# EMEnents 2013 Team Description

Arsalan Akhter[1,3], Usama Tariq Butt[1], Mahmood Azhar[2], Syed Mohsin Raza[3], Taimur Ahmad[3], Ghina Zia[2], M. Shahzad Bhatti[1], Jahanzaib Sohail[2], Muneeb Arshad[3], Zain Imran[2], Asif Mehmood[1], Mesum Tamaar Malik[2], Taousif Iqbal[2], Abbas Amir[1], Muhamamad Salman[2], Kunwar Faraz[3], Shoab Khan[1].

1 Department of Computer Engineering, 2 Department of Electrical Engineering,
3 Department of Mechatronics Engineering
NUST College of Electrical and Mechanical Engineering
National University of Sciences and Technology, Rawalpindi, Pakistan
robocupssl@ceme.nust.edu.pk

**Abstract.** This paper presents an overview of the technical details of team EMEnents, the Robocup Small Size League Team from NUST College of EME, Pakistan, for the year 2013. The paper discusses the current status of the team and the progress of the system developed so far.

## 1 Introduction

Team EMEnents is an interdisciplinary team in its inception phase at NUST College of Electrical and Mechanical Engineering, a constituent college of National University of Sciences and Technology, Pakistan. The vision for participation in Robocup Small Size League was manifested in 2009, and over the following years, the team has matured itself to its current state, where it now intends to participate in Robocup 2013.

In order to fulfill the requirements of the project, three teams, each from a different department, came together to form team EMEnents. Software implementation of the game play and planning modules of the project were implemented by the team members from Computer Engineering and Mechatronics Engineering, whereas the Computer Engineering Team also implemented the software side of wireless communication. The team from Electrical Engineering worked on boost circuit implementation, kicker and dribbler circuitry. The team from Mechatronics Engineering took up the task of design & fabrication of the mechanical structure and relevant electronic circuitry of the robot. The teams from Mechatronics Engineering and Electrical Engineering worked together on electromechanical actuation of motors, wireless communication and feedback control.

We have been working with a three pronged strategy, namely, development of robots and hardware, development of software, integration & testing of individual modules, all going in a parallel fashion. The paper discusses the current system in detail and casts some light on the future directions in which the system is intended to be developed.

# 2 Electronic Design

The Electronic Design Circuitry can be broadly classified into the following parts.

### 2.1 BeagleBone

The most important part of the robot hardware is its central processing unit. For our robots, we employed a relatively newer platform known as the BeagleBone [1]. BeagleBone is an ARM Linux based platform powered by AM3359 superscalar processor capable of clock speeds up to 720 Mhz. It has a number of general purpose input output (GPIO) pins for interfacing a number of modules. It also has dedicated Enhanced High Resolution Pulse Width Modulation Channels (EHRPWMs) which make it ideal for controlling multiple motors. The basic functions of the BeagleBone that we are using are:

- UART for Wireless Linkages
- Independent PWM channels: 4 for wheels, 1 for dribbler, 1 for boost circuit.
- Pin-outs for setting Motor directions
- Input capture modules for sensing motors encoders feedback
- Sensing feedback from IR sensor
- Support for implementation of Kicker and Chipper Module
- Support for implementation of Dribbler Module

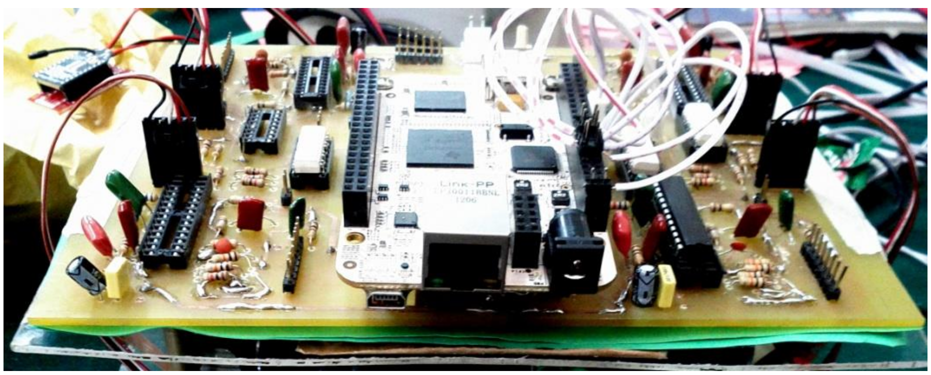

**Fig 2.1 BeagleBone and its peripheral circuitry**

All of the algorithms of the robot's modules like motor control, wireless communication, kicker and dribbler modules have been successfully implemented on this board.

## 2.2 Motor Control Circuitry

For our design, Maxon EC 45 Brushless DC motors were selected as wheel motor and Maxon EC 16 as dribbler motor, whereas L6235 is used to drive these motors. The L6235 IC includes all the circuitry needed to drive a three-phase BLDC motor including a three-phase DMOS Bridge, a constant off time PWM Current Controller and the decoding logic for single ended hall sensors that generates the required sequence for the power stage.

Feedback control is also implemented using US Digital E4P optical encoders attached separately with each wheel motor with a custom made back-extended shaft. Each module takes 3 inputs for each motor; PWM, Direction and Brake. Each robot contains a total of five L6235 modules, four for the wheel motors and one for the dribbler motor.

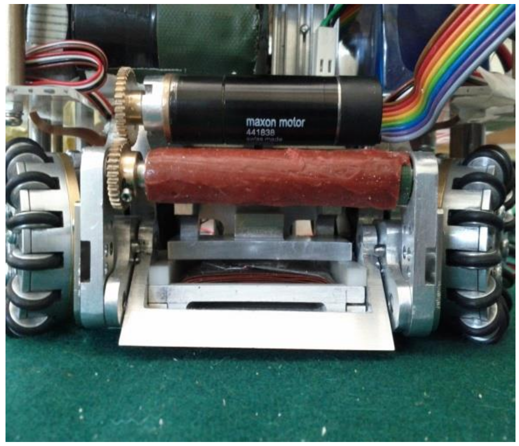


**Fig 2.2(a) Dribbler module with Maxon EC-16 BLDC motor and silicone coated rod.**

Pulses generated by the encoders are fed to Beaglebone which helps in determining the speed of the motor. As per setpoint, PID loop is implemented by Beaglebone by providing required output PWM to L6235. Data received wirelessly will determine the speed and direction of each motor as required by movement of robot. Fig 2.2(b) shows the 3D model of L6235 based circuitry implemented for motor control.

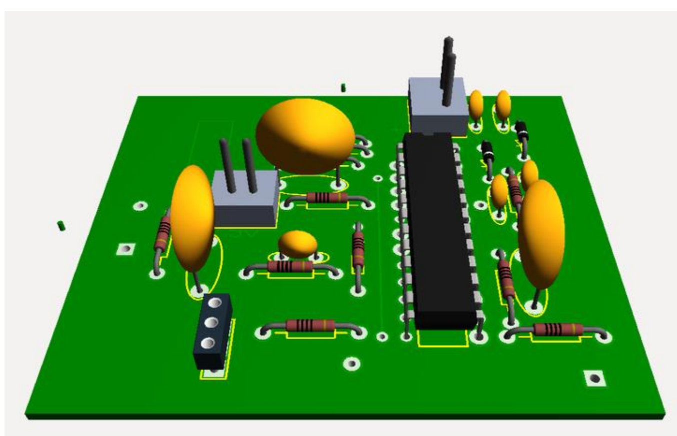

**Fig 2.2(b) 3D model of L6235 circuitry implemented for motor control**

## 2.3 Wireless Communication

For wireless communication between the robots and the AI server, Zigbee (802.15.4) based XBee Series 2 modules have been used due to their efficiency and ease of use. These modules are 3.3V logic devices. Since BeagleBone is also a 3.3V device so interfacing both the modules becomes very easy. The modules are configured in API mode in a point-to-multipoint topology in order to transmit and receive an entire frame of data. This frame consists of fields containing the information regarding motion control, kicking system, dribbling system. The format of the data packet is shown in Fig 2.3

| Field: | ID | Vx Sign Flag | Vy Sign Flag | W Sign Flag | Kick Flag | Chip Flag | Vx Integer Part | Vy Integer Part | W Integer Part | Kick Speed | Vx Fractional Part | Vy Fractional Part | W Fractional Part |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of Bits: | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 8 | 8 | 8 |

**Fig.2.3 Packet format for communication from AI Server to the Robot**

Currently we are employing one way transmission from the AI server to the robot, but we also plan to send the status of robot battery and control parameters from robot to AI server for features such as online debugging in future.

## 2.4 Ball Shooting Mechanism

For our kicking system, we are using solenoid based kicking carried out using a custom built solenoid. Two 2200 uF capacitors are charged to 200V by a boost circuit. Our robots have two kicking mechanisms, namely flat kick and the chip kick. The solenoid used for flat kicking has 700 turns of AWG 25 wire, while the chip kicker has 350 turns of AWG 25 wire. The kicker system is capable of delivering ball speeds as high as 10m/s, which is then limited to 8m/s in software as per the F180 rules. The kicking speeds are controlled by changing the on-time of IGBTs in the software.

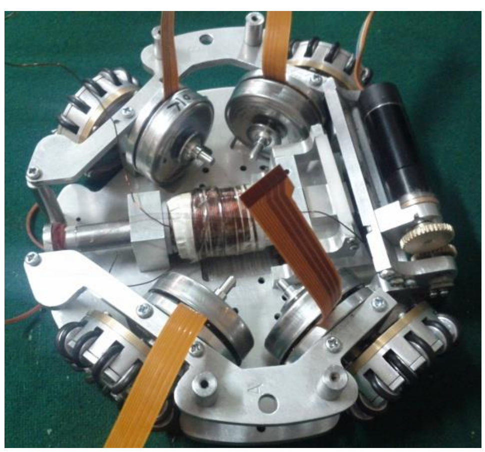

**Fig.2.4 (a) Top view of the kicker module showing the solenoid gun**

Data points based on on-time of the IGBTs and the output ball speed were collected and a function based on on-time in milliseconds vs the output ball speed at 200V was formulated. This function was then employed in the practical implementation of the kicker circuit and was used to calculate the desired ball speed. This function currently calculates values considering the assumption that the capacitors are charged at 200V. The graph showing our ball kicking speed vs. the on-time of power IGBT for 200V is shown in Fig 2.4(c).

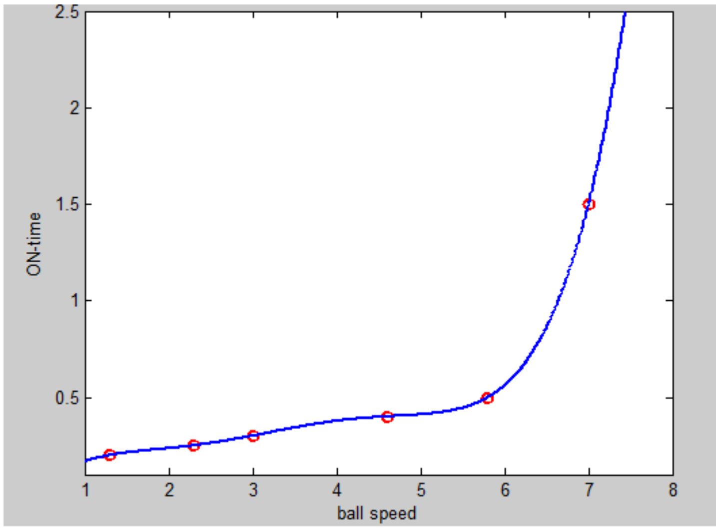


**Fig 2.4(c) Graph showing on-time vs ball speeds**

We employed acoustic measurement techniques for determining the kicking speed of kicker while development. Fig 2.4(d) shows our recording software in which we recorded two sound peaks, first when the kicker touches the ball and second when the ball hits the target present at known position. By measuring the time between these two peaks and using simple kinematic equations of motion, we were able to calculate the ball velocity with very high accuracy. This data can easily help us for accurate passing between different players.

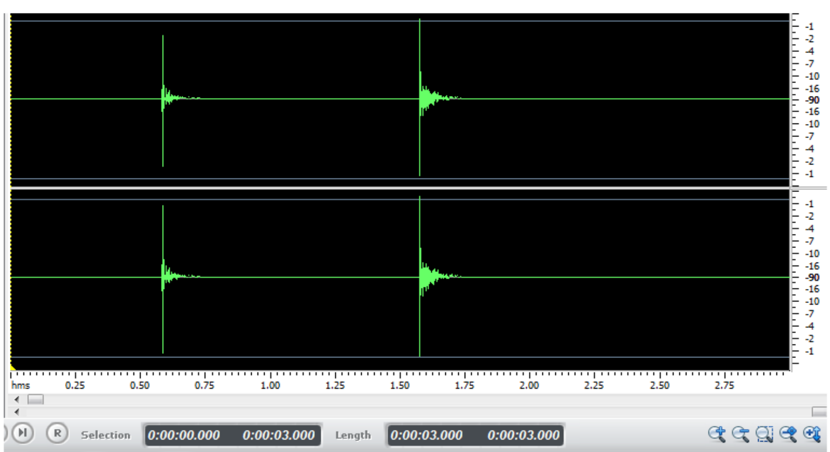


**Fig 2.4 (d) Acoustic measurement of ball speed**

## 3 Mechanical Design

Mechanical Designs of the robots used in robocup small size league share same basic structure. The designs of our robots were inspired from Skuba [2], which have a modular design, making the robot easy to debug in case of any problem. Fig 3.1 shows the complete CAD assembly of the robot, which was generated using the design software Pro-Engineer Wildfire. Dimensional limitations as per Robocup SSL were followed strictly to keep robot within 180 mm diameter and 150 mm height.

All parts were fabricated indigenously under supervision of our hardware team.  Fig 3.2 shows final fabricated assembly of the robot. We plan to make further modifications and improvements in the hardware in future for better performance, such as selection of appropriate materials for gears, using appropriate wheel diameter and better motors etc.

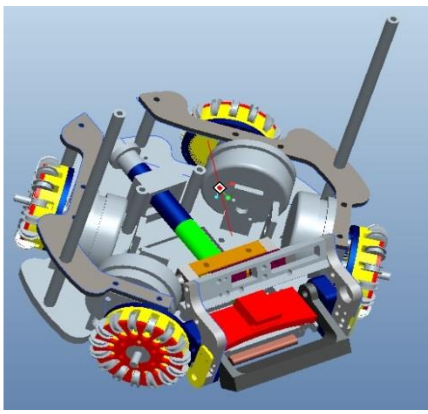
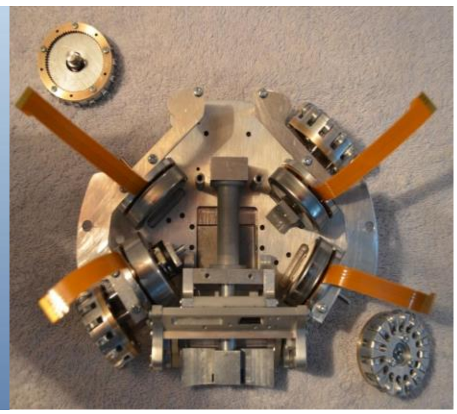

**Fig 3.1: Assembly of robot in Pro-E**    **Fig 3.2: Fabricated assembly**

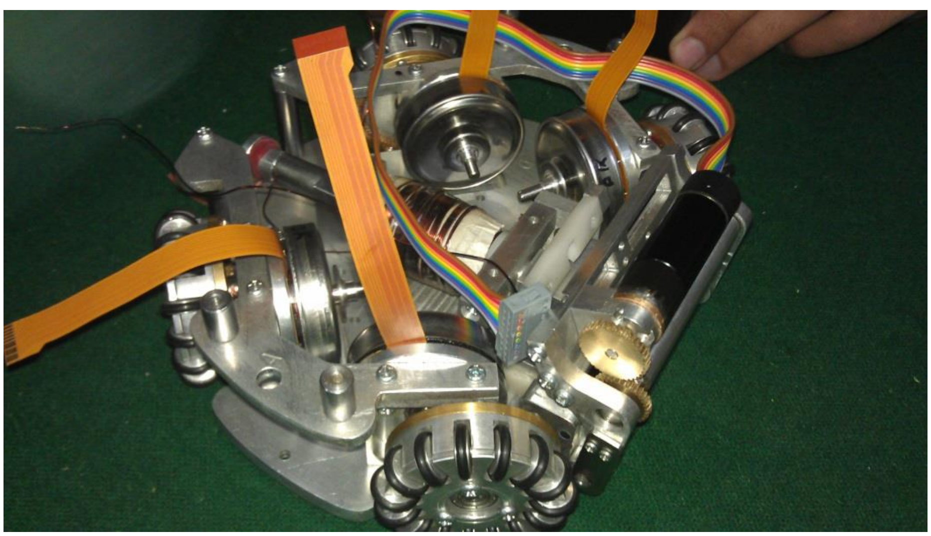

**Fig 3.3 Complete Assembly with Dribbler Mechanism**

## 4 Software Architecture

The Software architecture of the system follows a modular approach where different modules work collaboratively to run the AI agent with one module's output being used as the next module's input. Main flow of data between modules is shown in Fig 4.1 [2]

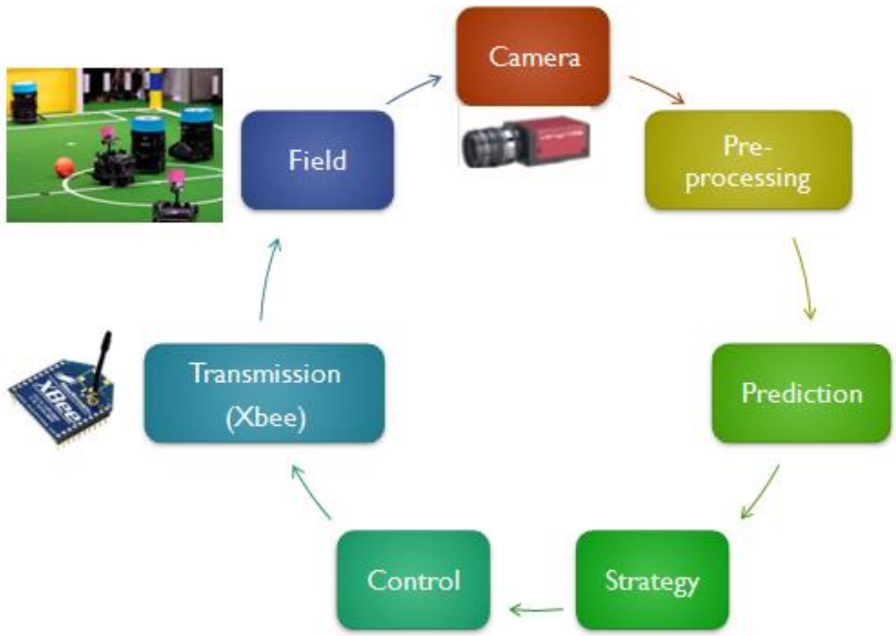


**Fig 4.1 Software System Architecture**

### 4.1 Prediction Module

A prediction module receives positions of robots and ball on the field from SSL-Vision. Purpose of this module is to remove possible noise in position of objects, predict their next positions and velocities to be used further in decision making.

### 4.2 Strategy Planning

Strategy module is mainly based on one presented by Michael Bowling in his PHD thesis [3]. Using bottom up approach the basic ability is a skill i.e. action that can be performed by a robot. Skill set contains a number of skills. For mutual behavior skills of individual robots are used to form a play. Play is a combination of skills for multiple robots to behave cooperatively. Plays are executed based on the conditions on the field.

### 4.3 Skill Set

Skill is an action that can be performed by individual robot independent of others. Skill set is a collection of all skills. A skill normally makes use of current state of robot and information from field to decide the desired state and actions to be performed by robot. Some examples from skill set are goto_loose_ball, pass, shoot_on_goal, best_deflection_position etc.

### 4.4 Control Module

After the destination decision by strategy module using skills, control module gets current and destination positions of each robot. Purpose of this module is to plan a path between these points avoiding obstacles in between and generate velocity commands accordingly.

Control module provides linear and angular velocities along with other command for each robot on the field. These commands are than to be transmitted to each individual robot on the field. To do so they are first converted to a packet format as shown in fig 2.3 and then transmitted to each robot over Xbee modules.

As a preliminary implementation, Tangent Bug algorithm [4] is used for path planning and obstacle avoidance assuming the environment static. A circular area of particular radius is taken around the robot as sensing area. Motion of robot is adjusted according to presence of any obstacle in that area. In a more advance approach and for actual environment we plan to use Rapidly Exploring Random Trees (RRTs) based approaches.

## 5 SSL Vision

The shared vision system of Robocup Small Size League named SSL vision [ref] was employed for position acquisition of robots. SSL-Vision was successfully integrated with the software AI agent.
Fig 5.1(a) shows the actual image of field where markers represent the location of robots. Fig 5.1(b) is color calibrated image on SSL-Vision. Fig 5.1(c) is the representation of robots on graphical client provided by SSL-Vision.

**Fig 8.1 (a) Actual Field**

**Fig 8.1 (b) Color calibrated on SSL-Vision**

**Fig 8.1(c) Positions on graphical client**

## 6 Conclusion

The Team Description paper outlines the general architecture of our software and hardware system and lays a foundation of the system as a whole. The paper also sheds light on the future projections and directions towards which the team intend to move. We, as a new team, look forward to having a contributing and learning experience with other teams and the SSL community in the future.

## Acknowledgements

We are thankful to Higher Education Commission of Pakistan for the initial funding for the said project.

## References

1. BeagleBone Rev A6 System Reference Manual - BeagleBoard

2. Krit Chaiso, Teeratath Ariyachartphadungkit, Kanjanapan Sukvichai : Skuba Team Description. In Proceeding of Robocup 2011.

3. Michael Bowling, Multiagent Learning in the Presence of Agents with Limitations, School of Computer Science,Carnegie Mellon University, Pittsburgh.

4. Choset H., Lynch K. M., Hutchinson S. Kantor G., Burgard W., Kavraki L. E., Thrun Sebastian. Principles of Robot Motion, MIT Press.