above figure. So;

- It may be used to seek leakages or damaged areas in pipelines
- It may be used as an explorer robot in case of disasters.

Considering these facts, the robot design is realized with respect to respond well.

In our thesis, the first chapter is emphasizing the importance of this work and expresses our motivation on the subject. In second chapter, a walking planning for an eight-legged machine is analyzed. Based on a biomimetric approach, a living scorpion's gait sequences are expressed mathematically. Third chapter focuses on CAN Network and Data Transfer. As we apply two disticnt CAN Networks, it gains importance in our system. In fourth chapter, mechanical design is shown and our proposed electrical design is introduced. Having a hierarchical control units being connected via CAN, our system appears to be a physical basis for complicated control algorithms. Fifht chapter is including our proposed low level motor control scheme and control unit test algorithms. Finally, the conclusion takes place.

2. WALKING PLANNING

Being an eight-legged autonomous walking mechanism, our robot is main task is move around the desired trajectory independently. Thus; before extracting inverse kinematic solutions, walking sequences must be planned by considering the whole multi-legged mechanical system.

In our robot, each leg has 3 active joints. Therefore, all 24 joints of all the legs in our mechanism combined with the pose of the robot itself are considered. In order to provide a robust motion to our robot, each of these 8 legs' movements are extracted by considering the synchronization and compability between them.

Biomimetric approach is applied in many ways to our robot, from constructing the mechanical structure to communication networks. Moreover, the walking planning is also based on real 8-legged creatures, especially scorpions. Leg placements in our robot is quite similar to scorpions. So that, we can easily implement a living scorpion's walking patterns into our system.

Considering the whole mechanical frame, walking planning is based on following three assumptions:

- The robot includes 2 segments and 8 homogenouslegs. Each leg has 3 DOF.
- Lateral and longitudinal symmetry of the robot must be kept.
- The passive joint between the segments is clenched.

2.1. Leg Structure

In figure 2.1, a detailed leg structure is displayed. As seen above, it has 3 degrees of freedom which makes it freely take position in 3 dimensions.

The link which is firmly attached to the boy houses the thoracic joint. Thoracic joint actuates the whole leg on the direction of forward and backward. This joint rotates a hip area of the leg which consists of basalar joint. Basalar joint elevates the thigh area of the lef on the direction. Stance or swing period of a leg is adjusted by the

basalar joint. That joint also rotates distal joint. In our system distal joint also includes a worm gear which transforms rotational movements into translational movements. Distal joint moves the leg's last segment on both lateral directions.

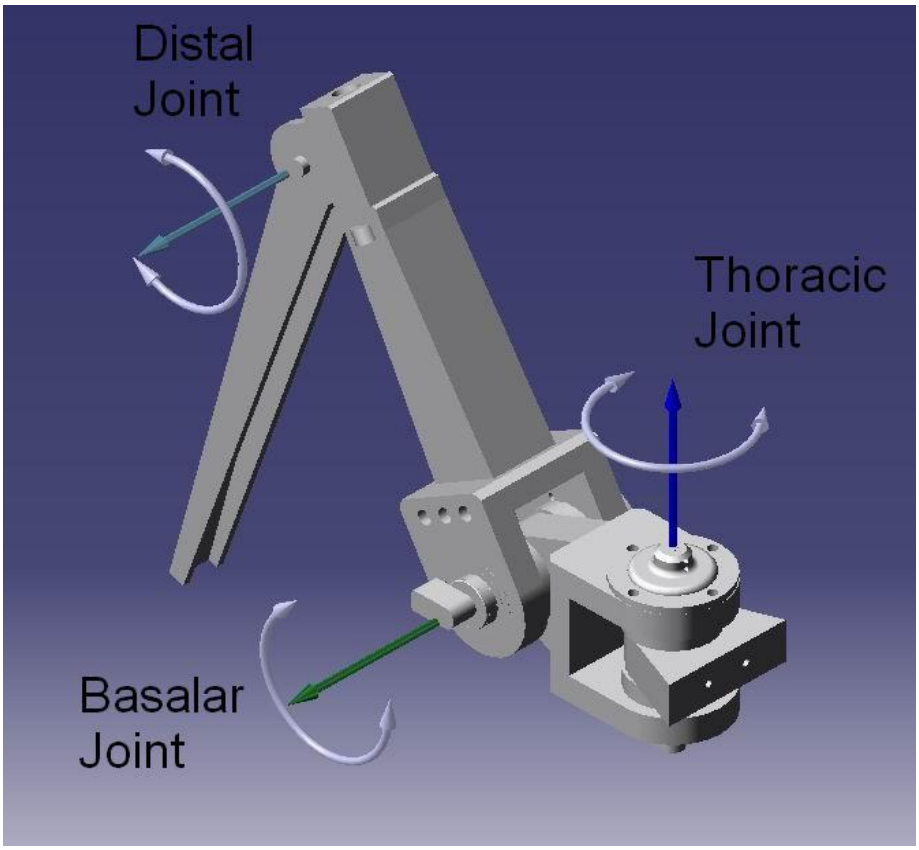

Figure 2.1 : Leg Structure

Considering the mechanism explained above, we can have a smooth ride over the desired trajectory. By varying these legs' segments in different dimensions any locomotion method can be applied to the system.

2.2. Motion Patterns

The gait sequence of living scorpion is nearly an alternating tetrapod scheme (Fig. 2.2). In a tetrapod, the walking sequence is triggered by posterior leg. Then this step is followed anteriorly by others on the directed order. All legs are following each other sequantally and each leg lagging the proceeding one by ten percent of the total cycle. In other saying, if we express a leg's movement with a sinusoidal function, the phase is 0.1 times of the full period.

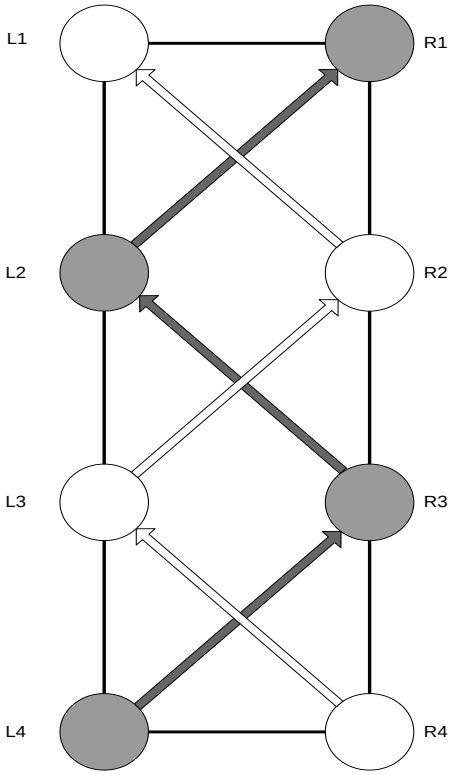

Figure 2.2 : Tetrapod Walking Gait

The tetrapod commences the wave of promotions when the first one is half way through its cycle. A cycle includes a leg's elevations in terms of swing and stance. In swing mode, the leg is lifted up which is called remotion. In stance mode, the leg is touching the groud which is called promotion. Phase is determining the time difference between the legs which are ordered anteriorly. Hence, phase, swing and stance are the parameters of the robot's motion pattern.

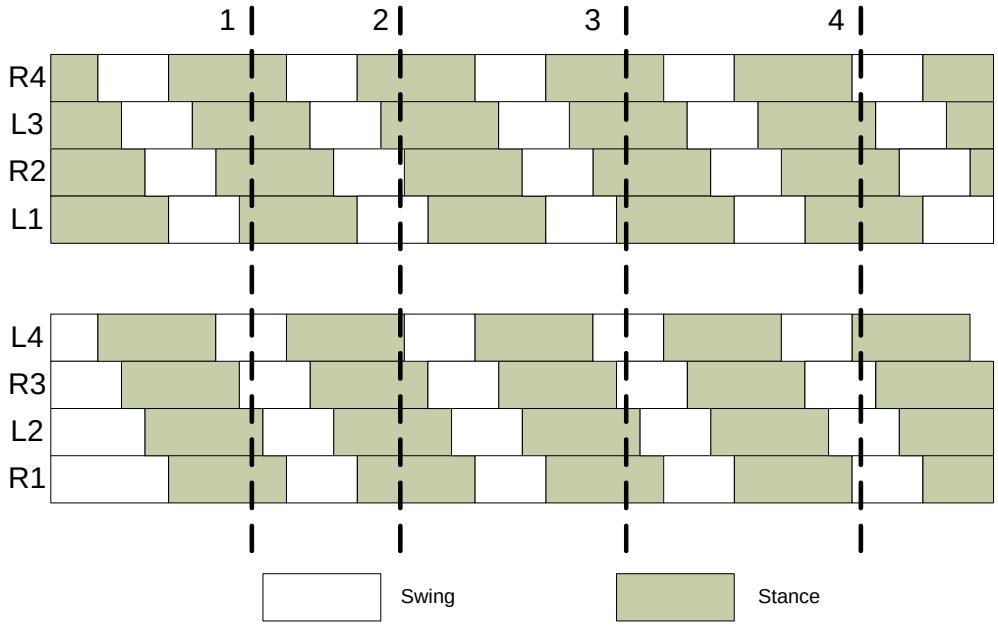

Figure 2.3 : Motion Patterns

In Figure 2.3., a motion pattern may be seen from the right start position. Vertical lines gives an idea about how many legs are stancing and how manya re swinging. Horizontal lines are revealing the time that passes through.

Bowerman[11] reports a linear change of the promotion period T_1, as well as the remotion period T_2, with the stride frequency. The slope is about 0.2 in case of promotions and about 0.8 in case of remotions. This relation allows us to express one parameter by the other. Swing and stance periods are coupled and cannot change independently. They both appear to be parameters of stride frequency.

$$f(t)=\begin{cases} A*\sin(\pi*(t-T_1)/T_2+\pi/2), 0\le MOD(t,T1+T2)\pi T1 \\ A*\sin(\pi/T_1-\pi/2), 0\le MOD(t,T1+T2)\pi T1+T2 \end{cases} \quad \textbf{(2.1)}$$

According to biomimetric approach the patterns may be transferred directly to our robot. The function f(t) represents the leg L4's movements while walking. As expressed above, it is a waveform that has 2 different sine wave. In biomimetric analysis the remotion period measured to be smaller then promotion period, so that, the body always kept in balance.

Walking starts from the posterior leg, and it is followed in anterior order. Therefore other leg's movements can be expressed by the same function if only we add the parameter phase. After a portion of a cycle, other legs function like g(t) :

$$g(t)=\begin{cases} A*\sin(\pi*t/T_1-\pi/2+\varphi), \quad 0\le MOD(t,T1+T2)\pi T1 \\ A*\sin(\pi*(t-T_1)/T2+\pi/2+\varphi), 0\le MOD(t,T1+T2)\pi T1+T2 \end{cases} \quad \textbf{(2.2)}$$

For any given leg movement, some of the sequential sinusoidal waveforms are shown below in Figure 2.4. Each of them follows each other within a small time difference, phase which is symbolized with φ.

Promotion period, T_1, is always bigger than remotion period T_2, as biomimetric approach justifies. In our motion patterns the rati of T_1 / T_2 is thought as 5 /3.

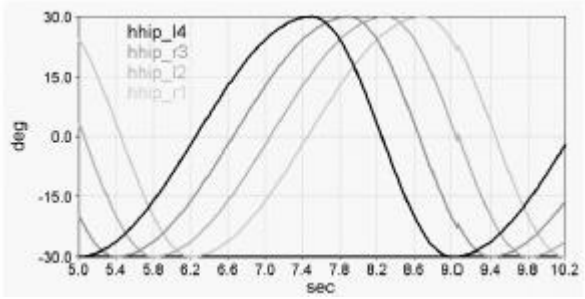

Figure 2.4 : Gait Sinusoids

To avoid disturbing, all three joints of a leg should only contribute to the forward motion of the body and not to an up/down or left/right. During each stance, the lateral distance between body and foot and height of the body above the ground should remain constant as living scorpions instictly follow this order. Therefore there are two angles for a basalar and a distal joint should be expressed as functions of the thoracic joint angle.

2.3. Biomimetric Comparison of Motion Patterns

Figure 2.5 : Scorpion Walking (2)

In Figure 2.3., motion patterns can be seen in terms of promotion and remotion periods. Also vertical lines are giving us an idea about leg's positions.

Considering the second vertical line[2]; L2, L3, L4, R4 and R1 is on the ground. L1 is in swing mode. Figure 2.5. is proving this given fact.

Figure 2.6 : Scorpion Walking (3)

If we look at third vertical line[3] in Figure 2.3., L1, L2, L3 and R4 is touching the groung. L4 is for example, swinging. Figure 2.6. is also proving this fact too.

Many momentary situations can be captured from scorpion walking. They all fit the motion pattern that is given in Figure 2.3.

3. DATA TRANSFER VIA CONTROLLER AREA NETWORK

The Controller Area Network, in short form CAN, is a serial communications protocol that efficiently supports distributed real-time control with a very high level of data integrity.

Generally CAN Bus is used in automative applications. However it is not restircted only in automative industry. CAN fulfils the communication needs of a wide range of applications, from high speed networks to low-cost multiplex wiring.

The intention of the CAN specification is to achive compability between any two CAN implementations. Compability, however, has different aspects with respect to, for example, electrical features and the interpreation of data to be transferred.

To achive design transparency and application flexibilty CAN has been divided into three layers

- The Object Layer
- The Transfer Layer
- The Physical Layer

The Object Layer and Transfer Layer comprise all services and functions of the data link layer defined by ISO/OSI model.

The scope of the Object Layer includes determining which messages are to be transmitted, deciding which messages received by the Transfer Layer are actually to be used and providing and interface to the Application Layer related hardware. There is considerable freedom in defining object handling.

The Transfer Layer principally concerned with the transfer protocol, for example controlling the framing, performing arbitration, error checking, error signalling and fault confinement. Within the Transfer Layer, it is decided whether the bus is free for starting a new transmission or whether reception of a message is just starting. Also,

some general features of the bit-timing are regarded as part of the Transfer Layer. Modifications to the Transfer Layer cannot be made.

The Physical Layer covers the actual transfer of the bits between the different nodes, with respect to all electrical properties. Within a network the physical layer has to be the same for all nodes. However, there are many possible implementations of the Physical Layer.

3.1. Data Transfer Process

Using the CAN Bus in our system our main purpose is to perform healthy data transfer between nodes.

A node originating a message is called the *transmitter* of that message. The node continues to be transmitter until the bus is idle or the node loses arbitration. A node is called the *receiver* of a message if it is not the transmitter of that message and the bus is not idle.

Message transfer is manifested and controlled by four different frame types :

- A Data frame carries data from transmitter to the receivers.
- A Remote frame is transmitted by a bus node to request the transmission of the data frame with the same identifier.
- An Error frame is transmitted by any node on detecting a bus error.
- An Overload frame is used to provide for an extra delay between the preceding and the succeding Data or Remote frames.

Data frames and Remote frames are seperated from preceding frames by an Interframe space.

3.1.1. Data Frame

A Data frame is composed of seven different bit fields : Start of frame, Arbitration field, Control field, Data field, CRC field, ACK field, End of frame.

The Data field can be zero though.

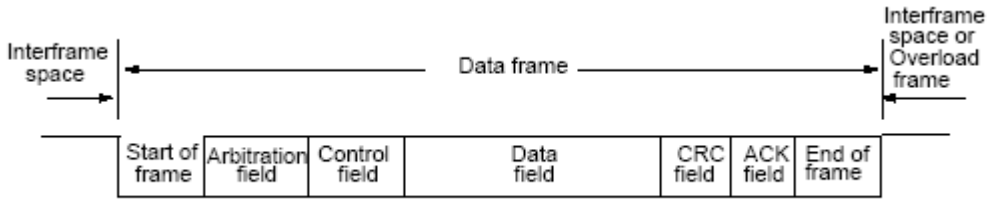

Figure 3.1 : Data Frame

Start of frame Marks the beginning of Data frames and Remote frames. It consists of a signal dominant bit. A node is only allowed to start transmission when the bus is idle. All nodes have to synchronize t the leading edge caused by start of frame.

Arbitration field consists of the Identifier and RTR bit.

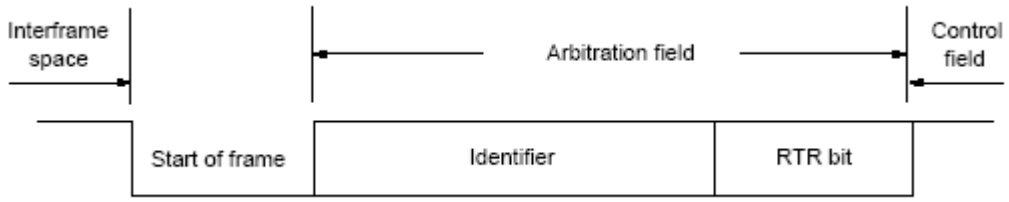

Figure 3.2 : Arbitration Field

Identifier is a a 11-bit data. These bits are transmitted in the order from ID10 to ID0. The least significant bit is ID0. The 7 most significant bits must not be all recessive. RTR is remote transmission request bit. In data frames the RTR bit must be recessive.

The control field consists of six bits. It includes the data lenght code and two reserved bits. The reserved bit must be sent as dominants. Receivers accept dominant and recessive bits in all combinations

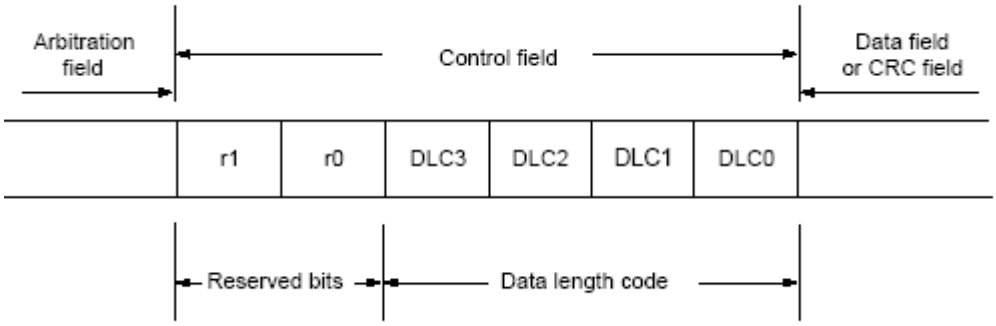

Figure 3.3 : Control Field

The number of bytes in the data field is indicated by the data lenght code. Data field is indicated by the data lenght code. This code is 4 bits wide and is transmitted within the Control Field. The DLC bits can code data lenghts from 0 to 8 bytes, other values are not permitted.

Data Field consists of the data to be transferred within a data frame. It can contain from 0 to 8 bytes, each of which contain 8 bits which are transferred MSB bits.

DATALENGHT CODE				DATABYTE
DLC3	DLC2	DLC1	DLC0	COUNT
d	d	d	d	0
d	d	d	r	1
d	d	r	d	2
d	d	r	r	3
d	r	d	d	4
d	r	d	r	5
d	r	r	d	6
d	r	r	r	7
r	d	d	d	8
d = dominant r = recessive				

Table 3.1 : Data Lenght Coding

The CRC field contains the sequence, followd by a CRC delimiter.

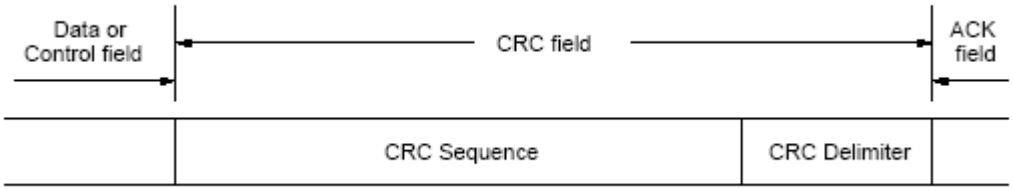

Figure 3.4 : CRC Field

The frame check sequence is derived from a cyclic redundancy code best suited to frames with bit counts less than 127 its (BCH code).

In order to carry out the CRC calculation the polynomial to be divided is defined as the polynomial whose coefficients are given by the destuffed bit-stream consisting of start of frame, arbitration field,control field,data field and fort he 15 lowest coefficients by 0. This polynomial is divided by the generator polynomial :

$X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$

The remainder of this polynomial divison is the CRC Sequence transmitted over the bus.

In order to implement this function, a 15-bit shift register CRC_RG(14:0) can be used. If NXTBIT denotes the next bit of the bit-stream, given by the destuffed bit sequence from Start of frame until the end of the Data field, the CRC sequence is calculated as follows :

```
CRC_RG = 0                                  //Initialize shift register
REPEAT
        CRCNXT = NXTBIT EXOR CRC_RG(14)
        CRC_RG(14:1) = CRC_RG(13:0)         //Shift left by..
        CRC_RG(0) = 0                       //..one position
        IF CRCNXT THEN
                CRC_RG(14:0) = CRC_RG(14:0) EXOR (4599 hex)
        ENDIF
UNTIL (CRC SEQUENCE start sor there is an ERROR condition)
```

After the transmission/ reception of the last bit of the Data field, CRC_RG(14:0) contains the CRC sequence.

The ACK field is two bits long and contains the ACK Slot and the ACK delimiter. In the ACK field the transmitting node sends two recessive bits.

A receiver which has received a valid message correctly reports this to the transmitter by sending a dominant bit during the ACK slot

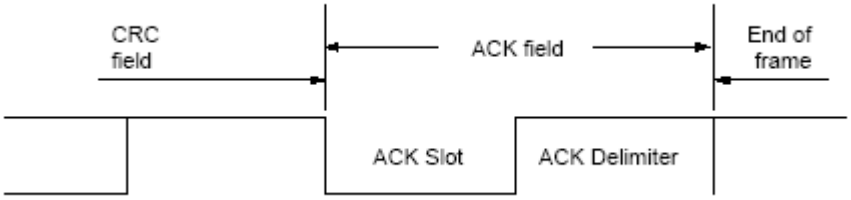

Figure 3.5: ACK Field

Finally, each data frame and remote frame is delimited by a flag sequence consisting of seven recessive bits called *End of Frame.*

3.1.2 Remote Frame

A node acting as a receiver for certain data can stimulate the relevant source node to transmit the data by sending a remote frame.

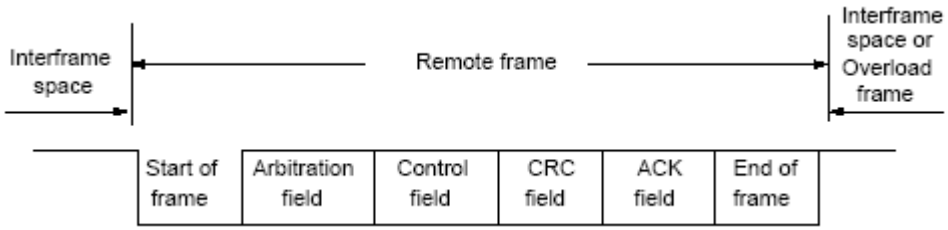

Figure 3.6: Remote Frame

A remote frame is composed of six different bit fields : Start frame, Arbitration field, Control field, CRC field, ACK field, End of frame.

The RTR bit of a Remote frame is always recessive.

There is no data field in a remote frame, irrespective of the value of the Data Lenght code which is that of the corresponding data frame and may be assigned any value within the admissable range, from 0 to 8.

3.2. Error Frame and Fault Handling

The Error frame consists of two distinct fields. The first field is given by the superposition of error flags contributed from different nodes. The second field is the error delimiter.

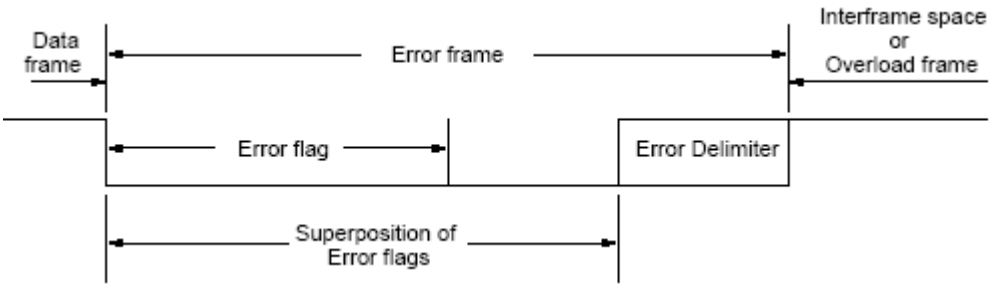

Figure 3.7: Error Frame

In order to terminate an error frame correctly, an error-passive node may need the bus to be bus-idle for at least three bit times. Therefore, the bus should not be loaded in full capacity.

Error frame's one of the component is error flag. There are two forms of error flag : active and passive. Active error flag consists of six consecutive dominant bits. Passive error flag consists of six consecutive recessive bits unless it is overwritten by dominant bits from other nodes.
An error-active node detecting an error condition signals by the transmission of an active error flag. The error flag's from violates the law of bit stuffing, applied to all fields from start frame to CRC delimiter or destroys the fixed form ACK field or End of frame field.

As a cosequence, all other nodes detect an error condition and each starts to transmit an error flag. So the sequence of dominant bits which actually can be monitored on the bus results from a superposition of different error flags transmitted b individual nodes. The total lenght of this sequence varies betwen a minimum of six and a maximum of twelve bits.

An error passive node detecting an error condition tries to signal this by transmitting a passive error flag. The error passive node waits for six consecutive bits of equal polarity, beginning at the strt of the passive error flag. The passive error flag is complete when these six equal bits have been detected.

3.3. CAN Bit Timing

In CAN based transfer there is a unit time called *Nominal Bit Rate*. Nominal bit rate is the number of bits per seconf transmitted in the absence of resynchronization by an ideal transmitter.

Nominal Bit Time = 1 / Nominal Bit Rate **(3.1)**

The Nominal Bit Rate can be thought of as being divided into seperate non-overlapping time segments. These segments are as shown below

- SYNCHRONIZATION SEGMENT (SYNC_SEG)
- PROPAGATION TIME SEGMENT (PROP_SEG)
- PHASE BUFFER SEGMENT1 (PHASE_SEG1)
- PHASE BUFFER SEGMENT2 (PHASE_SEG2)

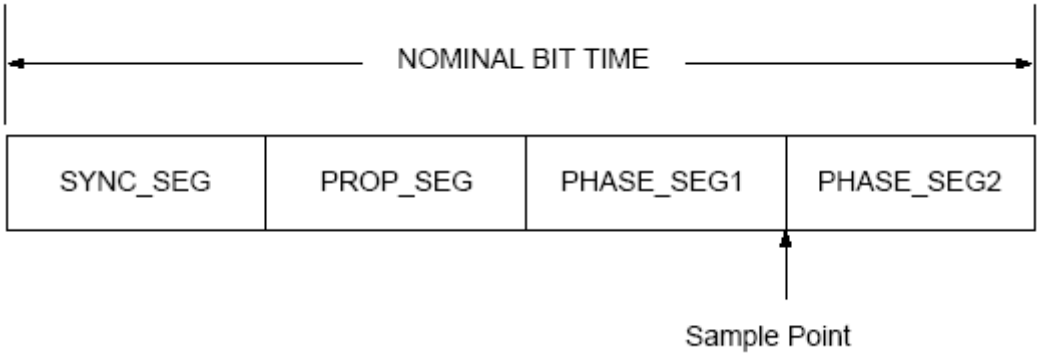

Figure 3.8 : Nominal Bit Rate

SYNC_SEG : This part of the bit time is used to synchronoize the various nodes on the bus. An edge is expected to lie within this segment.

PROP_SEG : This part of the bit time is used to compansate fort he physical delat times within the network. It is twice the sum of the signal's propagation time on the bus line, the input comparator delay, and the output driver delay.

PHASE_SEG1, PHASE_SEG2 : These phase-buffer segments are used to compansate for edge phase errors. These segments can be lenghtened or shorthened by resynchronization.

Sample Point : The sample point is the point in time at which the bus level is read and interpreted as the value of that respective bit. Its location is at the end of PHASE_SEG1.

Information processing time is the time segment starting with the Sample Point reserved for calculation of the subsequent bit level. *The time quantum* is a fixed unit of time which can be derived from the oscillator period. There is a programmable prescaler, with integral values (with a range of at least 1 to 32) which allows a fixed unit of time, the time quantum can have e lenght of :

TIME QUANTUM = m * MINIMUM TIME QUANTUM **(3.2)**

Where m is the value of prescaler.

Following are the lenght of time segments :

- SYNC_SEG is 1 time quantum long.
- PROP_SEG is programmable to be 1,2,…..8 Time quanta long.
- PHASE_SEG1 is programmabler to be 1,2,…..8 Time quanta long.
- PHASE_SEG2 is the maximum of PHASE_SEG1 and the Information processing time.
- The Information processing time is less than or equal to 2 Time quanta long.

The total number of Time quanta in a bit time must be programmable over a range of at least 8 to 25.

4. EIGHT-LEGGED WALKING ROBOT DESIGN

Arthropoda are one of the biggest phylums, which include numerous invertebrate organisms such as insects, spiders and cruscateans. Organisms in this phylum are characterized by a hard, segmented external covering and segmented limbs which strenghten them against negative circumstances as they are able to survive.

In this thesis, all designs and studies are aimed at constructing a multipod mechanism which can autonomously move around and accomplish specified goals. The robot includes 2 wagons that are connected to each other by a passive joint, so that, it is more advantegous than other single-wagon mobile multipods while locomoting on curved or twisted trajetories. In order to improve stability, each wagon has 4 legs and each leg has three degrees of freedom. All in all, the robot has 24 degrees of freedom which enhances flexibilty while moving on rough terrain.

The design process may be examined in two stages : Electrical Design and Mechanical Design. In Mechanical Design, some metodologies are applied in order to have a robust and stable mechanical structure. In Electrical Design, the main reason is to obtain intelligency for the robot as well as providing the necessary sensor - actuator interfaces and various automatic control strategies. Having completed these two stages, mixed system integration will be implemented to the whole system by a mechatronic point of view.

4.1 Mechanical Design

In this stage, the main purpose is obtaining the whole mechanical structure. Mechanical Design consists of such phases; constructing the whole body, specifying dimensions, gait analysis, performing inverse kinematic transforms and manifacturing the mechanical frame. In figure 4.1 and figure 4.2, our mechanical system is displayed from different angles.

In the below figures the whole mechanical frame can be seen from different angles. By constructing these computer aided drawings, a healthy manifacturing phase is aimed.

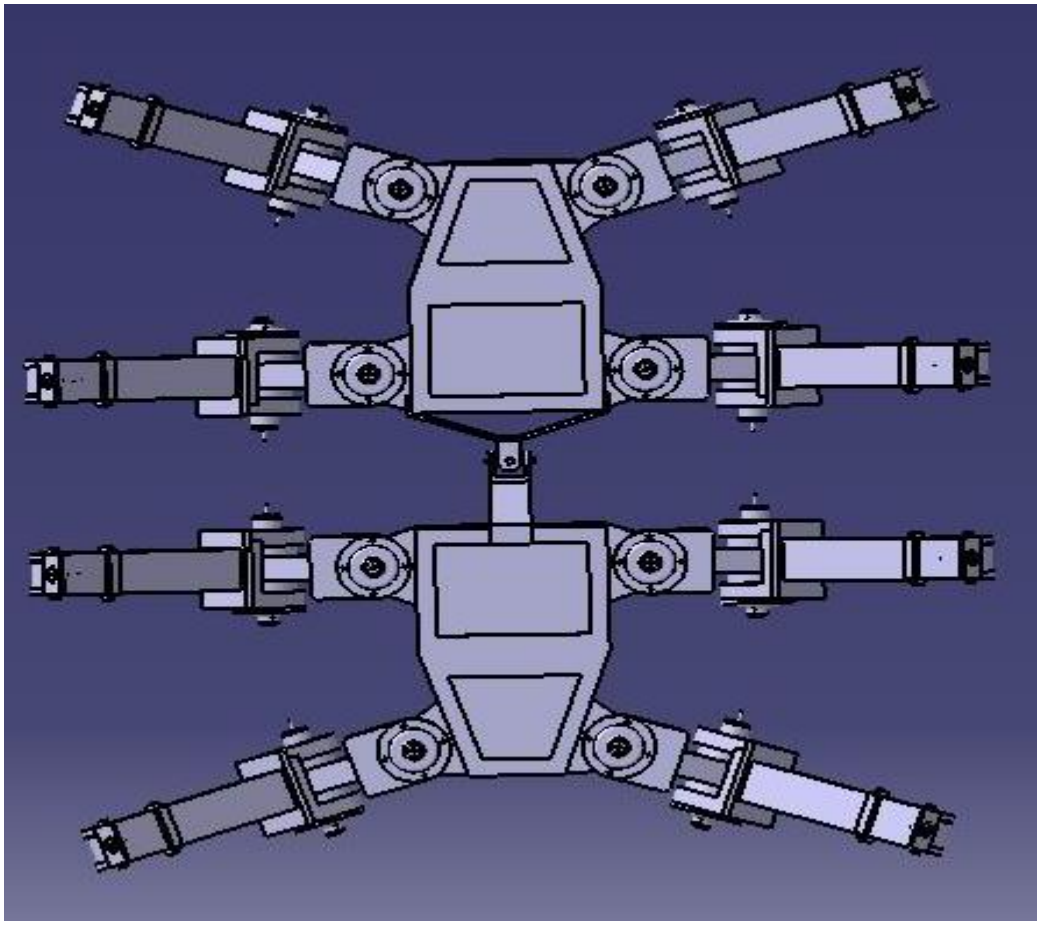

Figure 4.1 : Top Angle

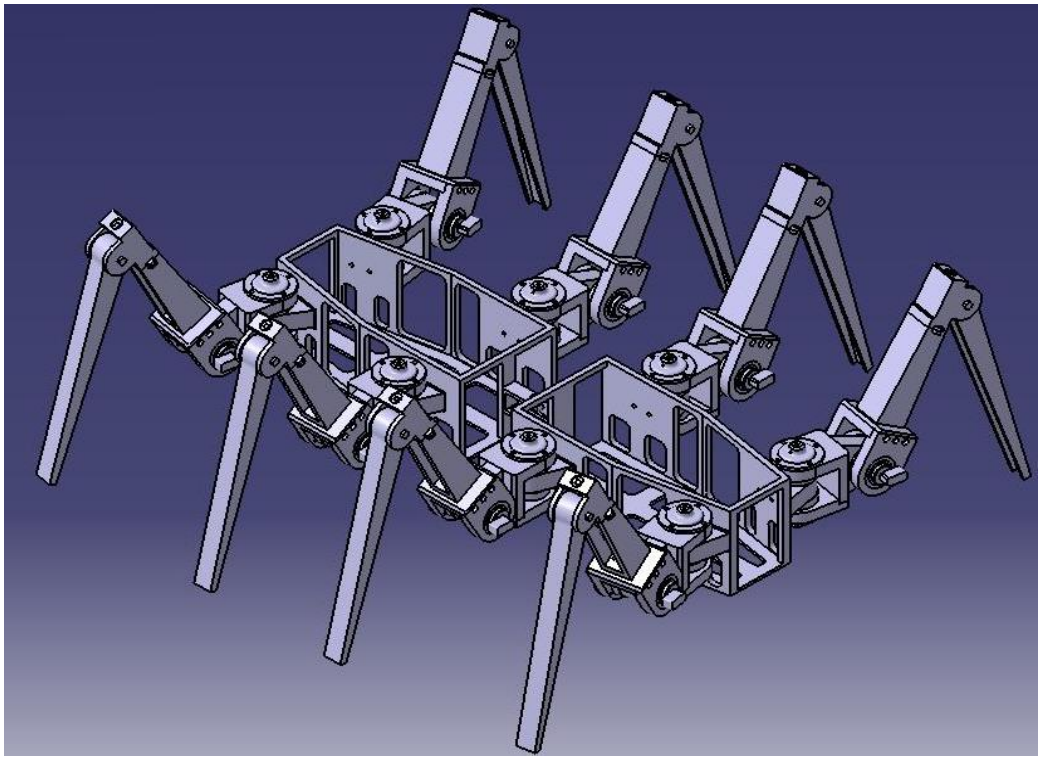

Figure 4.2 : Perspective Angle

As mentioned in the introduction, our robot has 2 wagons that are attached to each other by using a passive joint. These 2 wagons include 4 legs with 3 degrees of freedom. In total the robot has 24 degrees of freedom. Each leg is actuated by 3 motors and there is a force sensor at the bottom of these legs in order to determine the prussure at the pod. In Figure 4.3, actuator placements are displayed from the top angle.

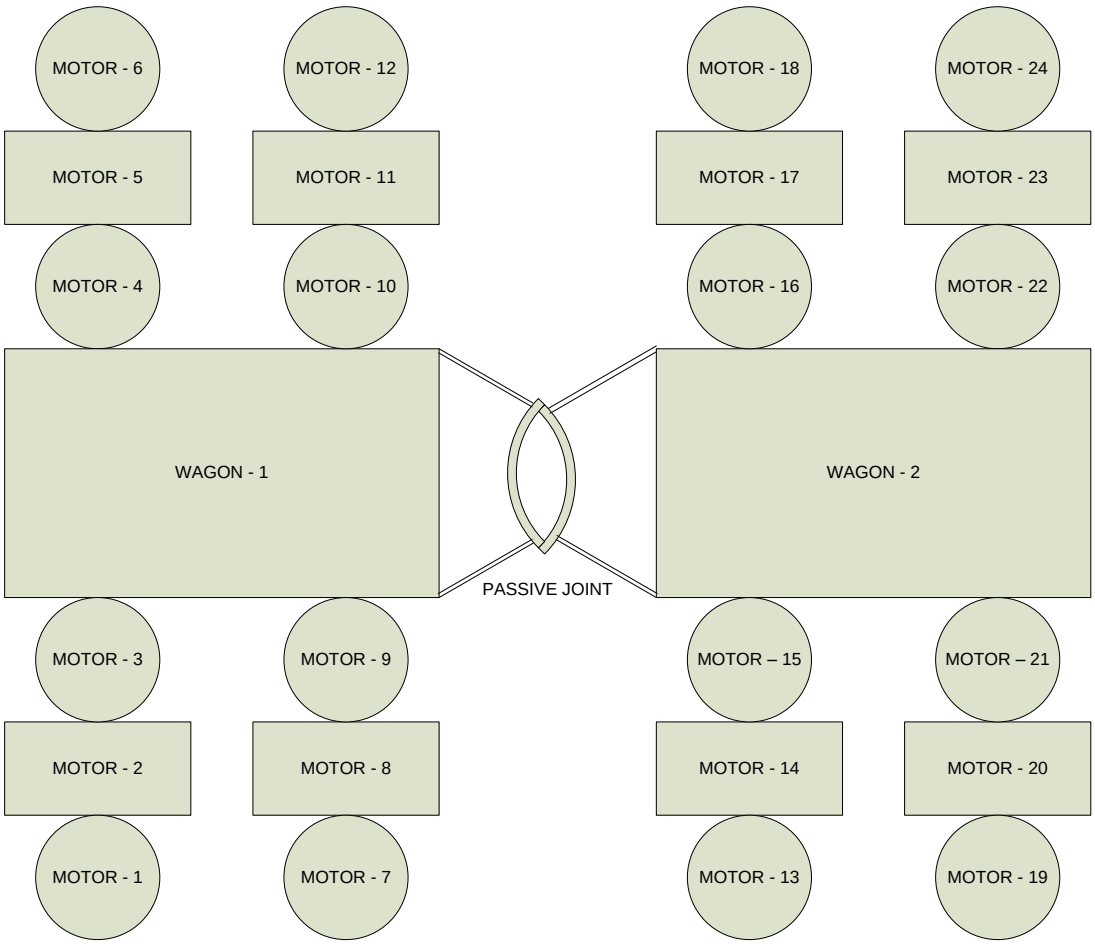

Figure 4.3 : Actuator Placements on the Robot

This part of design is realized in cooperation with another Master's Thesis[3] which is executed synchronously with this thesis. Executing this simultaneous thesis studies, a healhty fusion of these systems will be obtained.

4.2 Electrical Design

The main idea in this stage is to provide an hardware background in order to yield a physical basis, on which software applications and control strategies are implemented. Having this purpose explained above; Electrical Design's phases are constructing whole electronic circuits such as brushless motor drives and electronic control units, constituting printed circuit boards (PCBs), specifying sensor and actuator types, determining the control strategies and proper software design.

Before designing all circuits and necessary components, all hardware structures and their inner relations are specified as 3 block diagrams mentioned above. These block diagrams are called *logical layers*.

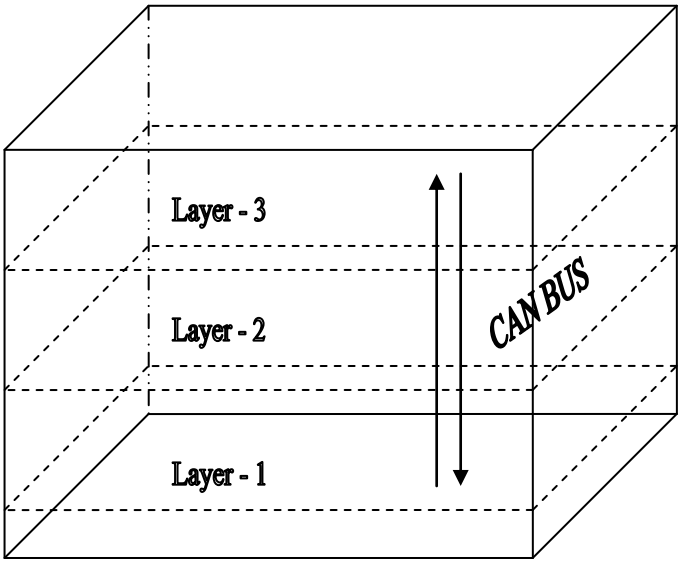

Figure 4.4 : Logical Hardware Layers

The whole hardware structure is groupped into 3 layers due to the fact that each of them has its own function. Eventhough they are realizing different operations, the interferences between them is positively matched and tuned.

Taking into account the fact that hierarchical control levels are to be implemented; this 3 layered model is also quite helpful during throughout development process.

4.2.1 Pre-Design Process and Logical Layers

Having explained previously, our design process has to cover many crucial functions. Hence, each of them are examined induvidally. In that part of the thesis, logical layers' functions will be explained, regardless of their physical meanings and with no respect to design techniques

In Figure 5.5. the fist layer is shown. As seen in the figure, this layer also includes three different sublayers. Sublayer 1 -1 surrounds all connection groups, including servo motor signal and power pins, encoder signal pins and force sensor connections. In this sublayer, distributed power levels are emitted to whole system. Degrading 24Volts to other voltage levels is realized in Sublayer1-2. Buck regulators are used to step down 24Volts. Layer 1-3 only includes and RJ45 socket which connects our central processing system to the internet.

One of the crucial problem that may occur in mobile robot designing is printed circuit board location. Having 24 degrees of freedom in our system, we need to stack 24 quantity of brushless servo motor drives. Therefore, all motor drives are placed vertically onto a base PCB in each wagon. This base PCB includes 12 quantity of female motor drive sockets and step down voltage regulators.

Seperating all connector groups and pins, we are able to adjust physical locations of these components easily. Totally, 120 outer connection is fulfilled in order to supply these components and maintain data transfer.

All cabling works and connection organizations are done with respect to mechanical dimensions. Therefore some of the cabling plans are also linked to mechanical structure while designing.

Being processed as a prototype, there is no batteries attached to the robot's body. In first manifacturing phase, whole system will be fed externally from an outer power supply. In other versions of manifacturing, these layer will also include battery - hardware interface within necessary charging circuits and commutation elements.

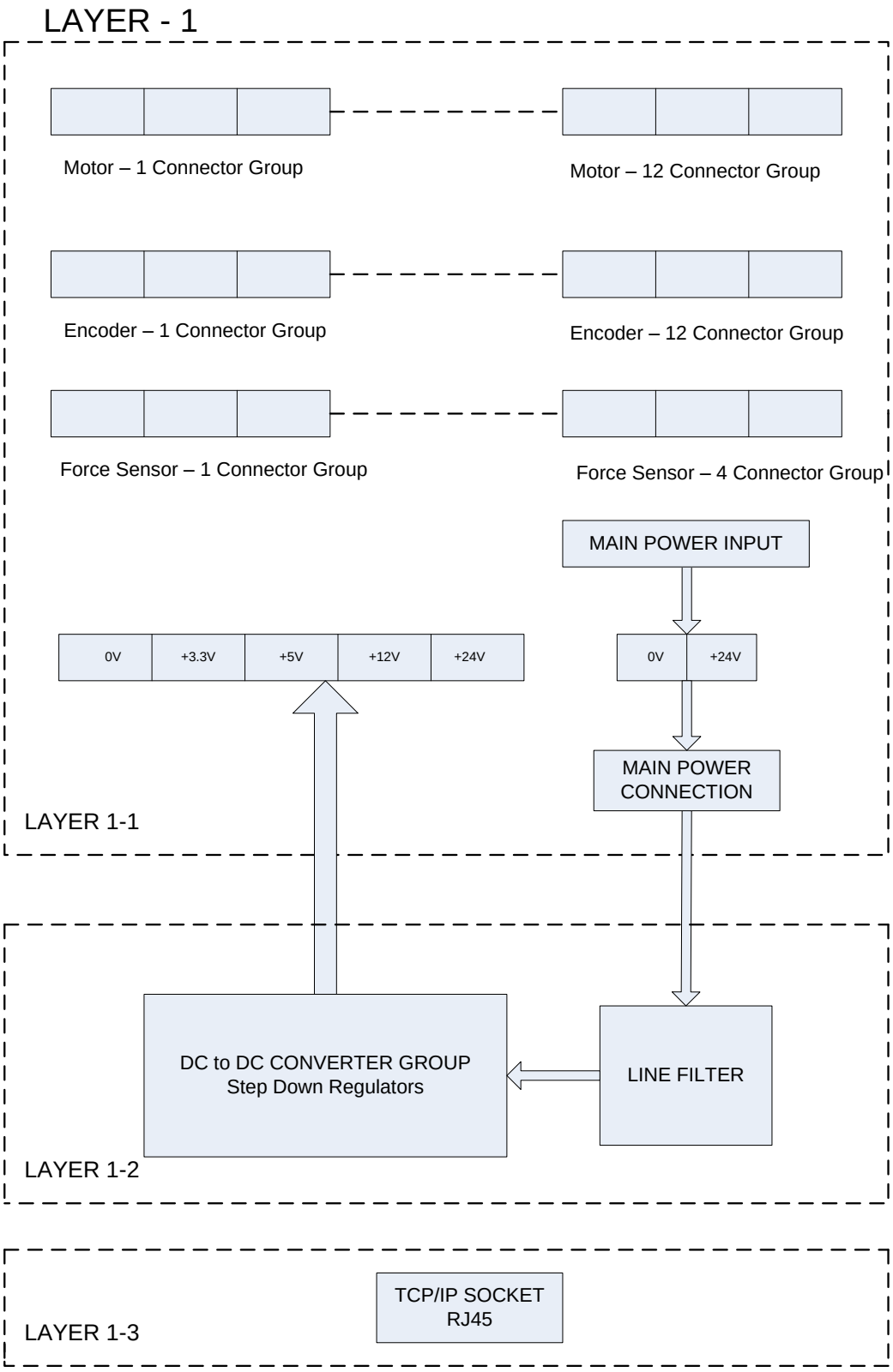

Figure 4.5 : Logical Layer – 1

To sum up, Layer – 1 only includes electrical connections and power distribution in order to feed the whole system.

Considering the whole system, Layer -2 appears to be one of the most crucial part as it includes 2 different control levels. Both mid level and low level control structures are designated in this layer. Therefore it gains importance as it is responsible for fundamental behaviours.

Relevant diagram for the Layer – 2 is displayed in Figure 5.6. As seen below; motor drives, low level and mid level controllers, force sensors and other related components are placed in Layer-2.

In each wagon, there are 12 motors. Each servo motor driver is driven by a controller as software based embedded control application is required for brushless servomotors. Being a complete system within its own behaviours, they are called *joint drivers*.

A joint driver includes a low level controller. This controller is responsible for generating necessary driver signals with respect to position information. This position information is provided by optical encoders that are attached to shafts. Other than that, each joint has a mechanical constraint. To be able to warn controller while the joint reaches the maximum or minimum angle, limit switches are added to each joint.

For converting DC energy into 3-phase AC energy, power electronics are used. A 2-level converter is implemented in order to accomplish this conversion by evaluating control signal which are generated by our low level controller.

Mid level controller is more like a bridge that provides a smooth trainsition between actuators and high level contol unit. There is only one mid level controller in each wagon and it appears to control joint drivers. It takes orders from the high level controller and interprets them for the low level controller as it performs inverse kinematics. On the other hand, force sensors are connected to mid level controller, so it is responsible to digitize their analog information and transfer it to high level control unit

In order to save important parameters in this layer, an EEPROM with a high capacity is linked to mid level controller via I2C Bus.

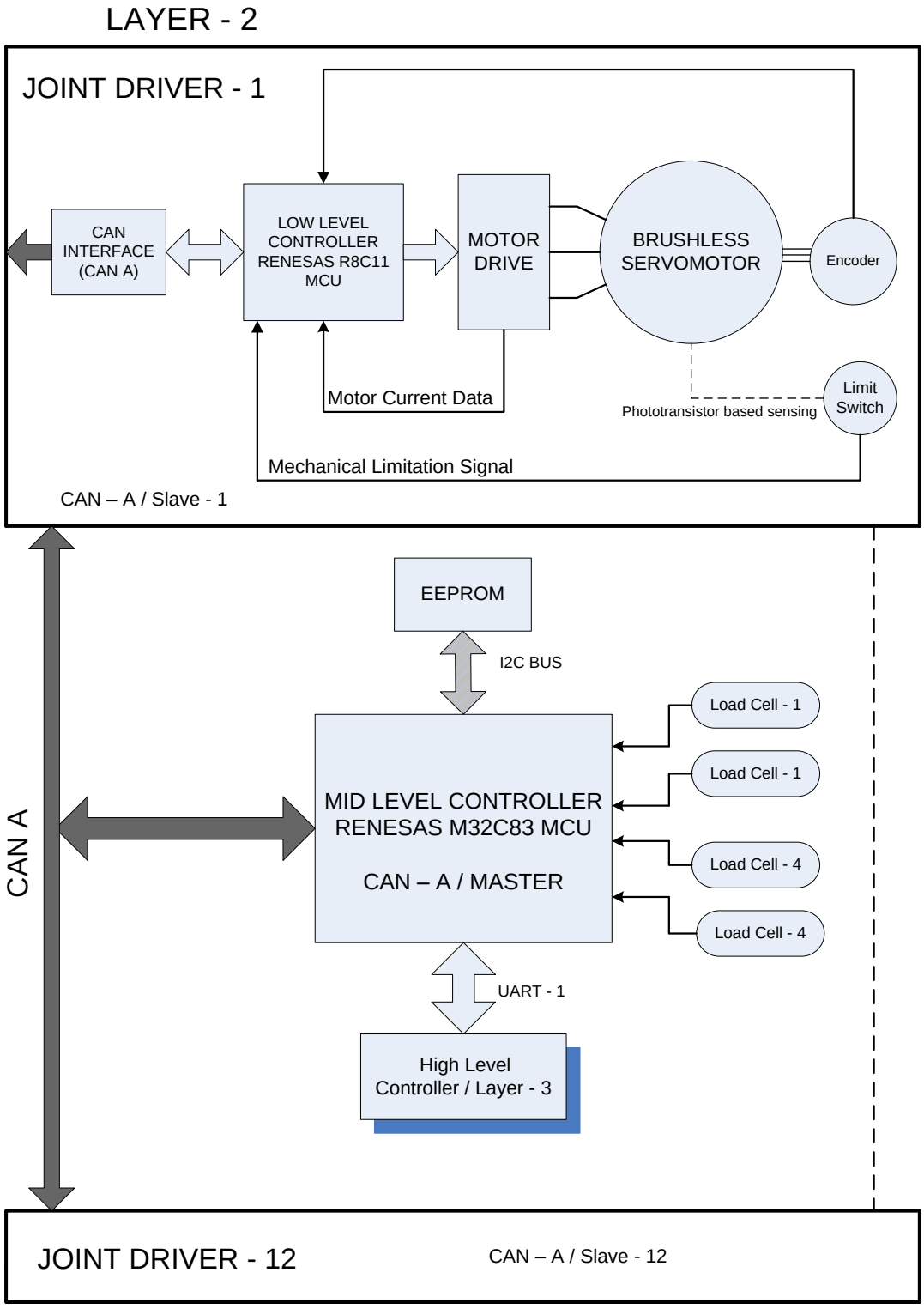

Figure 4.6 : Logical Layer – 2

As seen above, there are 12 joint drivers and only one mid level controller. The communication between them is performed via CAN Bus. CAN Bus is a very specialized communication bus with highly developed electrical noise immunization and addressing techniques.

The CAN Bus that is in Layer – 2 is called CAN – A in our system. Mid level controller is the *master* and all other joint drivers are *slaves* in CAN – A's hierarchy. The main reason to use CAN in Layer – 2 is to provide easy and stable addressing between mid level controllers and joint drivers.

Layer – 3's function is similar to brain in human body. It includes high level controller where the advanced control techniques are applied. There is not only high level controller unit, but also external sensors are placed within Layer – 3.

All external sensors, such as Inertial MEMS, doppler, CCD camera and many others are connected to high level controller. Also force sensors are evaluated by high level controller through mid level controller. Evaluating these sensor values within a sensor fusion algorithm; many tasks can be performed in this layer.

The communication network between sensors and controller is CAN Bus again. This CAN Bus is called CAN – B in our system. In CAN – B, high level controller is the *master* and all external sensors are *slave*s. The main reason to use CAN Bus in Layer – 3 is to enhance flexibility on numbers of sensors. For example, in following versions of the robot, different sensors may be added. Using CAN Bus herein, we can easily plug-in and plug-out external sensors as much as we want in respect to control methodology we are about to apply. It also offers a simple and stable addressing.

High level is mainly responsible for trajection control, force control, dynamic model based control, sensor fusion and other highly developed control methodologies. Using the synergetic combination of these phonomena,

High level controller is always in communication with mid level controller. The communication media is not a profibus system. There is only a serial communication

interface between these controller units. Using a simple serial communication interface is more advantageous over profibus systems. Because there is only two clients, so that, there is no need to addressing.

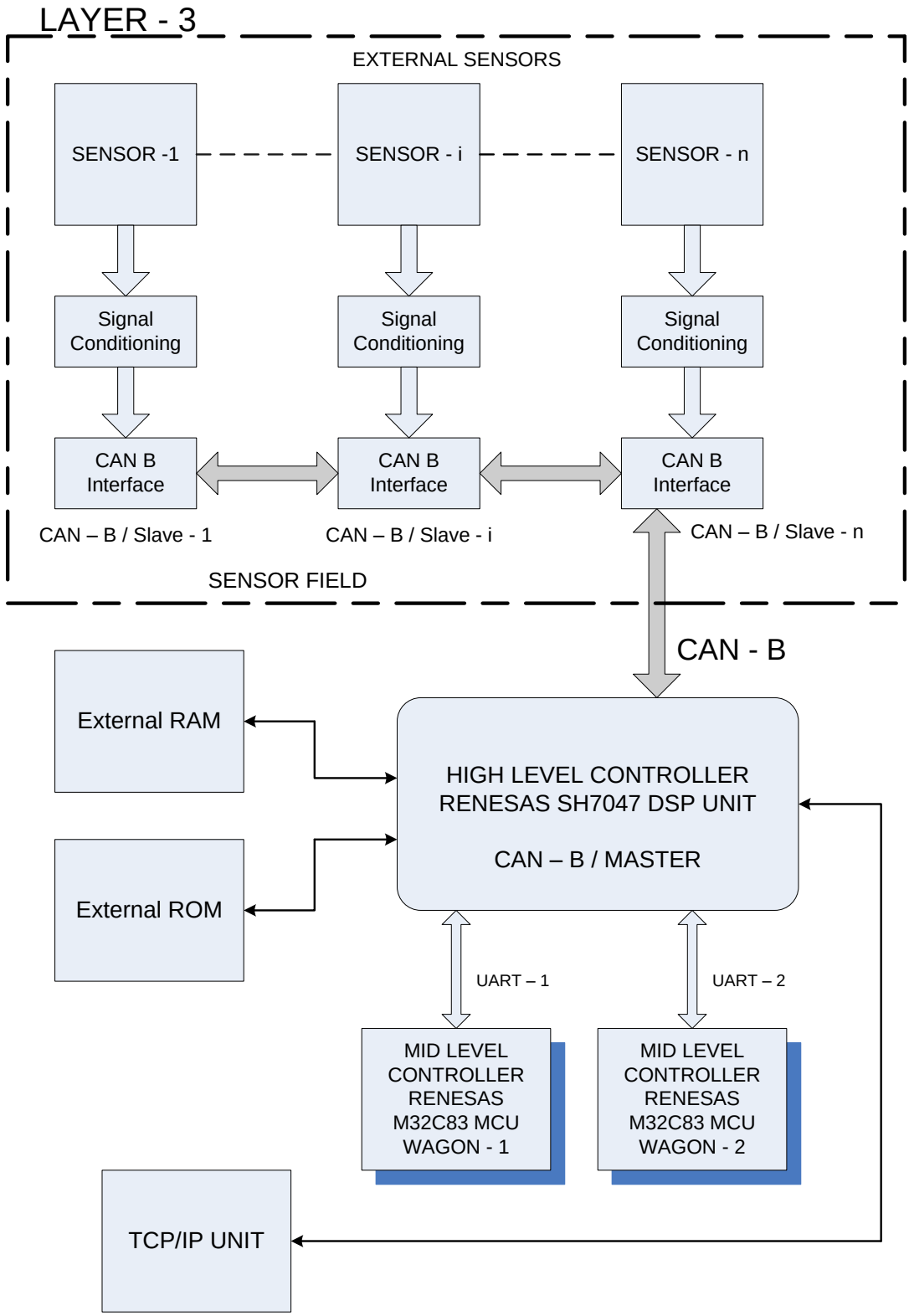

Figure 4.7 : Logical Layer – 3

Generally, high level controller gives orders to mid level controller with respect to control method we are using. In contrast to that, mid level contoller feedback it and transfers the force sensor data.

4.2.2 Joint Drives and Power Electronics

In this chapter, power conversion elements and actuator driver circuits are explained. Our power electronics based electronic circuits are DC to DC converters and brushless servo motor drives.

Being a multilayered control based system; our electronic circuits need different levels of DC voltage. Therefore, DC to DC converters are used to supply our circuits. Following values are our DC values that our system needs :

- +24V
- +18V
- +12V
- +5V (Vcc)
- +5V_CAN (CAN_Vcc)
- +5V Ref (V_{ref})
- +3.3V

+24V is our input voltage, other levels are obtained from this supply voltage by chopping it. Also +24V is used for motor feed and optical encoder supply voltage. +18V is necessary for Hall Effect Sensors. +12V is used for force sensors. Vcc is our logical supply voltage. CAN communication Bus requires a seperate supply voltage in order to prevent electrical noise that might be generated from peripheral circuitries. Thus, it has special +5V supply voltage, called CAN_Vcc. V_{ref} is reference voltage for embedded Analog to Digital converters. It must be quite accurate, so that, a special IC is used to provide the exact value of +5V. It is also seperated from other +5V loops. Our high level controller is fed with +3.3V, hence a regulator for +3.3V is also added.

Two types of DC to DC converters are used. Classical L78xx series of voltage regulators and special Buck Regulators. The classification is done by considering energy consumptions for each level.

To be able to obtain +12V, +18V and CAN_Vcc L78xx series of simple regulators are used. Because they only supply single components. For example, only force sensors need +12V and only Hall Effect Sensors need +18V. Likely CAN-Vcc is only responsible for feding CAN Bus. In these regions, power consumption is below 1A. Therefore L78xx series of simple voltage regulators are used because of their low price. Some capacitors are placed at these regulators' input and output terminals in order to eliminate ripples. 100nF capacitors eleminate noises with high frequencies. 100µF capacitors cancel out noises with lower frequencies.

Vcc value is generated by using a buck regulator. For that purpose, National Semiconductors' LM2596 is implemented to the system. It supplies maximum of 3A current value. It needs external componets, some capacitors, a schottky diyote and an inductance. All these components' values are recommended in the datasheet. Hence, the design is fulfilled by putting corresponding elements through these recommendations.

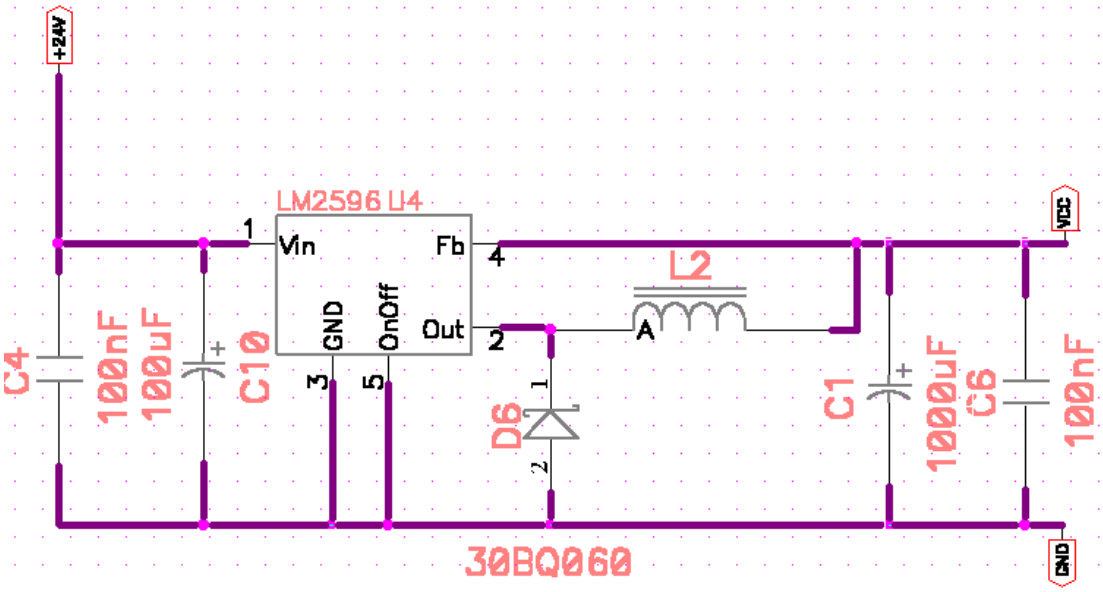

Figure 4.8: Buck Regulator

Analog voltage reference is generated by a special IC, called REF02SM. It outputs an exact value of +5V DC value which prevents errors while converting analog data into digital data.

As explained previously, each wagon has 12 degrees of freedom, thus each wagon has 12 joint drives. However, there is physical constraints because of mechanical limitations. Therfore, all joint drives are stacked and placed onto a base PCB vertically. On this base PCB, DC to DC converters are placed too. EDG type sockets are used to to stack joint drives. All connections and DC power connectors are drawn in the schematic called “driver_base&power” which can be seen in Appendix part of the thesis.

As seen in Figure 5.6, a joint driver includes following subblocks: low level controller (CPU Block), CAN Bus interface, motor drive unit, encoder and limit switches.

In joint drivers, Renesas R8C11 Tiny microprocessors are used as controller. It is a 16-bit highly developed low cost microprocessor with 16K ROM and 1K RAM. Other than that it is very suitable for floating point operations and includes 10 bit ADC module. These qualifications are excellent enough for our needs.

In CPU Block, there are peripheral IC devices in order to support CPU functions. ADM705 is a micrprocessor supervisor IC, which is able to deteckt power fail. When input voltage is below the threshold it generates a signal and triggers the NMI interrupt in our controller. Therefore the controller save the necessary data and resets itself. Other IC is analog reference IC, REF02SM. Its function is explained above.

Since, our brushless motors need 3-phase AC voltage, a DC to AC converter is implemented to our joint drivers. For that purpose 2-level converters are used.

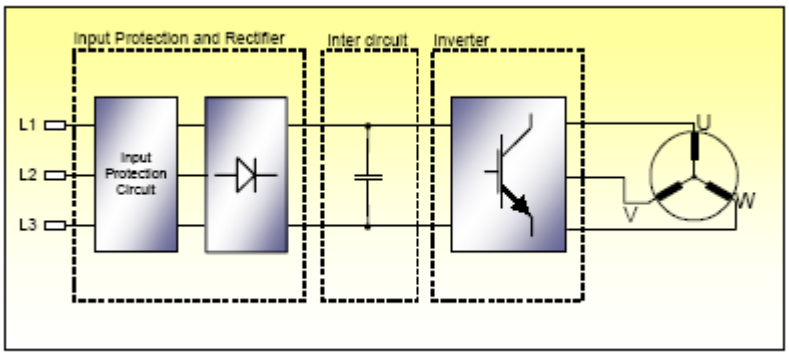

Figure 4.9 : Basic Converter Structure

2- level converter consists of six pulse bridge which provides conversion to AC from DC. The phases are controlled by control signal pairs, A, B and C. Each pair consists of two switches, upper and lower. In our drive, mosfets are used for these switches. Upper swithces are is sourcing, a p-channel mosfet IRF9540. Lower switches are sinking, an n-channle mosfet IRFZ44N.

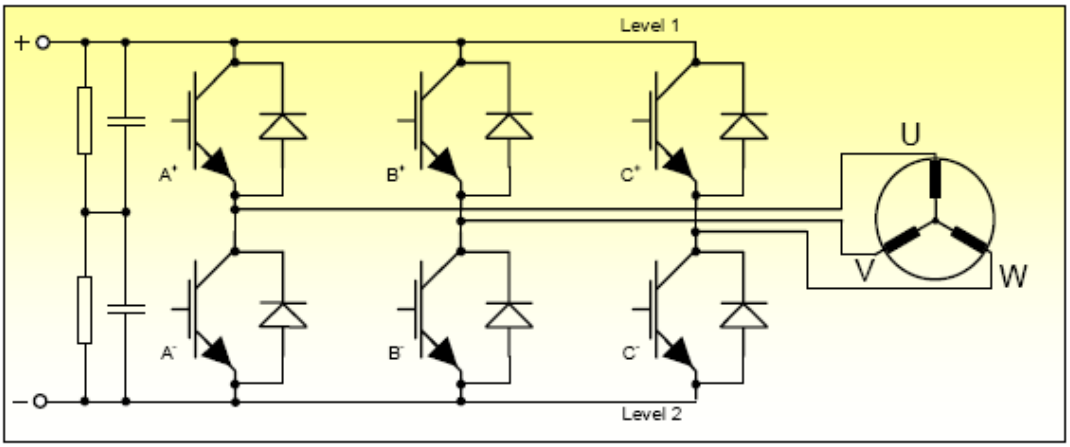

Figure 4.10 : 2-level AC to DC Converter

Eventhough different types of motors are used in the mechanical design, only one brushless servo motor is prepared because of compability. Joint drivers are able to drive any motors that are used.

Mosfets are driven by general purpose transistors. P-channel mosfets are driven by npn transistor BC846 and N-channel mosfets are driven by pnp transistor BC858. The maximum current that flows is 5A. Because of that, mosfets are configured to conduct at least 6A.

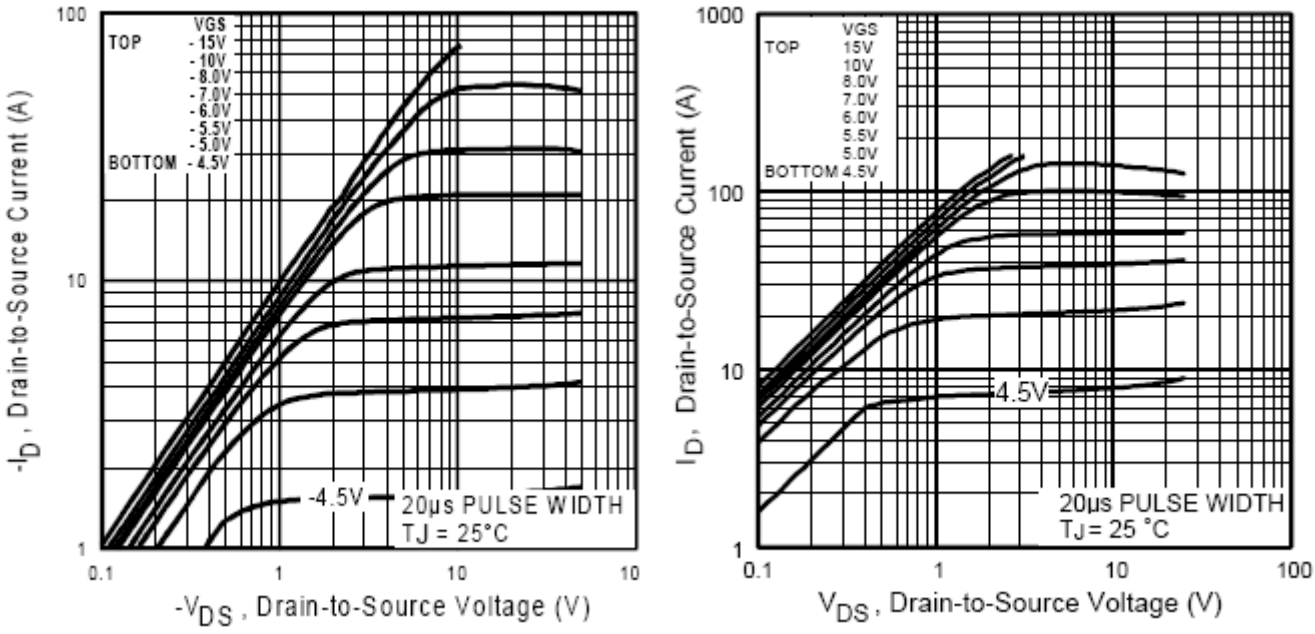

Figure 4.11 : Mosfet Characteristic Curves, IRF9540 and IRFZ44N respectively

Examining the characteristic curve which is above, for V_{GS} = -6V, we can obtain 10A channel to use while feeding the system with +24V.

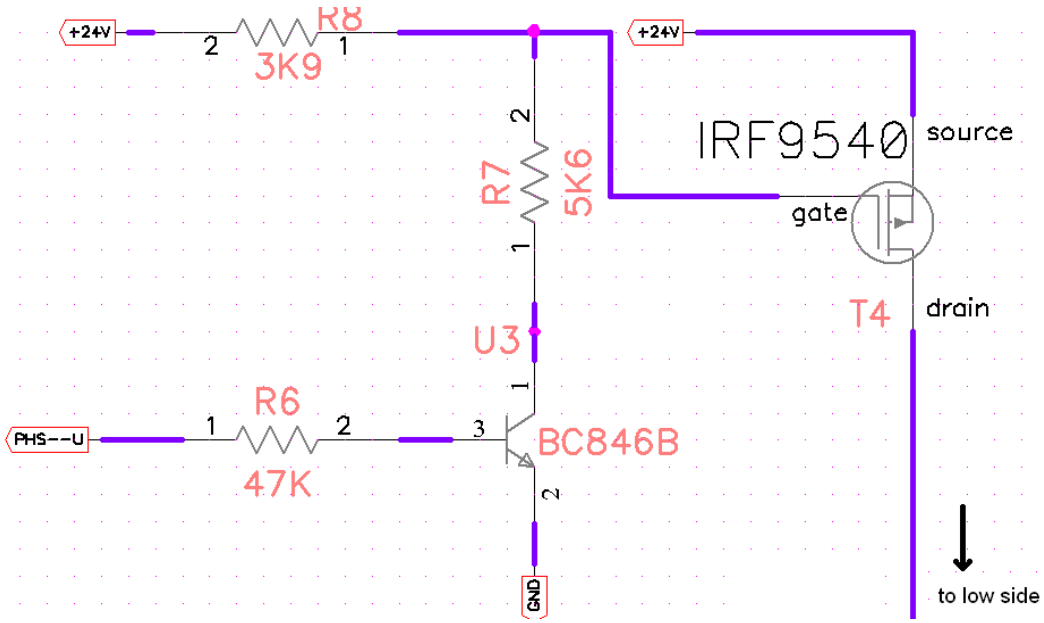

Figure 4.12 : Motor Drive, Single Hide Side Switch Circuitry

Herein, V_S = 24V and V_{GS} = -6V. Then we can calculate the Gate voltage, V_G = 18V. In above circuitry, our configuration can be seen. BC846 npn transistor has a

working point at which V_{CE} = 5V, I_C = 2mA and $\beta_{DC(mean)}$ = 200. Within the arrangement shown above, there are two meshes.

$$V_{CC} = I_C * (R_8 + R_7) + V_{CE} \quad \textbf{(5.1)}$$

$$V_{BB} = (I_C / \beta_{DC}) * R_6 + V_{BE} \quad \textbf{(5.2)}$$

$$24 = 2 * (R_8 + R_7) + 5 \quad \rightarrow \quad (R_8 + R_7) = 9.5\ K\Omega$$

If we choose R_8 = 3K9 and R_7 = 5K6, it is justified $(R_8 + R_7) = 3.9 + 5.6 = 9.5\ K\Omega$. What is more our V_G and then V_{GS} voltages are provided as desired

$$V_G = V_{CE} + I_C * R_7 \quad \textbf{(5.3)}$$

$$V_{GS} = V_G - V_S \quad \textbf{(5.4)}$$

$$V_G = 5 + 2 * 5.6 = 16.2V$$

$$V_{GS} = 16.2 - 24 = -\ 7.8V$$

From (5.2), R_6 resistor's value can be calculated :

$$5 = (R_6 * (2/20)) + 0.7 \quad \rightarrow \quad R_6 \cong 47\ K\Omega.$$

This V_{GS} voltage provides a channel for more than 10A value. If the maximum current values are analyzed in our system, they are all much below of the 10A value. Thus, this configuration provides a quite safe driving chance, even if the motor consumes its maximum power.

In the low side part, there is a similar circuity with a fundamental difference. Since our explained circuitry sources; low side sinks the current. In order to provide it, n channel IRFZ44N mosfets are used. This mosfets are driven by BC858 general purpose pnp transistor within a similar configuration.

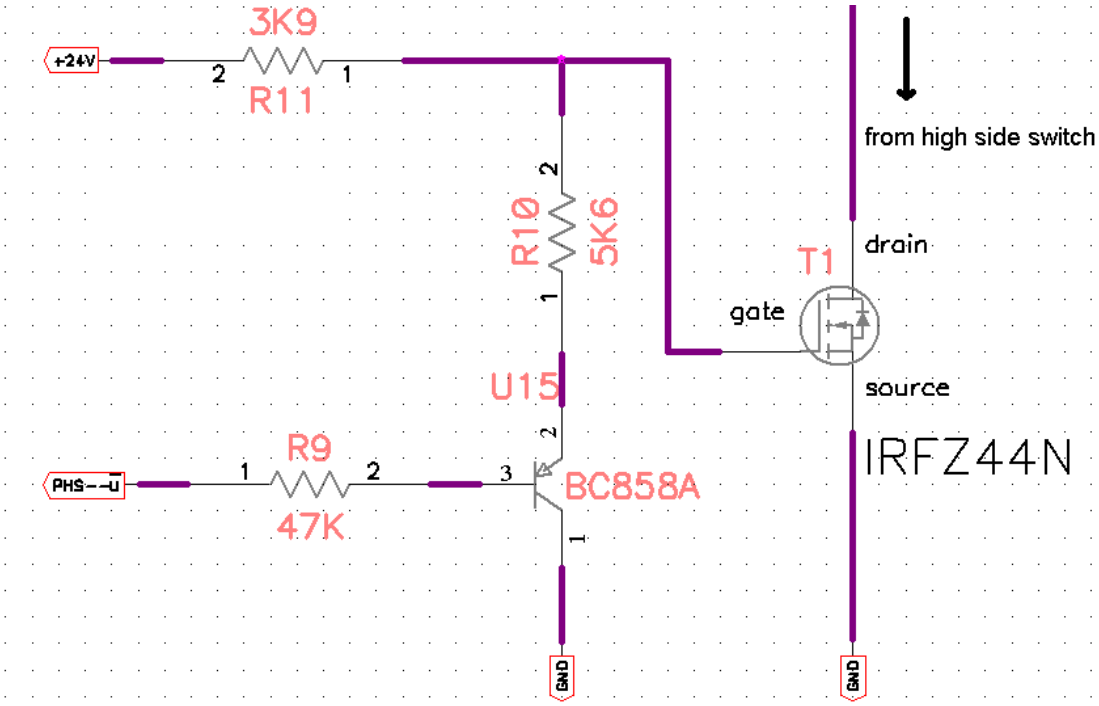

Figure 4.13 : Motor Drive, Single Low Side Switch Circuitry

This circuitry has the same configuration as we applied to its high side. However, only VGS value is a bit different. In this part;

$V_G = V_{CE} + I_C * R_{10}$ → $V_G = 5 + 2 * 5.6 = 16.2V$

$V_{GS} = V_G - V_S$ → $V_G = 16.2 - 0 = 16.2V$

This positive V_{GS} value is pretty enough to drive our N-channel mosfet, as it may be seen in Figure 5.11. It provides a channel that can support greater than 10Amps.

Distributing those coupled bridges into 3- phase, the whole 2-level AC to DC converter is realized. Hence, 3 distinct couple bridges are used. The most important thing while driving is, a full bridge's switches must not be triggered at the same time as it causes short circuit. The precaution that prevents this short circuit is applied in software of the Joint Driver.

While applying current control method to our actuators, we may need current information. Therefore, current sensors are applied to collect real-time this current values.

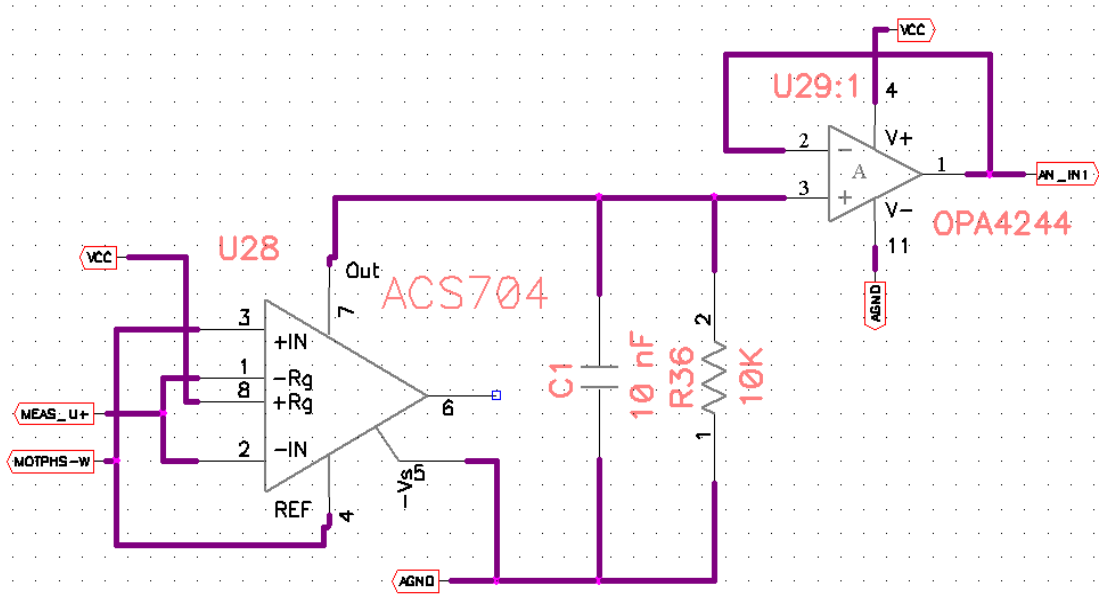

Figure 4.14 : Motor Current Sensor and Peripherals

ACS704 current sensors are employed to gather current data. It has an input-output couple, so that, a serial current bus is implemented. It outputs an analog voltage value, proportional to the current value which flows through in it. Digitizing these analog voltage value in the low level controller, we can exract the simultaneous current value. Between the microprocessor and sensor output, a voltage follower is placed in order to provide signal conditioning. C_1 and R_{36} are placed in paralel and they function as a simple passive filter. Their values are recommended in sensor datasheet.

Signal conditioning is quite important in a microprocessor based electronic control unit. As explained above, our motor drive's converter's base pins are triggered by sine pwm, which is generated by our microprocessor. However there must be a buffer zone between those 2 blocks to prevent unwanted situations, such as inverse currents. To realize it, series of optocouplers and schmitt triggers are employed between motor drive's converter's base pins. Same configurtion is also applied between encoder signal pins and microprocessors.

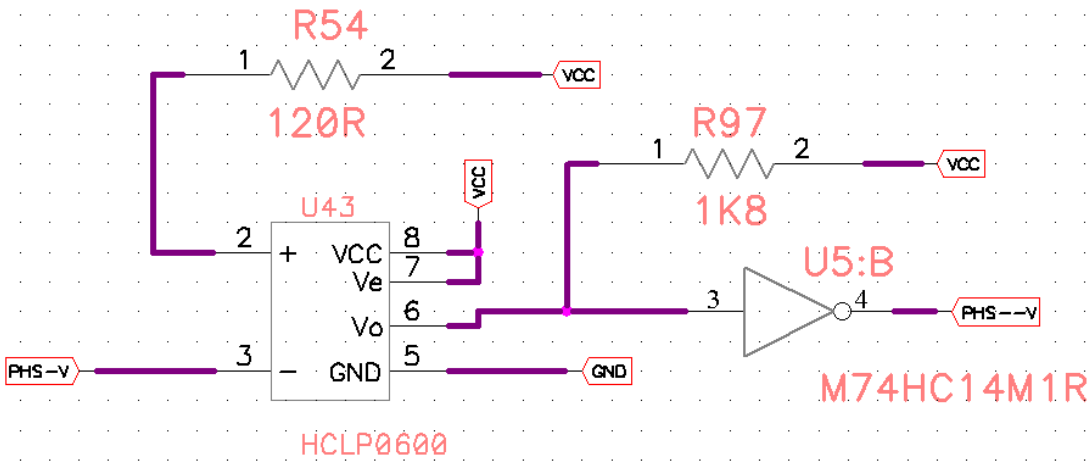

Figure 4.15 : Digital Signal Conditioning

In this type of a buffer zone, the signal is firstly applied to optocoupler which causes electrical isolation. As an optocoupler, HCLP6000 is chosen because of its very high slew rate that allows us to conduct signals with high frequencies. Schmitt trigger's role is enhance the current value of the system. Since a data signal is more immunized to electrical noises when it is conducted with high current; our schmitt trigger can supply 25mAmps.

CAN Bus is commonly used in our electronic control units. Our mid level controller Renesas M32C microprocessor has an embedded CAN Bus module. Our low level controller Renesas R8C does not have embedded CAN Bus module, so that we applied an external module. Moreover, CAN Modules need to interact with a transceiver to acquire CANH (CAN-High) and CANL (CAN-Low) data lines.

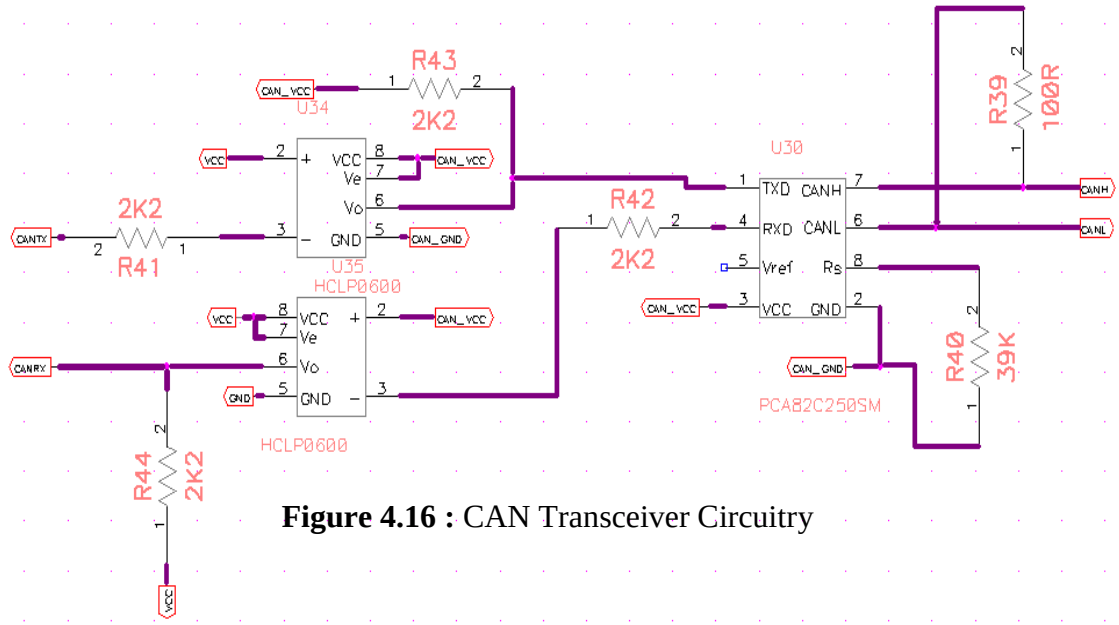

Figure 4.16 : CAN Transceiver Circuitry

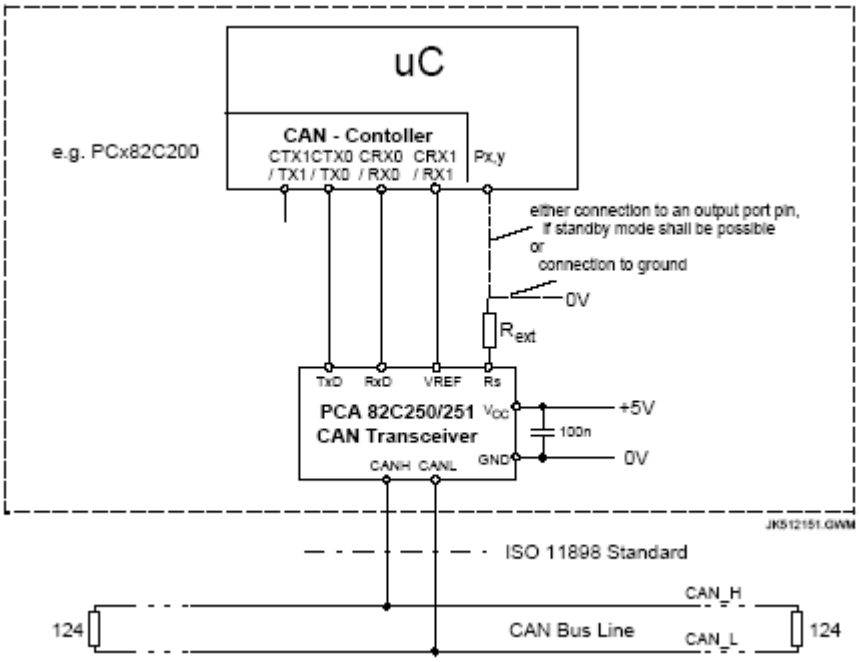

Figure 4.17 : CAN Transceiver Environment

As seen in Figur 5.17, CAN Transceiver Environment must be used to obtain CAN data lines. In this environment, an isolated 5V supply voltage, CAN_VCC and CAN_GND is used. Firstly CAN Controller pins,CANTX and CANRX, are electrically isolated from the transceiver by using optocouplers.

Distributing the CAN data lines, CANH and CANL, whole network is installed. The underground CAN specification is explained in Software Section.

Whole brushless servo motor drive schematic diagram is added to Appendix, complete circuit can be seen in a schematic, called “brushless_drive”.

4.2.3 Logic Design and Sensory Electronics

Having designed a system with three levels of controllers, three electronic control unit is desired. Low level controller is also Joint Driver’s controller, hence, it already has a hardware environment within necessary peripherals.

In spite of that, mid level and high level controllers are employed just for processing data. Because of that, their hardware structure is relative simple and only requires microprocessor peripherals.

In mid level controller, Renesas M32C 3083 microprocessor is employed. Its memory capacity is supported with a 16K EEPROM, FM24C16, which is connected serially through I2C Bus. Also, a supervisor chip, ADM705, is used in order to detect power fail and arrange reset operation.

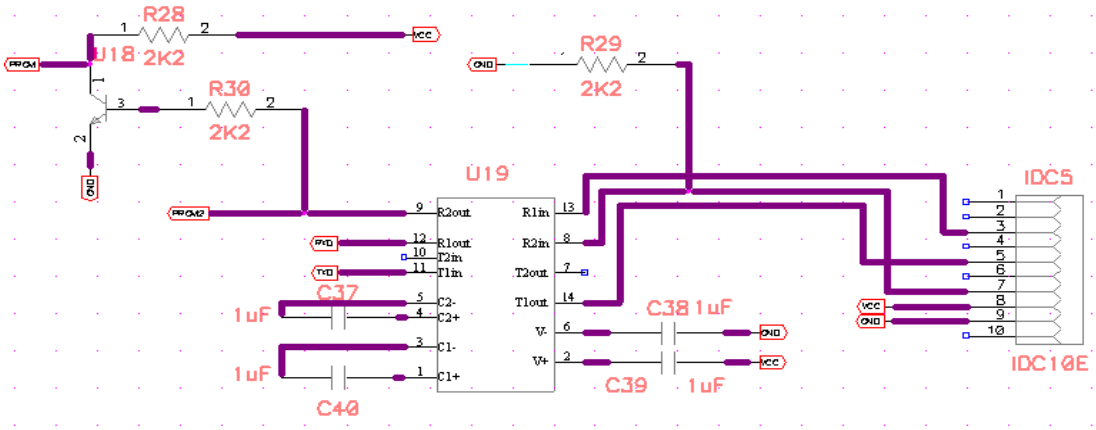

Figure 4.18 : RS232 Interface

The mid level controller switches to boot mode while its software is changed. To create a communication between host PC and M32C microprocessor, a serial line is used through an RS232 interface as shown above, in Figure 5.18

Mid level controller also includes force sensors. Actually force sensors are considered as external sensors, so that, they had to be replaced Sensor Field in Layer-3. Due to some difficulties in cable handling, they hooked up to mid level controller. However, mid level controller does not process force data. It only digitize it through its ADC and directly transmit it digitally to high level controller.

For high level controller, we did not prefer designing a hardware structure, as we used an Evalutaion Development Kit of Renesas SH7047 DSP that is ready to use. It has been configured while manifcaturing, so that, we just used its I/O lines and CAN Bus Lines. Detailed information is added to Appendix about this board.

External sensors are all connected to high level controller, because advanced algortihms and sensor fusion techniques are realized within our DSP. Having explained previously, there is a CAN Bus between external sensors and high level controller .

In figure 5.19, Sensor Field's hardware structure is shown with block diagrams. Following studies on this robot may require additional sensors. Thus, it is required that, any sensors on the field can be plugged-in easily. CAN Bus brings along with this qualification as so many nodes can be freely connected to CAN Network. It is one of the main reason that we used CAN Bus here.

It is assumed that external sensors' outputs are analog. In order to connect them into CAN Network, we need to digitize them. However, it also be intelligent data to be able to interact with CAN Interface. Considering these issues, we used Renesas R8C microprocessor again. This microprocessor includes 10-Bit ADC, in which analog data can be digitized. It also provides intelligency and software configurable data transmission which is required while interacting with the CAN module. Using these system, all external sensors are connected to high level controller through CAN Network, even if their outputs are analog.

Some of the sensors in the field might be put in a sleep mode in case of not using its function. By doing it, power save is more efficiently implemented to our system while feding the system with batteries.

However, after the sleep mode, sensors must have the fastest access to controller units. To realize it CAN Wake Up Interrupt function is chosen. Therefore, all sensors are able to transit between run mode or sleep mode.

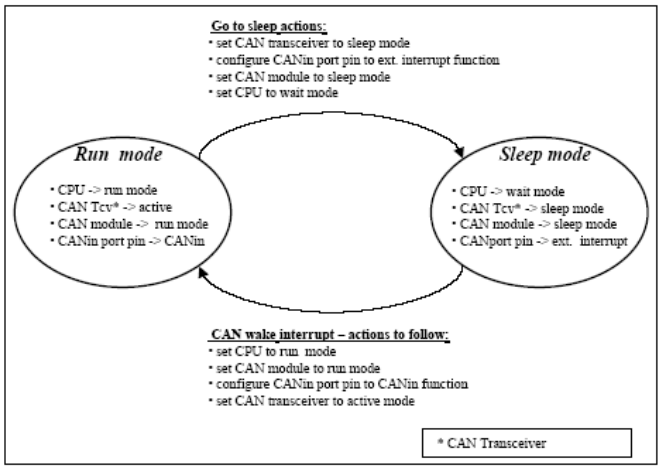

Figure 4.19 : CAN, Transition Scheme

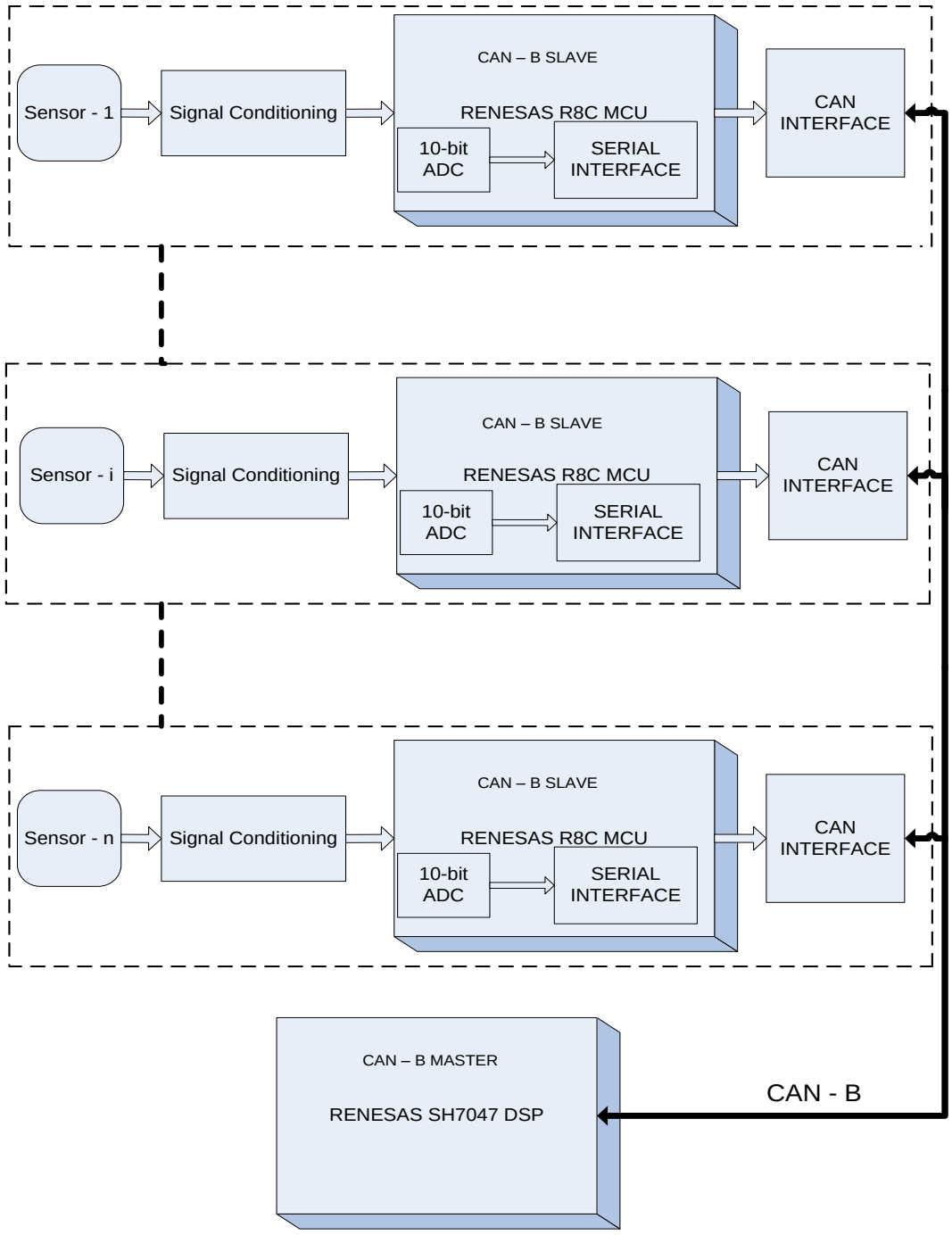

Figure 4.20 : Sensor Field

A general controller circuit is a microprocessor with some peripherals. These peripherals are some specific integrated circuits which asists the microprocessor while realizing the controller scheme.

In Figure 5.20, a common controller block diagram is shown. Every controller unit has similar structure that is displayed below.

Analog Reference Chip, REF02SM is in charge of providing necessary reference voltage for the ADC based operations. It outputs an exact value of +5Volts which enhances the quality of digitizing analog values. Microprocessor Supervisor Chip, ADM705 is arranging reset operations and manages the watchdog. Moreover it detects power failure errors and then recalls the nonmaskable interrupt in which valueable variables are tried to save into EEPROM.

Various memory chips are used in controller circuits. In mid level an EEPROM,FM24C16 is implemented into our system. It communicates with the mid level controller by using I2C bus. In high level controller, RAM and ROM blocks are used.

In order to put microprocessor into boot mode, RS232 interface is applied between the host computer and microprocessor. For internal communication CAN Transciever, 82C250 chips are connected to microprocessors.

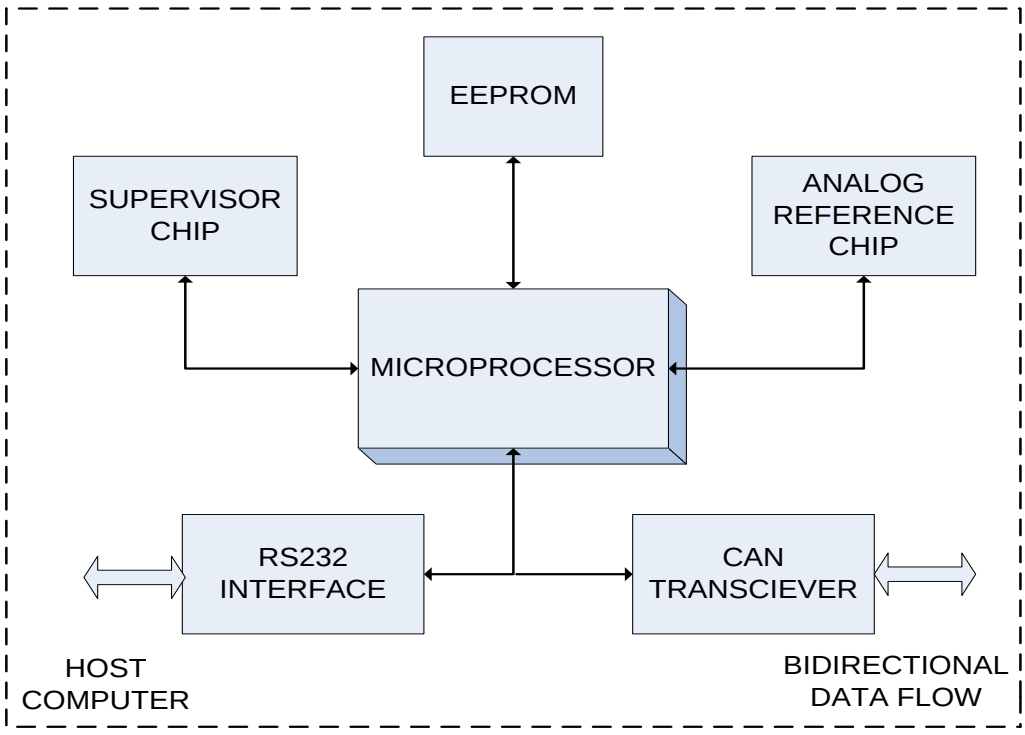

Figure 4.21 : General Controller Circuit

All DC voltage levels have their own dedicated DC regulator. Among them, only Vcc is above 1Amps. All Vcc fed components' current consomption is nearly 1.2Amps. Thus, the power consomption is approximately 6 Watts in the circuits.

In maximum torque; our distal joint's motor consumes 1.2Amp, basalar joint's motor consumes 3.1Amps and thoracic joint's motor consumes 5.7Amps. Therefore, the total power consumption while the whole 24 motors consumes maximum current is above 100Amps. So that, we will fed the system externally with a cable during the tests and experiments on our robot

All circuits are designed all except high level controller circuit. It is an Evaluation Development Kit, EDK7047, which is directly implemented to our system.
Sensor Field's and mid level controller's schematic diagram is added to Appendix. Detailed information can be gathered from this section within complete circuits and their Printed Circuit Board layouts.

5. CONTROL SCHEME AND ALGORITMH DESIGN FOR ACTUATORS

Motor control applications are increasingly gaining importance as they are getting more and more functional. Control features are implementing by using embedded microcontrollers. Complicated calculation algorithms replace expensive sensors for position and velocity. These positive changes and improvements become difficult to maintain the overview about motor application concepts. Especially the principles of controlling the motor via microcontroller require a broad knowledge of motor technology, functionality and electrical mathematics basics.

Since the discovery of the first electric motor many kinds of motors invented and producted. The topologies of these motors are all the same essentially. Meanwhile, the electronics has extended the usage of electrical machines. The progressive entry of embedded systems in Electrical Driving has come along with high efficiency and low cost applications. Machines like DC commutators replaced by Micrcontroller/DSP based control structures. Controlling the speed and torque, electrical machines serve well in real-time applications.

5.1. Permanent Magnet Brushless DC Motor

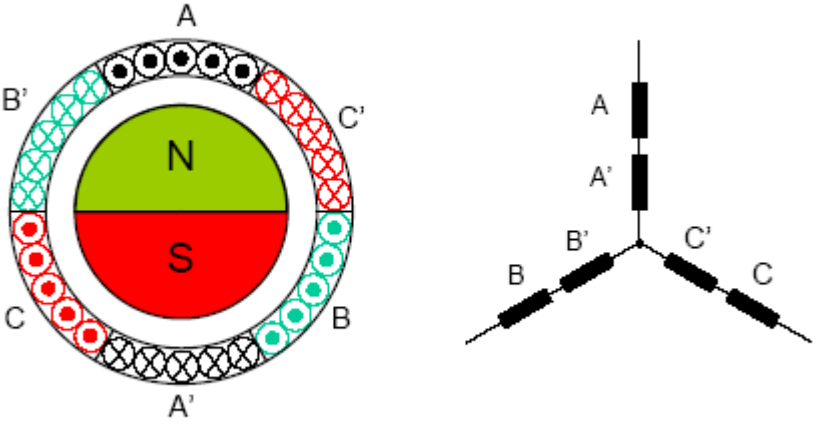

Figure 5.1 : Brushless DC Motor Topology

Brushless DC Motors have similar to AC Induction Motors. Materials, electrical and magnetic circuits are identical. However, in permanent magnet brushless DC motors, instead of having a shorted electrical circuit placed along the rotor axis, the rotor consists of a permanent magnet. The magnet is surrounded by a constant magnetic

field, which creates an interplay with the stator electric field a force which is resulting in a rotation of the rotor. In figure 2.1. the topology is displayed.

The reduced numbers of coils leads to a torque ripple creation while rotating. The maximum torque is obtained when the rotor magnetic field and stator electrical field are aligned in 90^o angle to each other. In order to generate a constant torque, it needs to keep the 90^o angle between stator electrical field and rotor magnetic field. Limting the number of coils to three pairs means that the angle increment is only 60^o. It is actually 360^o, divided by coils. Due to this fact, the stator field is kept at the same position during a 60^o of shaft position. Such constant torque cannot be maintained, but a generation of 90^o by working between 60^o and 120^o can be achived.

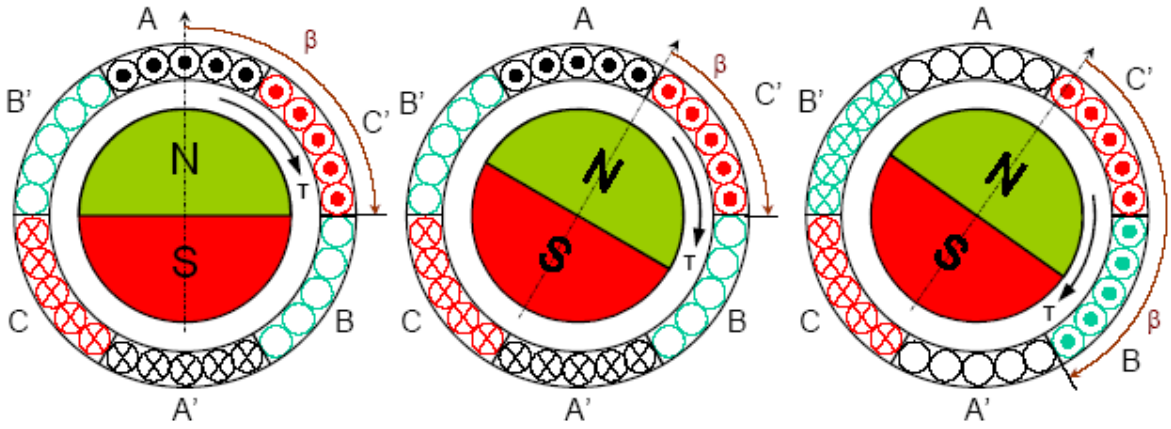

Figure 5.2 : Motor Rotation According to Block Commutations

Lets consider β angle is the stator field with respect to rotor position. When the angle β is decreased to 60^o the stator field advances it to 120^o. Then it stays for the next 60^o angle of the upcoming cycle.

While driving these motors with sine wave, they can be called AC Brushless DC Motors. In our application this method is chosen. It also returns with a sinusoidal Back EMF. The current of phases need to have an appropriate phase displacement of 120^o in case of 3-phase motor. Considering the fact that our motorsa re 3-phase, our hardware structures are able to do it. All phase currents need to be generated that all are in a correct relationship to each other at every rotor position. In order to realize

that, our commutaition detect circuit runs on very high resolution. Due to these configurations, a constant torque is aimed.

5.2. Torque and Position Control of Brushless Motors

As seen in figure 2.3. our motor control scheme is including two different control loops : Torque Control and Position Control. Combining these two methods; a proper motor control system is desired in order to rotate in giving angular displacements. This control algorithm is applied to low level controller unit in joint drives.

Position control loop is receiving a position reference from mid level controller. Mid level controller is performing inverse kinematics, so that, reference position information comes from this layer. Encoders are also providing real time position information from motors. All in all, evaluating these values provides a position control method is implemented to low level controller system.

After a proper PID block, Torque reference is obtained from the position information. Displacement appears as a paramater in Torque function as shown below.

$$T_{ref} = J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} \quad \textbf{(2.1)}$$

$$T_{ref} = (s^2 J + sB) * \theta \quad \textbf{(2.2)}$$

Having extracted the Torque reference for Torque Control Loop, we need to perform Park's Equation in order to have winding variables. This transforms are very sutiable for obtaining winding variables to a coordinate system in which the rotor is stationary. We identify equivalent stator windings in the direct and quadrature axes. The direct axis is equivalent of one of the phase windings, but aligned directly with the field. The quadrature winding is situated, so that, its axis leads the field winding by 90 electrical degrees. The transformation used to map the electrical currents,fluxes and so forth onto the direct and quadrature axes are celebrated Park's Transformation. The mapping takes from ;

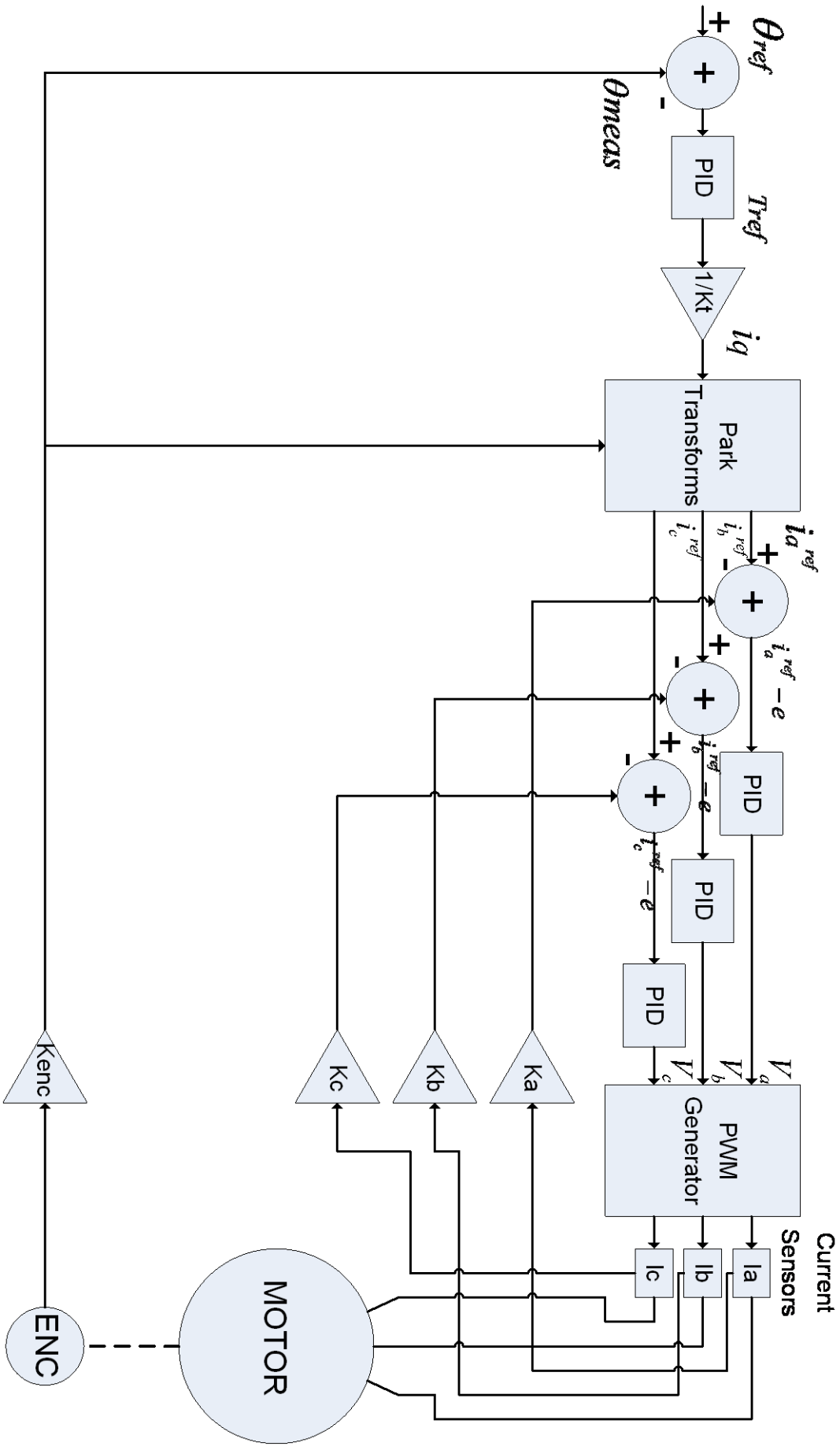

Figure 5.3 : Torque and Position Control Scheme

$$\begin{pmatrix} U_d \\ U_q \\ U_o \end{pmatrix} = \underline{U}_{dq} = \underline{\underline{T}}\underline{U}_{ph} = \underline{\underline{T}} \begin{pmatrix} U_a \\ U_b \\ U_c \end{pmatrix} \quad \textbf{(2.3)}$$

$$\underline{\underline{T}} = \frac{2}{3} \begin{pmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \quad \textbf{(2.4)}$$

$$\underline{\underline{T}}^{-1} = \begin{pmatrix} \cos\theta & -\sin\theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{pmatrix} \quad \textbf{(2.5)}$$

This transformation maps balanced set of phase currents into consistent currents in the d-q frame. That is, if rotor angle is $\theta = \omega \cdot t + \theta_0$,the phase currents are;

$$I_a = \cos\omega \cdot t \quad \textbf{(2.6)}$$

$$I_b = \cos\left(\omega \cdot t - \frac{2\pi}{3}\right) \quad \textbf{(2.7)}$$

$$I_c = \cos\left(\omega \cdot t + \frac{2\pi}{3}\right) \quad \textbf{(2.8)}$$

The transformed set of currents are ;

$$I_d = \cos\theta_0 \quad \textbf{(2.9)}$$

$$I_q = -\sin\theta_0 \quad \textbf{(2.10)}$$

Above Park's transformation is extracted for AC Synchoronous machines. What we will do is configuring this technique for brushless servo motors

In our system this angle has to be kept around 90 electrical degrees as explained previously. Thus, we configure Park's Transformation in respect to our needs. Therefore, I_d current has to approach zero in order to obtain maximum torque. Then, we can obtain current value from the torque.

$$I_d \rightarrow 0 \quad \textbf{(2.11)}$$

$$T_{ref} = K_t \cdot I_q \quad \textbf{(2.12)}$$

After that, we extact current values for each phase as follows :

$$\mathrm{I_m} = \sqrt{\mathrm{I}_d{}^2 + \mathrm{I}_q{}^2} = \mathrm{I}_q \quad \textbf{(2.13)}$$

$$\varphi = \tan^{-1}\theta\left(\frac{I_q}{I_d}\right) \quad \textbf{(2.14)}$$

$$I_a{}^{ref} = \mathrm{I_m} \cdot \sin(\theta \cdot p + \varphi) \quad \textbf{(2.15)}$$

$$I_b{}^{ref} = \mathrm{I_m} \cdot \sin(\theta \cdot p + \varphi - \frac{2\pi}{3}) \quad \textbf{(2.16)}$$

$$I_c{}^{ref} = \mathrm{I_m} \cdot \sin(\theta \cdot p + \varphi + \frac{2\pi}{3}) \quad \textbf{(2.17)}$$

Using the Park Transformation shown above, we have extracted sinusoidal matches of a constant value. Hence, we estimate three phase currents' reference values.

Current sensors are supplying real-time current information as the motor rotates. It gives an analog output, then this anaog output is multiplied with a constant and applied to low level controller. Comparing this reference and measured current values, torque control method is implemented.

The error signal on each phase is the difference of the reference current value and measured current value. Considering the fact that, both of these current values are sinusoidal; the error signal is also sinusoidal. This sinusoidal signal is carrying exactly the same pattern of phase voltages. Each phase voltage might be derived from this error signal after applying a simple PID block.

Sinewave PWM, SWPWM, signals are adjusted with respect to that signal. So that V_a, V_b and V_c values are obtained by using both current and position control methods. As explained before, sinusoidal waveform currents s are applied to motor windings in which, sinusoidal waves are generated in PWM mode.

The phase voltage is chopped at a fixed frequency with variable duty cycle, which depends on the sine wave current. All phases are displayed in 120^o angle to each other as our motors have 3-phase. The switching pattern of a phase requires a non-overlapping time between injecting of two transistors per phase.

Our proposed way to create SWPWM is to create a signal that has a fixed carrier frequency and sinusoidally weighted duty cycle. The implementation is realized by creating look-up table that contains sinusoidally weighted PWM values, which are constantly transferred to the PWM mode.

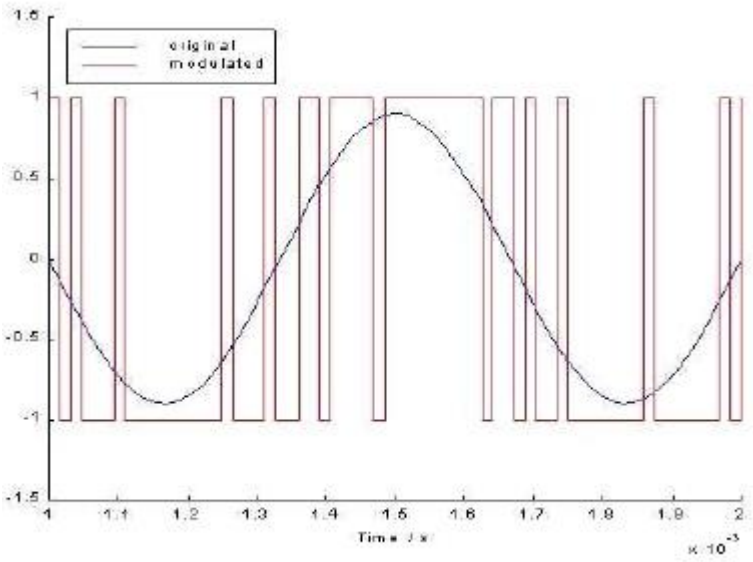

Figure 5.4 : A Sample SWPWM

One of the biggest advantages of SWPWM is that generation of the sine wave is realized by using a fourth of sign and put the values into a look-up table. There is almost no CPU load demand. Another possibility is the calculation in algebric mode, however it would be a burden to our controller. Thus we choose to generate it by using look-up tables.

5.3. Motor Control Test Algorithms

All embedded control application requires specific algorithms that defines the pathway of implementation. However, before realizing control implementation, a series of test algorithms must be processed.

Test algorithm is always started by the high level controller. High level controller firstly tests communication interfaces such as CAN bus and serial ports. If it detects an error, it defines a communication error.

Having tested communication interfaces, the high level controller tries to reach sensor in sensor field. If the high level controller's test signals cannot be responded by its peripherals, it defines a peripheral error.

After that operation, the high level controller submits a request to each mid level controller for data exchange via serial port. If this mutual handshake is not achived, the high level controller defines a mid level controller error.

After each error detection, the high level controller suspends itself and pass into sleep mode. At this level, an external intervention is required.

If the high level controller unit detects no error, the system passes the high level control test algorithm which also triggers mid level control algorithm.

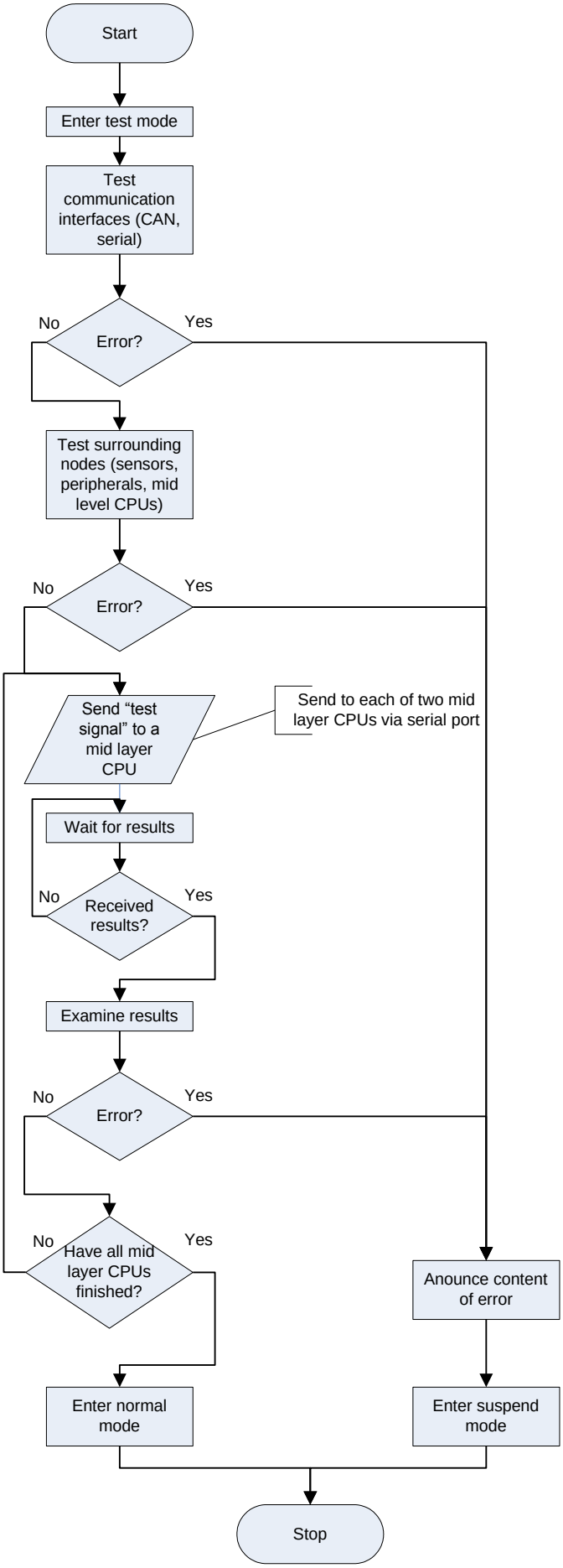

Figure 5.5: High Level Control Test Algorithm

Mid level control test algorithm is also similar to those of high level control test algorithm. The test starts with controlling the CAN interface. If the CAN Interface responds negatively, mid level controller generates a communication error.

If there is no communication error on this level, mid level controller unit detects all joint drivers. In case of detecting an error, the mid level controller defines a drive error. There are 12 drive errors, so that mid level controller flashes the i^{th} LED if the i^{th} joint driver is not responding during the test algorithm processes.

In case of detecting errors, the mid level controller suspends itself and waits for the external intervention to eliminate errors. While no error case is obtained, it triggers the low level controller test algorithm respectively.

Each low level controller unit executes their test algorithms in order. Firstly low level controller tries to turn the motor for θ degrees. As a result of this action, it expects encoder signals. If there are no encoder signals read, it defines a motor error.

Having tested all encoders respectively, low level controller is the las part of the test algorithm. As expressed above, in case of detecting errors, the low level controller passes to sleep mode and waits for the external intervention to restart. If there are no errrors detected, the whole control system justifies that there are no hardware based errors.

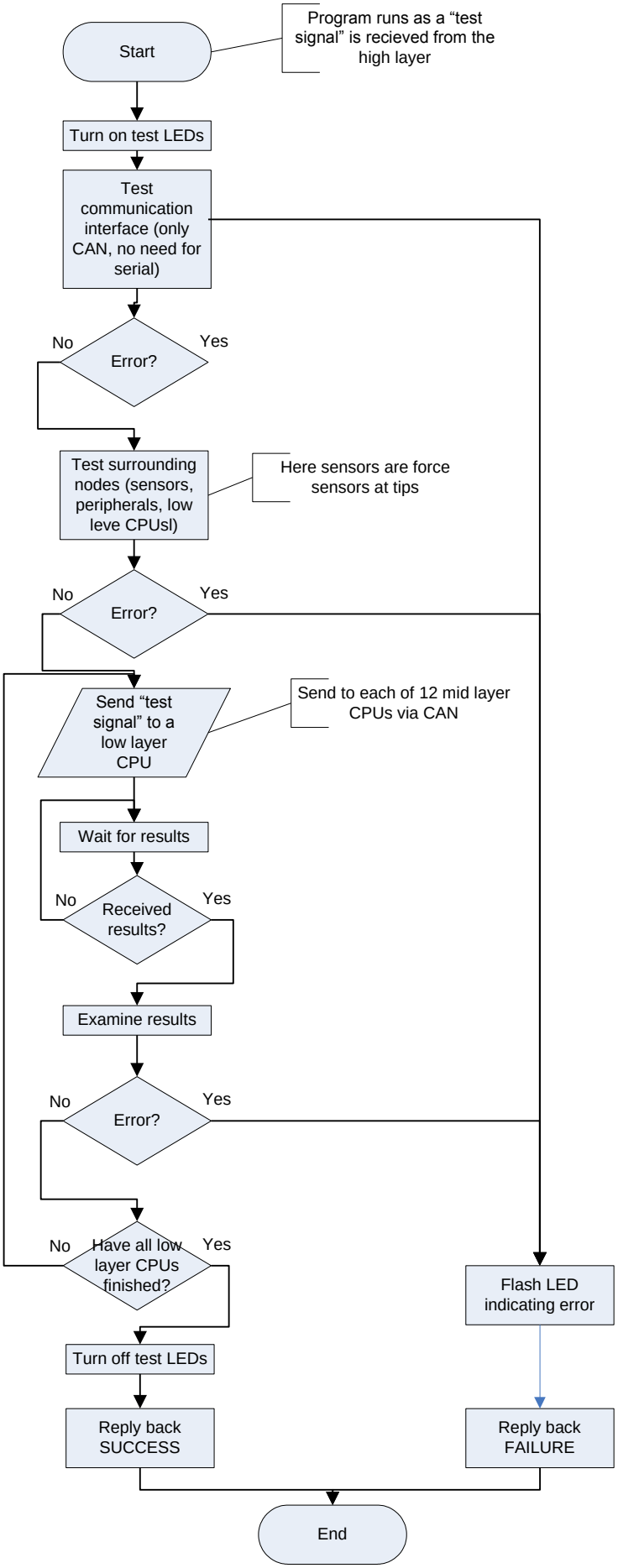

Figure 5.6: Mid Level Control Test Algorithm

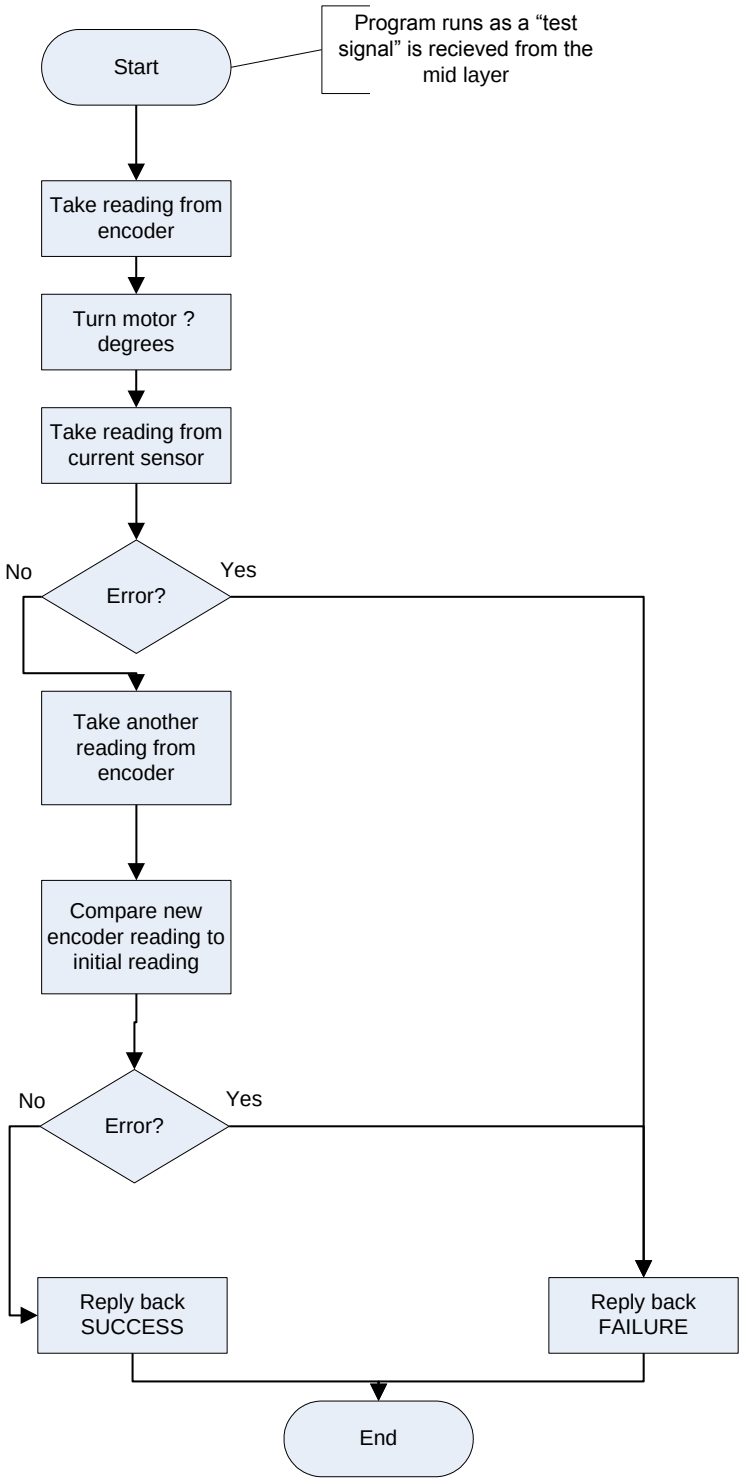

Figure 5.7: Low Level Control Test Algorithm

6.CONCLUSION

In this study, all designs and methodologies are aimed at understanding a better comprehension about biomimetric approach on mobile robots. Therefore, more and more optimized robotic system may be designed as we look into the nature more deeply.

Having obtained a mechanical frame that appears as a skeleton of the robot; hardware realizations provide moving capability and physical basis for software applications. Realizing real time control algorithms by using these physical basis; the robot can achive goals as it gains the intelligency.

CAN Bus implementation is makes the hardware structrue quite flexibile. Especially for addressing and slave adding. For example, for further applications, a different type of sensor might be added and another one can be removed without reconfiguring the electronic circuits.

Biomimetric approach on robotics allows designers to go through right direction over a shortcut as the nature herself has been iterating from the beginning of the universe. Many creatures are now has the best suit for movements. So that, any robotic application may take place on every kind of situation. Examining the nature makes it easier.

The main purpose of the thesis is hardware realization. In order to do that, PCB files are created and schematic files are designed. All these engineering works make the robot testable in real-time. All hardware is designed by using surface mount components as we have limited place on the robot.

Further studies on this robot will high possibly involved in software applications. Up to date, physical basis for the robot has gained, both mechanically and electrically. From now on many control application based on embedded software could be added to system in order to gain intelligency

REFERENCES

[1]**Beaty, W. and Kirtley J.,** 1998. Electric Motor Handbook, Matbaası, McGraw-Hill Publishing.

[2]**Hanselman D.,** 2003.Permanent Magnet Brushless Motor Design.The Writer's Collective, Rhode Island.

[3]**Uzundere, M.**, 2006. Mechanical Design and Dynamic Analysis of an Eight Legged Mobile Robot, *Master Thesis,* I.T.U. Institute of Science and Technology, Istanbul. (being submitted)

[4]**Mitsubishi Electric**, 2000. CAN Interface Operation, Japan

[5]**Renesas Semiconductor**, 2004. Motor Control with Renesas Microcontrollers, Japan

[6] **Tak Y., Jeong Y., Kim, B**., A Ciliary Based Walking 8-Legged Robot Using IPMC Actuators. Proceedings of the 2003 IEEE Intermtiond Conference on RoboticS &Automation Taipei, Taiwan, September 14-19, 2003

[7] **Xiao J., Minor M., Tummala, K**., Modelling and Control of an Underactuated Crawle Robot. Proceedings of the 2003 IEEE Intermtiond Conference on RoboticS &Automation Maui, Hawaii,USA November 03-19, 2001

[8] **Masakado S., Ishii T., Ishii, K**., A Gait-Transition Method for a Quadruped Walking Robot. Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, California, USA, 24-28 July, 2005.

[9] **Brandon L., Quinn R**., A Robot with Cockroach Inspired Actuation and Control. Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, California, USA, 24-28 July, 2005.

[10] **Klaassen B., Kirchner B**., Biomimetic Walking Robot Scorpion: Control and Modeling. Robotics and Autonomous Systems Journal, USA, 2005

[11] **Bowerman R. F.** (1975) The Control of Walking in the Scorpion I. Leg Movements during Normal Walking. J comp Physiol 100:183196

[12] **Summer M., Davis L**., Design and Control of a Dexterous Biomimetic Multi-Limbed Walking Robot. Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, California, USA, 24-28 July, 2005.

[13] **Massari M., Nebuloni S**., Realization and Control of a Prototype of Legged Rover for PlanetaryExploration Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, California, USA, 24-28 July, 2005.

[14] **Gökaşan, M.,** 2006. Personal meeting.

[15] **Atılgan, B.,** 2005. Personal meeting.

[15] **Kırman, H.,** 2005. Personal meeting.

PROFILE

Name : R. Barkan UĞURLU
Birth Date / Place : 08.05.1981 / Çankırı

Education

2004 - Istanbul Technical University
Institute of Science and Technology
Department of Mechatronics Engineering

1999 – 2004 Yildiz Technical University
Electrical and Electronics Faculty
Department of Electrical Engineering

1998 – 1999 Uskudar Burhan Felek High School
1995 – 1998 Haydarpasa Technical High School

Professional Interests : Microprocessor/DSP Based Embbedded Hardware Design, Artificial Intelligence, Discrete Time Control, Robotics, Biomimetics

Future Work : Biped Robots

APPENDIX

All electronic schematics and their placements on PCBs (with routings) are included in this section