Robocon 2015
Electrical Team Documentation

Robotics Club, IIT Delhi

Abstract

Robocon is an Asian undergraduate level robotics competition, held annually with the aim of advancing engineering technologies in the region. Each year, the host country comes up with a problem statement in which the participating teams are asked to build two or more robots, either manual or automatic, which have to perform a specified task under a given set of time, weight and dimensional constraints.

The national round of Robocon is held every year in Pune and is organised by Doordarshan. The best engineering institutes from across the country compete with each other for the honour of representing India at the international level of the competition.

The theme for Robocon 2015 was 'Robominton – Badminton Robo-Game'. The problem statement was to manufacture two robots which had to play a doubles game of badminton against two robots of the opposite team. The first team to score five points would win the game. The robots could be either manual or automatic and should be capable of playing in a standard badminton court with standard rackets and shuttles. They had to be limited by a collection of mechanical constraints in dimensions and weight, as established by the competition rules.

The Electrical team was faced with challenge of implementing new subsystems such as the holonomic drive, voltage regulation, wireless communication as well as integrating these systems with the new mechanical framework. The main focus of the team was to come up with systems which were more robust and reliable. The control algorithms were improved to make it easier for the operator to drive the robot. Previous circuit designs were refined to make them more efficient. Extensive research was also carried out in computer vision techniques, ultrasonic sensors and laser sensors which could prove to be useful in the years to come.

This document contains the detailed description of all the subsystems used, their subsequent integration as well as the other research work carried out in fabricating the two robots for Robocon 2015.

Contents

Chapter 1

Overview

The electrical team of the Robocon 2015 was responsible for designing, building and implementing all the electrical circuits used in the two robots, in addition to writing the software running on the onboard microcontrollers. This year, importance was also laid on refining the old systems by using new components. These include smaller solenoids, different connectors, better motor drivers and wireless remote controllers.

1.1 The Team

The electrical team comprised of one 3rd year student and seven 2nd year students.
Team Members:

1. Nupur Kumari (Coordinator)
2. Animek Sahu
3. Ayush Shukla
4. Pranjal Maheshwari
5. Rishabjit Singh
6. Vaibhav Gupta
7. Varan Gupta
8. Yash Kumar Bansal

Our faculty advisers were Prof. Sunil Jha and Mr. Dharmendra Jaitly.

1.2 Team Management

1.2.1 Meetings

Meetings were held twice a week during the initial design phase. In the first meeting after the release of the problem statement, the members were grouped into four teams. Each team had to come up with its own design for the two robots. These designs were improved upon in each subsequent meeting. Feasibility of the designs was verified by performing the requisite calculations and analysis. After two months of brainstorming, the final design was formulated by combining the best elements from each team's design.

Once the final design was conceptualised, meetings were held more regularly, almost on a daily basis. In these meetings, our work was focused on arranging the required electrical components and designing the circuits while the efforts of the mechanical team were concentrated on fabricating the design.

1.2.2 Discussion Group and Email

A Google group was made on which all the important announcements were posted which reached each member through Gmail. In each meeting, a different member was selected who had to post the minutes of that meeting on the group. Discussions on the practicality of various designs were also held in this group.

1.2.3 Cloud Storage

An online folder was maintained on Google Drive where all the circuit designs, Arduino codes and datasheets of the various components were available for the group members.

1.2.4 Inventory

One member of the team was assigned to maintain the inventory of all the components available in the club. The inventory was an online spreadsheet stored on the Google Drive which contained a list of all the parts available along with their quantities.

1.2.5 Practice

From the beginning, it was clear that this year's competition required a lot of practice for the operators. Practice sessions were held regularly every night from 11 P.M. - 3 A.M. at the badminton court of the Student Activity Center. These sessions started immediately after the completion of manufacturing part in the beginning of January and continued till the end of February. A specific document was maintained to record all necessary the data during these test runs. This included :- time taken for each service, the number of returns, time taken to set up the next serve etc. By analysing the recorded data, mistakes were pointed out to the operators and strategies were devised for use in the competition.

1.3 Electrical Requirement for the robots

- The potential difference between any two points on the robot must be less than 24 Volts
- Only laser beams classified less dangerous than Class 2 lasers can be used
- An emergency switch must be present in both of the robots
- Fuses or circuit breakers have to be used in order to prevent damage in case of a short circuit
- Only one of the robots can be controlled using a wired controller
- Method of wireless communication is limited to only Bluetooth, IR, sound wave and visible radiation.
- Wireless communication shouldn't cause any interference with the robots of the other teams

1.4 Subsystems Approach

The problem statement of Robocon 2015 demanded the integration of multiple systems. These subsystems were identified to be the following:

- Drive
- Shuttle hitting mechanism

- Shuttle dropping mechanism
- Remote controller for Manual operation
- Shuttle detection for automatic operation

The members were each allotted one of the above subsytems and were tasked with implementing that particular subsystem.

1.4.1 Drive

A 3-wheel holonomic drive was selected because it removed any kinematic restrictions in the 2D plane and gave the robot the necessary freedom to move in any direction in the badminton court. In addition to being quick enough to move and intercept the shuttle, the drive system also needed to be agile and able to change its direction swiftly. For the drive, Maxon Motors were used in the Front bot while Banebots were used in the Back bot.

1.4.2 Shuttle Hitting

In order to ensure that the shuttle travelled the required distance, it was proposed that the hitting mechanism be quick with minimal delay. Two options were considered for the actuation, motors or pneumatic cylinders. The latter option was chosen since it provided the same performance at a much lower cost.

1.4.3 Shuttle Dropping

For dropping the shuttle, a mechanism similar to the one employed in the barrel of a gun was utilised. The barrel contained seven compartments, six of which were loaded with shuttles. The barrel was rotated using a high torque motor and ultrasonic sensors were used to detect the dropped shuttle.

1.4.4 Remote Controller

As it was necessary for one of the robots to be controlled wirelessly, it was decided that Sony DualShock 3 controllers would be used for this purpose. These controllers operated on Bluetooth 2.0 interface.

1.4.5 Detection

In order to the decrease the dependence on the rider, several methods were tried to detect the shuttle for automatic actuation of the rackets. These included - using a 2D array of lasers, using multiple ultrasonic sensors or using image processing with the help of a camera. Though a certain level of detection was achieved using lasers and ultrasonic sensors, the system was not reliable enough to be implemented for the competition.

Chapter 2

Voltage Regulation

2.1 Introduction

One of the problems faced by last year's team was overheating of the Maxon motors used in the drive. The same motors were used in this year's front bot. These motors have a nominal voltage rating of 18V. The overheating was attributed to the fact that last year, the motors were operated using two batteries (each of 12.6V) connected in series, which made the total voltage at the motors to be 25.2V. Thus, this time it was necessary to make a circuit that would step down the voltage to 18V. Besides decreasing the voltage from 25.6V to 18V voltage, the circuit also acted as a voltage regulator (constant voltage source) providing 18V even if the batteries got drained to 22-23V or even 20V.

The main challenge was to come up with a circuit which capable of handling a very large amount of current while still maintaining the voltage level and overall efficiency. The continuous current that a maxon motor consumes is around 5A. Since we were using three such motors, the circuit should endure not only the continuous current that each of these motors consume but also the spikes in the current arising due to motor acceleration and deceleration. By assuming that the maximum amount of current that a motor would take is double its continuous value, it was established that each of the individual motor circuits should be able to withstand a maximum current of 10A and thus the voltage regulator should be able to withstand around 25A. But, we did not find any IC that can sustain a current of more than 25A.

The subsequent sections contain the description of the two methods that we attempted to make the voltage regulator. The latter one was implemented.

2.2 LM723 with transistor

The LM723 is a monolithic integrated programmable voltage regulator, assembled in 14-lead dual in-line plastic package. The circuit provides internal current limiting. Following are its characteristics

- Input voltage up to 40 V
- Output voltage adjustable from 2 to 37 V
- Output current to 150 mA without external pass transistor

It is a very cheap but reliable IC and is used widely in commercial products such as the constant voltage sources kept in our labs. As this IC can handle only 150mA of current, an external transistor is required to increase its current carrying capacity.

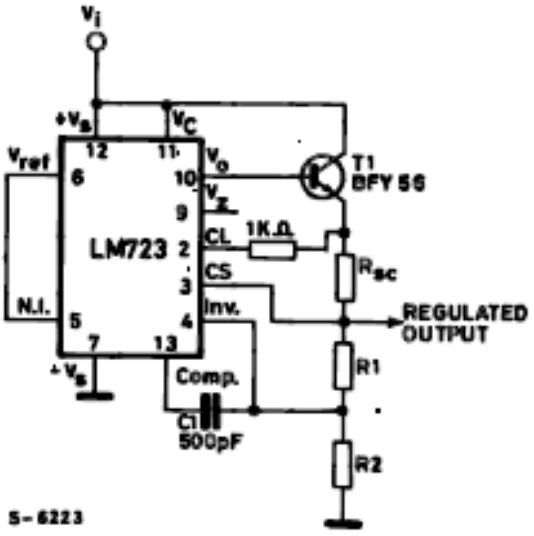

Figure 2.1: Positive voltage regulator (external NPN pass transistor)

For the purpose of testing, 'SL100' transistor was used and the following characteristics were obtained:

Supply Voltage: 25.6V, R_{sc} = 1Ω - 3Ω

R1(Ω)	R2(Ω)	P1	Added(Ω)	R3(Ω)	Load(Ω)	$_{out}$ (V)
7.5k	3.87k	0	1k	2.2k	0	21
7.5k	3.87k	0	1k	2.2k	66	20.8
7.5k	3.87k	0	1k	0	0	21
7.5k	3.87k	0	1k	0	66	20.8

Table 2.1

Testing was later carried out using the '2N6547' power transistor which has a current rating of 15A.

Supply voltage : 25.6V-22V

R(Ω)	V_{out} (V)	I(A)
0	19.7	0
33	18.9	0.62
18	18.6	1.22
12.8	18.6	1.72
10.7	18.6	2.29
8.2	18.6	2.74

Table 2.2

It was found that the output voltage was independent of input voltage but was dependent on the current output. Therefore, as the output current increased, the output voltage decreased.

This circuit was later dropped because neither were we able to solve the problem of the large decrease in voltage with increase in current nor did we find any transistor with a higher current rating.

2.3 LT1083 in parallel

The LT1083 is a positive adjustable voltage regulator that provides current of 7.5A. It is designed to operate down to 1V input-to-output differential (i.e. diff. b/w input and output voltage) and the dropout voltage is fully specified as a function of load current. Dropout is guaranteed at a maximum of 1.5V at maximum output current, decreasing at lower load currents.

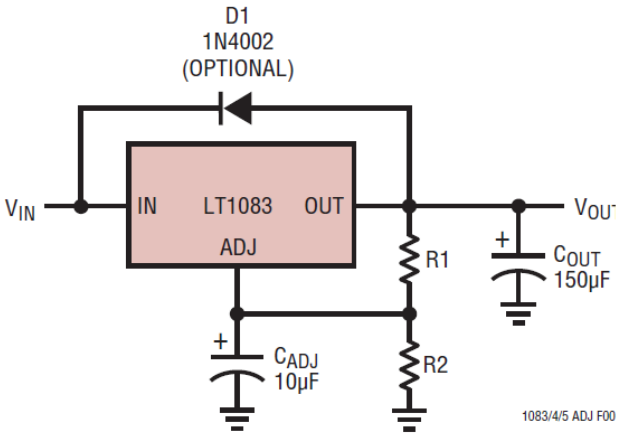

Figure 2.2: LT1083 circuit

Unlike NPN regulators, where up to 10% of the output current is wasted as quiescent current, the LT1083 quiescent current flows into the load, increasing the efficiency.

The advantage of using LT1083 is that a number of these ICs can be connected in parallel. As a consequence, three of these ICs were used in parallel which increased the total current carrying capacity of the circuit to about 22.5A.

Chapter 3

Shuttle Detection

3.1 Ultrasonic Sensors

The idea was to use ultrasonic sensors (SR04) to detect the shuttle as it came near any of the rackets.

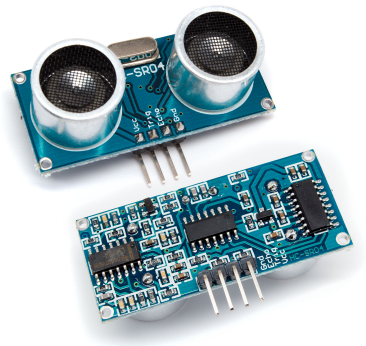

Figure 3.1: SR04 Ultrasonic Range finders

The problem with this sensor was that the delay between successive readings was 30ms. Furthermore, since the sensor works on the principle of reflection of ultrasonic waves, shuttles were not being detected because of their spherical shape.

3.1.1 Removing 'pulseIn()' function

The most important task was to reduce the delay and make the code faster. The basic code available on the internet uses 'pulseIn()' function which is slow.

```
duration = pulseIn(echoPin1, HIGH,5000); // 5 millisecond time-
out
```

was replaced by:

```
while(!digitalReadFast(echoPin1)); //Wait for sensor to send w-
ave and make echo pin HIGH
a=micros(); // Start timer
while(digitalReadFast(echoPin1)&&(micros() - a) < 5000); // Wa-
it for reflected wave
duration = micros() - a;
```

This change improved the readings but the delay of 30ms was still present.

3.1.2 Using multiple sensors

The delay was removed by using more than one sensors at a time. Four such sensors were used in succession with a gap of 4ms between them (4ms*4 + 5ms*3 = 31ms). The 5ms term is the timeout of each of the sensors.

Given below are the top and bottom layer designs of the ultrasonic PCBs, which were placed above each of the rackets. Though, they contain pins for six sensors, only four of them were used initially.

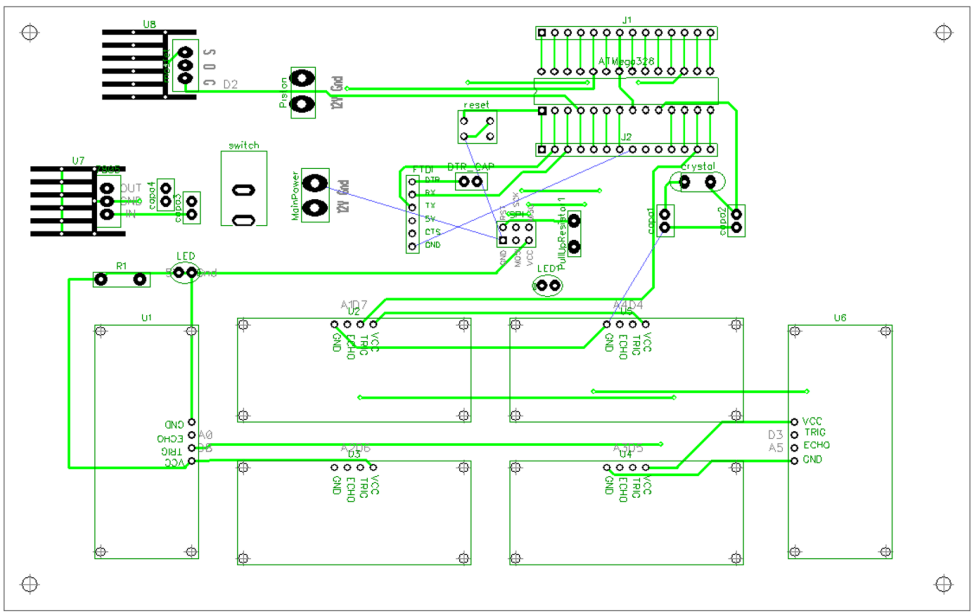

Figure 3.2: Ultrasonic PCB - Top

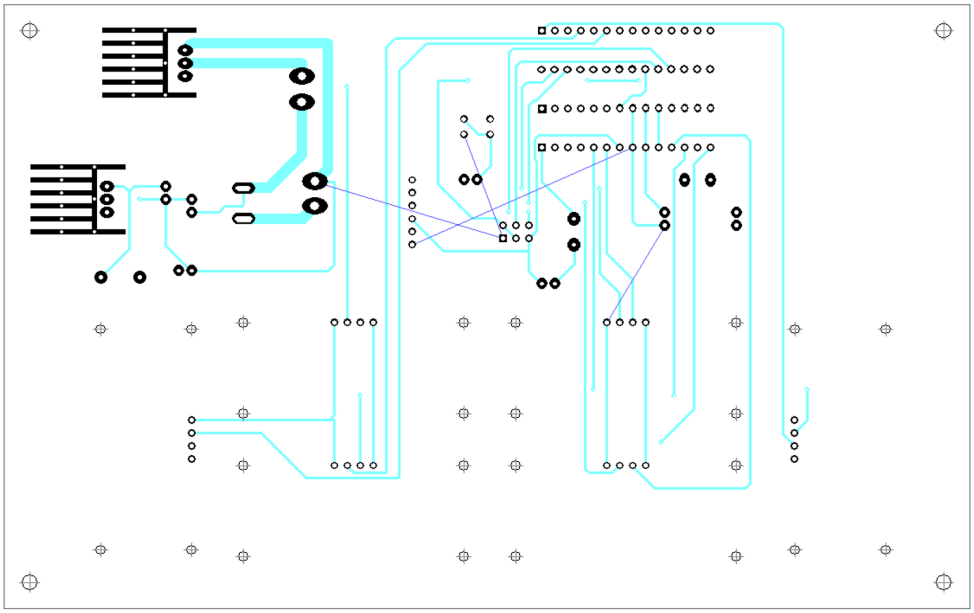

Figure 3.3: Ultrasonic PCB - Bottom

While this removed the problem of the delay, some shuttles were still not being detected because of the shape of the shuttlecock. We could only achieve an accuracy of 10 out of 20 shuttles.

Later, the two remaining sensors were added making the total to six sensors. Various combination of these sensors were attempted - using each of them in succession, using two at time , three at time etc. The maximum accuracy that could be achieved while the robots were still stationary was 18 out of 20 shuttles which still wasn't enough.

In order to increase the accuracy, the receiving area had to be increased.

3.1.3 Custom Ultrasonic module

It was established that the best option to increase the accuracy was to make an ultrasonic sensor module itself which had a larger receiving area, instead of using the commercial ones.

Transmitter

The ultrasonic transmitter generated a sound wave of frequency 40kHz. In order to achieve this, a square wave of frequency 40kHz and peak to peak

R1=R2=220Ω, R3=390Ω, R4=10kΩ, R5=R6=1.22Ω

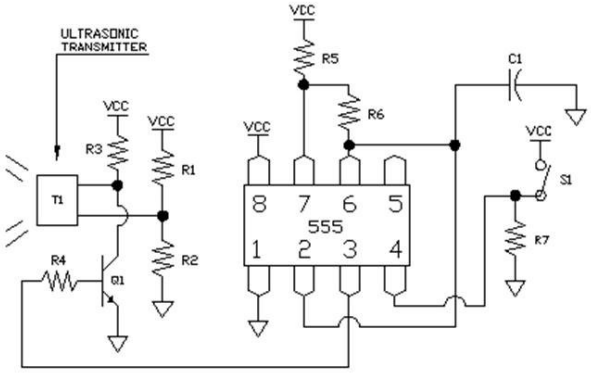

Figure 3.4: Transmitter Circuit

voltage 12v was required. This wave could be generated using a 555 timer IC .

The duty cycle required was 1/3 which is less than 50%. So, instead of the 555 timer, 4047 IC was used in the end.

Receiver

The Ultrasonic receiver generated a sine wave having frequency of 40kHz. The sine wave had a very low peak to peak voltage and was susceptible to noise. Thus, it was imperative for the signal to be amplified and the noise removed. This was achieved using the following circuit.

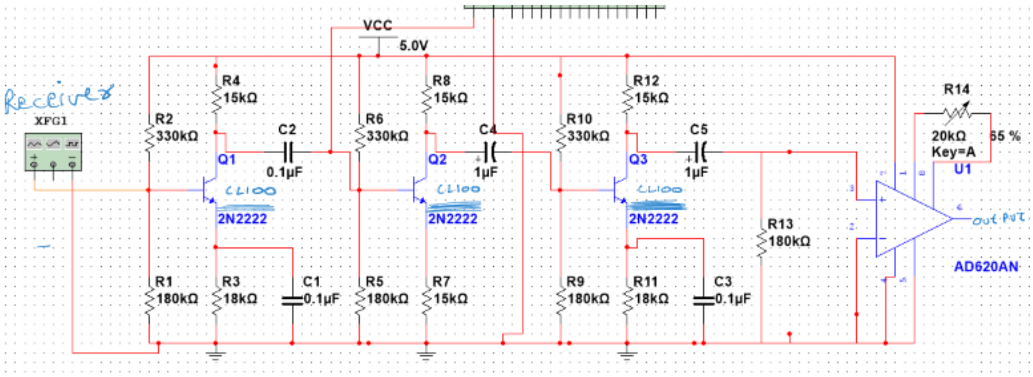

Figure 3.5: Receiver Circuit

In the the transmitter circuit, three transmitters were used in parallel. Similarly, three receivers were also used.

The accuracy increased but since this circuit was completed only around 10-12 days before the competition, extensive testing could not be done. Hence, the circuit was discarded and not used during the competition.

Chapter 4

Wireless Control

4.1 Introduction

As per the rules, it was necessary for one of the robots to be operated wirelessly. Wireless communication was restricted to only :-

- Bluetooth (after version 2.0)
- IR rays
- Sound/sonic waves
- Visible radiation

After much brainstorming, it was decided that both the robots would be operated wirelessly using Sony DualShock 3 controllers. Wireless communication was favoured because wired controllers would have required the use of long wires which would provide needless obstruction to the rider and also had the risk of getting stuck in the wheels of the robot.

DualShock 3 (DS3) controllers were chosen because of the team's prior experience with the similar DualShock 2 controllers. DS3 controllers operate on Bluetooth 2.0 interface which provided the least latency among all the wireless options available while still ensuring a working range of around 10 metres. They also had the advantage of having analog triggers which were utilized to control the rotation of the robots.

Though the robots were operated wirelessly, a wired Xbox 360 controller was still kept as a backup in case things went wrong.

At this point, it is important to mention that there are a lot of fake DS3 clones available in the market at cheap prices. These controllers, which look exactly the same as the original ones, tend to have analog sticks and triggers which are less sensitive to movement.

(a) Incompatible dongles

(b) Kinivo BTD

Figure 4.1: Bluetooth Adaptors

4.2 Bluetooth Adaptors

The team used standard USB Bluetooth adapters aviailable in the market for the wireless communication at the microcontroller end. Interfacing DS3 controllers with an Arduino using Bluetooth adaptors was well documented and the Arduino libraries for it were readily available. Hence, for the sake of convenience, these were chosen instead of bluetooth pin modules.

One problem faced by the team using bluetooth adaptors was that most of the adaptors available in the market were made by Chinese companies and these adaptors were either incompatible with the Arduino or had the same MAC address. The problem with having the same MAC address on both the robots was that there was interference during the wireless communication.

In the end, Kinivo BTD-400 Bluetooth 4.0 dongles were used, which were free from such defects.

Figure 4.2: DualShock 3 Controller

4.3 Serial Communication

Serial communication with the DS3 controller was achieved using the USB Host Shield 2.0 library for Arduino. The MAC address of the bluetooth adaptor was first stored in DS3 controller by connecting it to the Arduino using a mini-USB cable. Pairing was achieved subsequently by pressing the PlayStation button on the controller. Once the controller was paired with the Arduino, the data from the analog sticks and various buttons was easily accessible using the functions available in the USB Host Shield library.

Chapter 5

Wire Selection

5.1 Introduction

This year we decided to use BaneBot motors in the drive which had much higher current rating then any of the motors previously used in the club. So, it was decided to do all the necessary calculations for the wires and choose them according to need. Since the motors used in both the bots were different and had quite significant difference in their current ratings, we decided to have separate analysis for both bots.

5.2 Front Bot

Front bot had following devices:

- 3 Maxon Motors
- 1 Robokit Motor
- Arduino Mega ADK
- Sensors
  - Ultrasonic Sensors for detection
  - IMU
  - Current Sensor
  - Pneumatic Pressure Sensor

S. No.	Connection end points		Optimum Current Rating(A)	Designed Current Rating(A)	Wire Size (AWG)
1	Battery (24V)	Power Board	$6 \times 3 = 18$	30	14
2	Battery (12V)	Power Board	$8 \times 1 + Arduino + Sensors \approx 12$	20	16
3	Power Board	Sabertooth motor driver (Maxon)	$6 \times 2 = 12$	20	16
4	Sabertooth motor driver (Maxon)	Maxon Motor	$6 \times 1 = 6$	10	16
5	Power Board	Sabertooth motor driver (Robokit)	$6 \times 2 = 12$	20	16
6	Sabertooth motor driver (Robokit)	Robokit Motor	$8 \times 1 = 8$	12	16

Table 5.1: Wire Selection for Front Bot

5.3 Back Bot

Back bot had following devices:

- 3 BaneBot Motors
- Arduino Mega ADK
- Sensors
  - Ultrasonic Sensors for detection
  - IMU
  - Current Sensor
  - Pneumatic Pressure Sensor

S. No.	Connection end points		Optimum Current Rating(A)	Designed Current Rating(A)	Wire Size (AWG)
1	Battery (Motors)	Power Board	$11 \times 3 = 33$	50	12
2	Battery (Arduino)	Power Board	$Arduino + Sensors \approx 5$	10	16
3	Power Board	Sabertooth motor driver	$11 \times 2 = 22$	30	14
4	Sabertooth motor driver	BaneBot Motor	$11 \times 1 = 11$	20	16

Table 5.2: Wire Selection for Back Bot

5.4 Guideline for Wire Selection

The length of the wire include the wire length from one end point to other and back from second end point to first end point. This table is for 12V circuit, so to use it with 24V, we convert our current rating to the table's current rating keeping the power($V \times I$) through the wire same.

Amps @12 Volts	LENGTH OF WIRE (Centimeters) American Wire Gauge (AWG)						
	100	150	200	300	450	600	750
0 to 1	18	18	18	18	18	18	18
1.5	18	18	18	18	18	18	18
2	18	18	18	18	18	18	18
3	18	18	18	18	18	18	18
4	18	18	18	18	18	18	18
5	18	18	18	18	18	18	18
6	18	18	18	18	18	18	16
7	18	18	18	18	18	18	16
8	18	18	18	18	18	16	16
10	18	18	18	18	16	16	14
11	18	18	18	18	16	16	14
12	18	18	18	18	16	16	14
15	18	18	18	18	14	14	12
18	18	18	16	16	14	14	12
20	18	18	16	16	14	12	10
22	18	18	16	14	12	12	10
24	18	18	16	14	12	12	10
30	18	16	14	12	10	10	10
36	16	14	14	12	10	10	10
40	16	14	12	12	10	10	8
50	16	14	12	10	10	10	8
100	12	12	10	10	6	6	4
150	10	10	8	8	4	4	2
200	10	8	8	6	4	4	2

Figure 5.1: Wire Selection Criteria

Chapter 6

PCB Designing

6.1 Introduction

In the last few years, we had quite a problem with the our designed PCBs. We faced quite a bit of problems from overcurrent and excessive heating of the boards due to improper design. This year we decided to use IPC-2221 guideline to design our circuit boards.

6.2 Power Board

Power Board were the main focus of this designing phase as in previous few years almost all of the power boards which were printed had some fault or other in them.

6.2.1 Front Bot

It had maximum design current of 18A passing through it from battery to the sabertooth driving the maxon motors. We had the copper thickness of 0.1mm and we wanted maximum temperature rise of $10°C$ at the ambient condition of $40°C$.

S. No.	Connection end points	Current Rating(A)	Required Trace Width(mm)	Designed Trace Width(mm)
1	24V line	$6 \times 3 = 18$	4.43	6
2	12V line	$8 \times 1 + Arduino + Sensors \approx 12$	2.53	4
3	Ground	$18A$	4.43	6 + Copper Pour

Table 6.1: Power Board for Front Bot

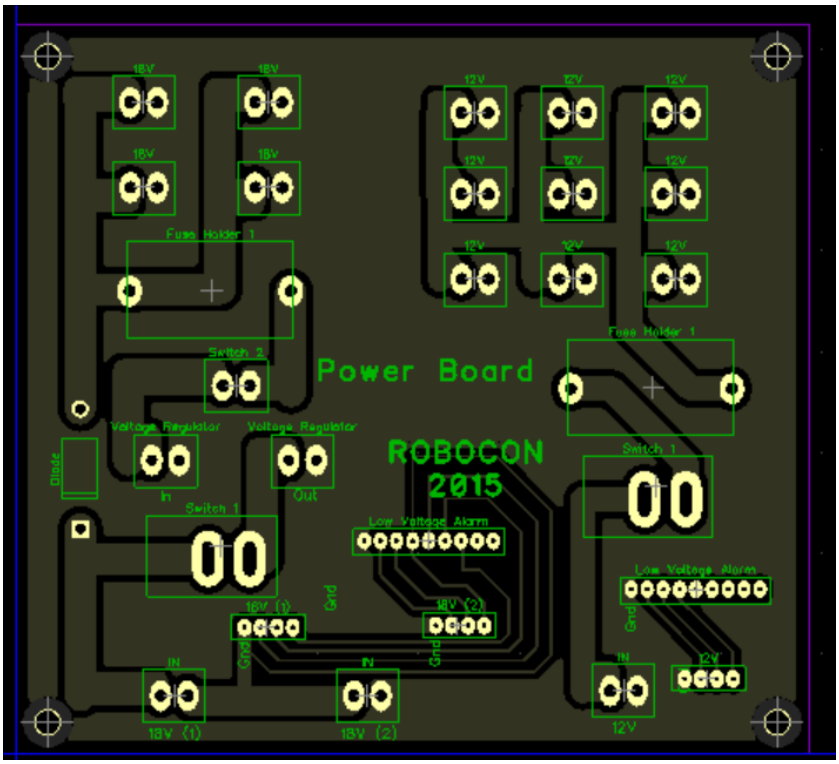

Figure 6.1: Power Board for Back Bot (Top Layer)

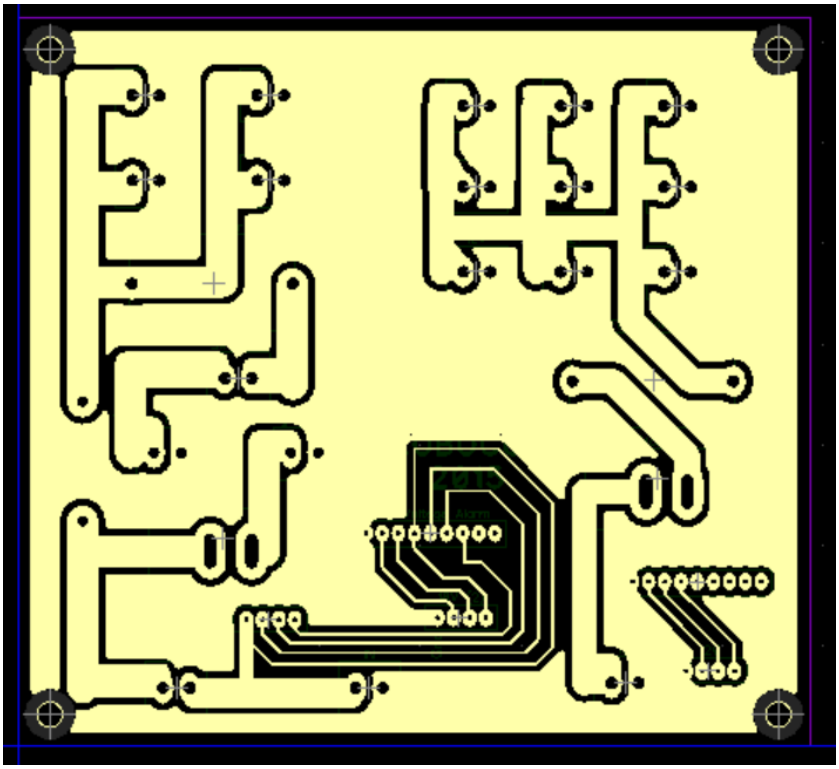

Figure 6.2: Power Board for Back Bot (Bottom Layer)

6.2.2 Back Bot

It had maximum design current of 33A passing through it from battery to the sabertooth driving the BaneBot motors. We had the copper thickness of 0.1mm and we wanted maximum temperature rise of $10°C$ at the ambient condition of $40°C$.

S. No.	Connection end points	Current Rating(A)	Required Trace Width(mm)	Designed Trace Width(mm)
1	24V line	$11 \times 3 = 33$	14.2	20
2	12V line	$Arduino + Sensors \approx 5$	1	3
3	Ground	$18A$	14.2	18 + Copper Pour

Table 6.2: Power Board for Back Bot

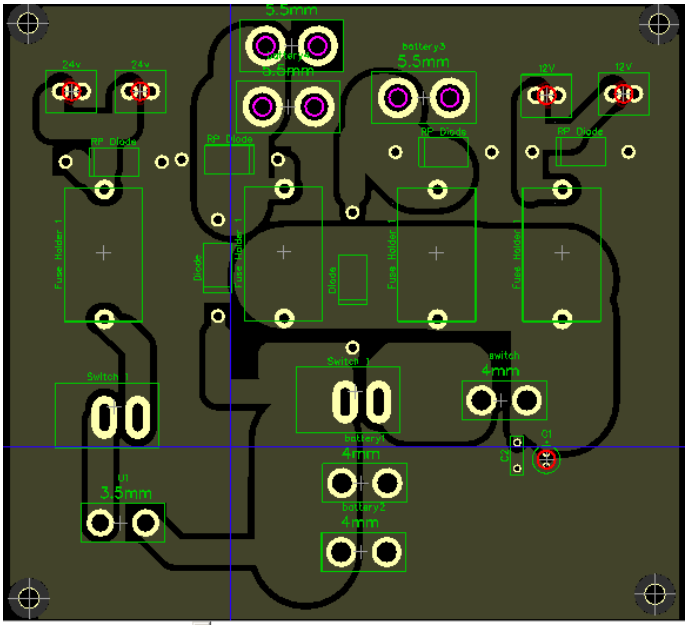

Figure 6.3: Power Board for Back Bot (Top Layer)

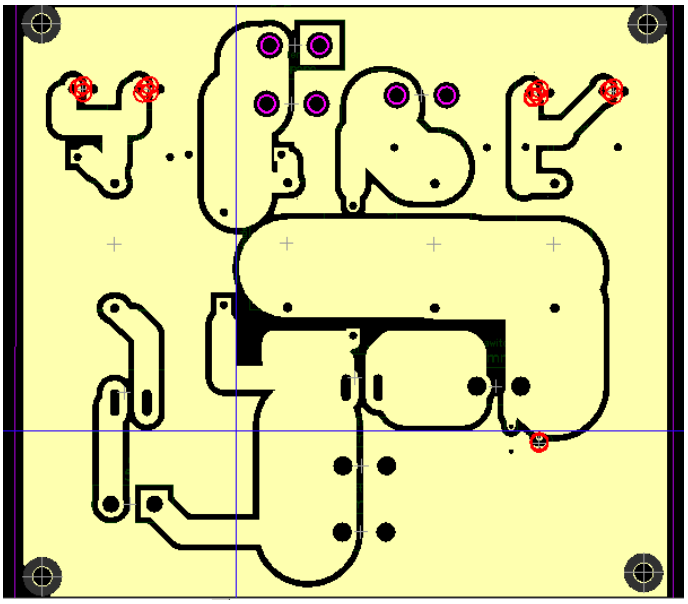

Figure 6.4: Power Board for Back Bot (Bottom Layer)

6.3 Main Board (Drive Board)

It had digital signal on it with maximum current of 500mA and we had copper thickness of 0.05mm.As we wanted maximum temperature rise of $10°C$ at the ambient condition of $40°C$, we got the required trace width to be 0.08mm. Due to manufacturing restrictions and as a factor of safety, we designed the PCB with trace width of 0.2mm.

In the main board of Front bot, we designed it such that we had to but sabertooth on the PCB itself. It created quite a bit of problem as now we had digital and analog signal on the same board. Also, as the area beneath the sabertooth was not used, the economic efficiency of the PCB was quite low. In the starting we decided to design it such that all the electrical components would come in the three triangular sections of the chassis but due to wrong reference diagram, it didn't happen so.

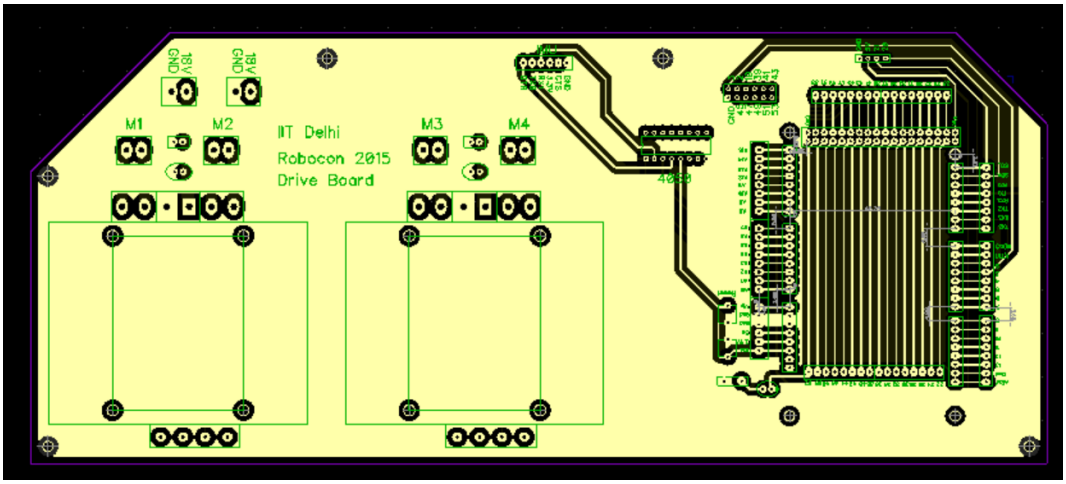

Figure 6.5: Main Board for Front Bot (Top Layer)

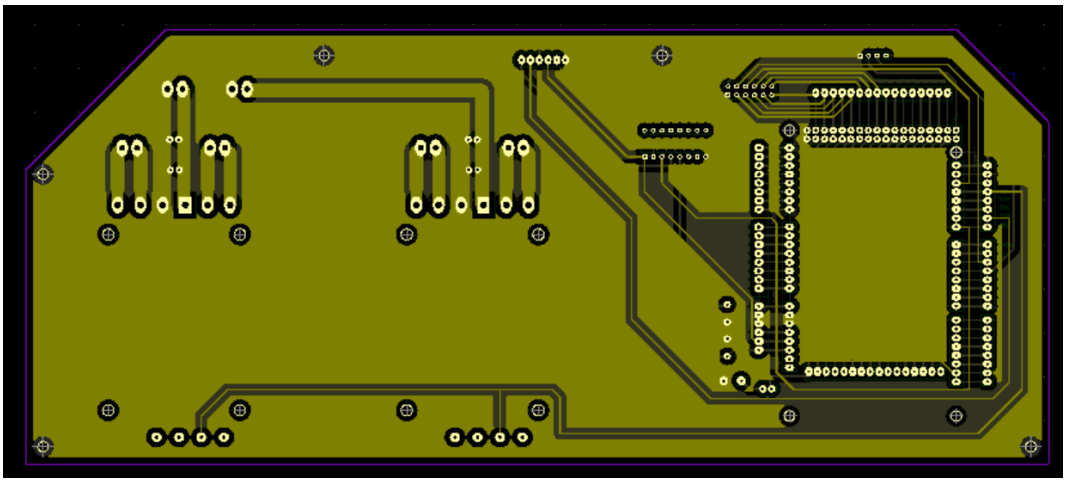

Figure 6.6: Main Board for Front Bot (Bottom Layer)

While designing the main board of Back bot we took care of all the problem that we faced in the Front bot. We put sabertooth separate from the PCB.

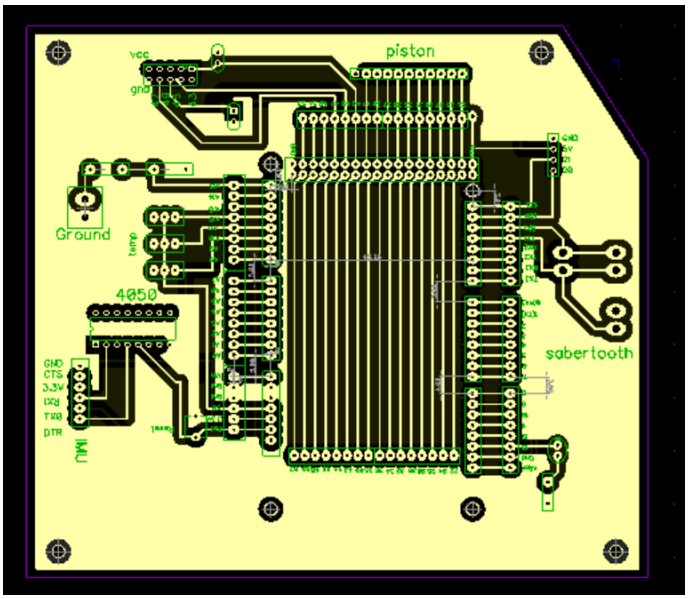

Figure 6.7: Main Board for Back Bot (Top Layer)

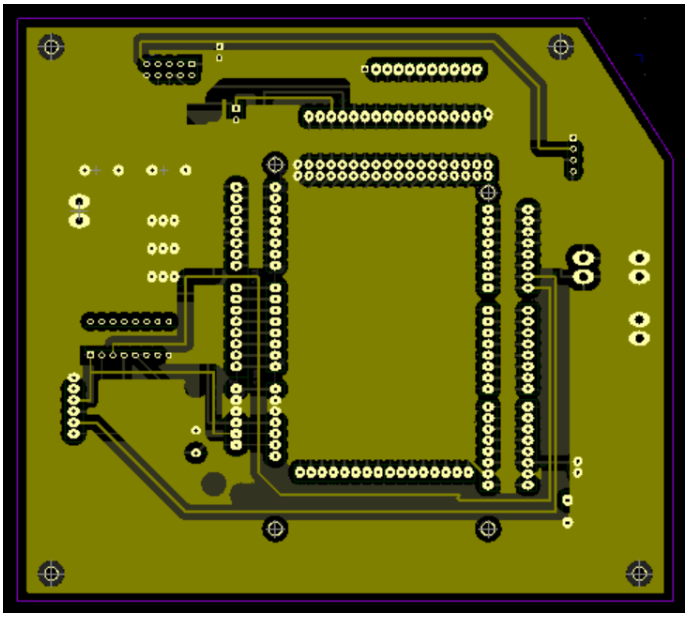

Figure 6.8: Main Board for Back Bot (Bottom Layer)

Chapter 7

Current Detection

7.1 Introduction

As discussed earlier, one of the major problem was the overheating of the Maxon motors. One of the possible reason for this was thought to be over-current in the Maxon motors. So, a sensor module was to be made to keep a log of current to the motors during the whole run. Also, this data was to be used to decide the maximum current limit for the voltage regulator circuit.

7.2 WCS1800

It is the current sensor that we used to get the value of current. It works on the principal of hall effect sensor with current sensing range from $0 - 35A$ at $5VDC$. It gives ratiometric output from voltage supply.

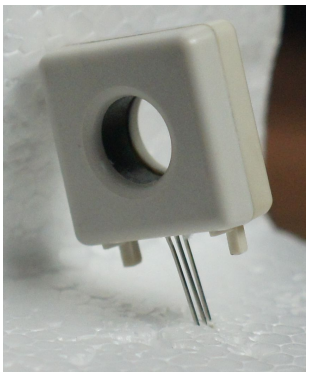

Figure 7.1: WCS1800

7.3 Integration of WCS1800 with Arduino

WCS1800 gives data as a analog signal which has to be analyzed correctly to get the correct value of the current.

7.3.1 Getting Zero value

Since WCS1800 can also be used to get the direction of current, it outputs the zero current at the middle of its output range ($\sim 2.5197V$) and the amperage through the wire is calculated by its deviation from the zero value. As there are many environment factors which were seen to affect the value of zero current, it was decided to calculate the zero value at the start of the arduino sketch.

A function was written which did this work and updated the value of the mean variable. It found an average value of the analog reading for 2 seconds when there was no current flow through the circuit.

```
void getMean()
{
  // get mean value
  long int sensorValue = 0;
  long int counter = 0;
  unsigned long old_millis = millis();
  while ((millis() - old_millis) < 2000) {
    sensorValue += analogRead(currentSensorIn);
    counter++;
    delay(2); //wait for analog pins to refresh
  }
  mean = sensorValue / counter;
}
```

7.3.2 Sensitivity of the sensor

The change in voltage with different current was calculated from the value in the datasheet and converted to proper form to be used in the Arduino sketch.

```
sensitivity = (5/1023)/0.0631;
```

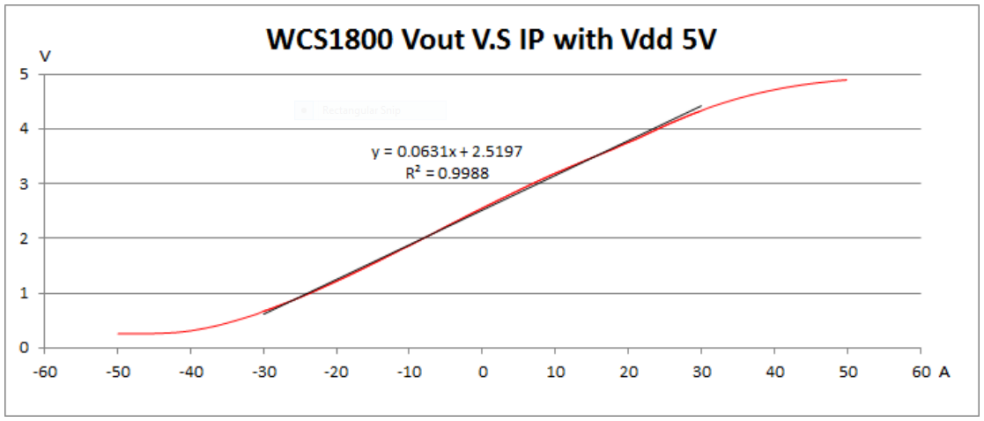

Figure 7.2: WCS1800 sensitivity at 5V supply

7.3.3 Extraction of sensor readings

To take care for the noise induced during readings and other factors, it was decided that the consecutive average of the reading would be taken. The averaged reading was corrected so that now it varied around 0 where 0 was the zero value of the readings and negative sign showed the flow of current in opposite direction than that defined by the sign on the top of the sensor.

The following code was used to read data and convert it into a readable format and store it on SD card. This function was put in the loop, so that, it may run for as long as the bot is powered up.

```
if (counter > 5)
{
float dataValue = ((sensorValue / 5) - mean) * sensitivity;
    writeData(fileName, datavalue); //print data to a file on SD card
    delay(10); //wait for 10ms for data to write on SD card
    counter = 0; //resets counter
    sensorValue = 0; //resets sensor value
}
else
{
counter++;
}
delay(2);
```

7.4 Data Logging

As the setup was run for various times before the data was extracted from the SD card, a function was created that created a new file name for each instance in CSV format.

A new file name is generated for each instance and now the data for that run was saved in the file of that name.

```
char* newDataFile()
{
  // Check if last file number exists
  if (!SD.exists("index.txt"))
  {
    File prev_file = SD.open("index.txt", FILE_WRITE);
    Serial.println("index.txt doesnt exist");
    prev_file.println("0000");
    prev_file.close();
    number = 0;
  }
  else
  {
    // Opening the last file number.
    File prev_file = SD.open("index.txt");
    i = 0;
    while (prev_file.available() && i < 4)
    {
      num[i] = prev_file.read();
      i++;
      Serial.println(num[i]);
    }
    prev_file.close();

    Serial.println(num);
    number = String(num).toInt();
    // generate new file number
    number++;
  }

  // converting file number to 4 bit string.
  i = 3;
  while (i >= 0)
```

```
  {
    int r = number % 10;
    num[i] = r + '0';
    number = number / 10;
    i--;
    Serial.println(String(num));
  }

  //file for data
  filename = "CSP" + String(num) + ".csv";
  filename.toCharArray(file, 12);

  // Remove previous file number and insert new file number
  SD.remove("index.txt");
  File logfile = SD.open("index.txt", FILE_WRITE);
  logfile.println(String(num));
  logfile.close();

  return file;
}
```

7.5 Overall module

A module was created such that it became a standalone unit for a bot and could be used or removed whenever needed. It was a shield for arduino Uno with support for SD card and WCS1800.

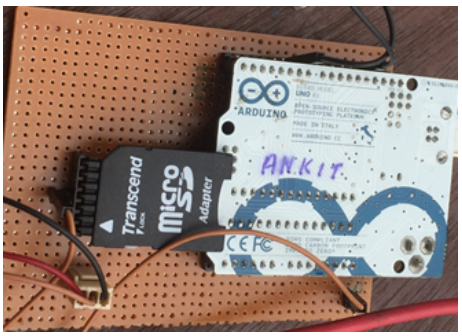

Figure 7.3: Module in Action