E- Yantra Project Documentation

Remotely Controlled Robot

TEAM : 9

Bhaskar Bandyopadhyay

Nilekh Chaudhari

Uddhav Arote

VIDYAVARDHINI`S COLLEGE OF ENGINEERING AND TECHNOLOGY, VASAI.

Table Of Contents

Contents

1. **Introduction**:

The system consists of a web interface which help the user to control the motion of the
robot sets its IO port values and also gives them the chance to send batch scripts which will be executed at the robot side. The communication will be done via Zigbee protocol (IEEE 802.15.4) working on IEEE 802.15 standard of communication.

The client, a web browser, requests the Web server via HTTP protocol. The server interprets the action from the user at the client side and accordingly takes the necessary step. At the server, action packet is created and is sent via serially connected Zigbee module to similar module on the robot side.

The robot interprets the action in the packet and acts accordingly. The robot, then sends the packet to the server via same Zigbee module. The packet contains the information necessary to the user at the client side.

After server receives the packet, it sends this packet data to the client in the form HTTP response. The client side user uses this data for some good reason.

2. Problem Statement:

Controlling the robots over the Internet and monitoring its actions via a web interface, setting resetting of the I/O port. The problem includes controlling multiple robots and the customized hardware like a robotic arm via the interface. Creating the replica of the arena or industry area on the HTML5 canvas element and tracking the robot movements. Giving the alert messages to the user at the client side regarding low battery level of the robot.Similarly, giving the mission critical messages as per the sensor values like , when the sensors detects any obstacle proximity , the system says “Obstacle ahead”. Replicating the behavior of Gmail chat system in the interface, where the robots will be chatting with the user at the client side.

3. Requirements:

The system has following hardware and software requirements :

<u>Hardware</u>

Hardware	**Use in system**
SPARK V robot	The robot that will be controlled is SPARK V robot from Nex-robotics by the user.
Zigbee modules	The communication between sever computer and SPARK V robot will take place via 2 Zigbee modules[Maxstream XBee / XBee Pro] .
Server computer with serial port	The server computer with serial port is required to communicate with robot and the XBee module will be connected to its serial port.
Client computer with RJ 45 port	Client computer with RJ 45 port is needed to communicate with the sever computer using TCP/IP protocol stack.

<u>Software</u>

Software	**Use in system**
Web browser with any OS	Any web browser like Mozilla Firefox, Opera on any OS platform is used to send requests to the Apache Server.
Apache server with GNU/Linux OS	Apache is a web server on which PHP scripts can be run.
GCC C compiler	To compile the C programs.
AVR Boot Loader	The hex code of the program is burnt into the microcontroller using this software.

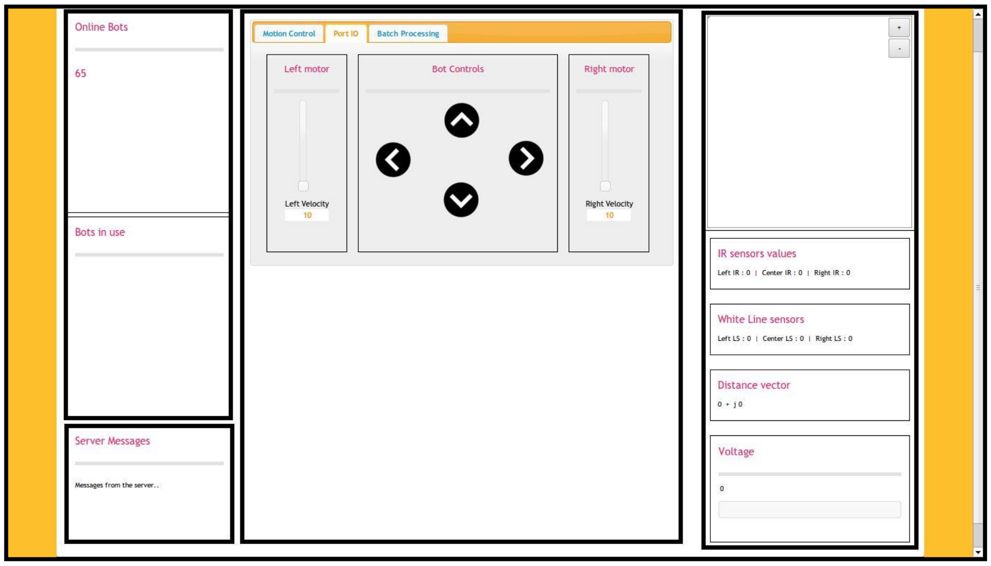
MySQL database server	The database server keeps the information of the registered users and the bots.
AVR Studio	An IDE used to develop the program for the robot, build the program, simulate the program.

4. Implementation:

4.1 Abstract code for the Web interface:

Note: Tags are not standard, they are defined so that the reader can get the idea of interface properly

```
<html>

        <body>

                <main>

                        <left_col>

                                <div_1>

                                        <! -- bots available-->

                                </div_1>

                                <div_2>

                                        <! -- bots in usee-->

                                </div_2>

                                <div_3>

                                        <! -- server messages-->

                                </div_3>

                        </left_col>


                        <middle_col>

                                <tab_1>

                                        <! -- motion control -->

                                </tab_1>

                                <tab_2>

                                        <! -- port io -->

                                </tab_2>

                                <tab_3>

                                        <! -- batch processing -->
                                </tab_3>
                        </middle_col>

                        <right_col>
                                <div_1>
                                        <! – canvas -->
                                </div_1>
                                <div_2>
                                        <! -- IR sensor values -->
                                </div_2>
                                <div_3>
                                        <! -- white line sensors -->
                                </div_3>
                                <div_4>
                                        <! -- distance in vector form -->
                                </div_4>
                                <div_5>
                                        <! -- robot battery voltage -->
                                </div_5>
                        </right_col>
                </main>
        </body>
</html>
```



The User interface consists of three parts:

- Section I: It is aligned to the left of the user interface and contains 3 divisions as follows :

  O Division 1: It will contain the number of robots online and who have registered to the system.

  O Division 2: It will contain the robots which are currently in use. These robots cannot be utilized by the other users who try to use these robots

  O Division 3: It will contain the messages from the server..

- Section II : It is aligned to the right of the user interface and contains 5 divisions as follows:

  O Division 1: It will contain the canvas which graphically displays the position from the robot.

  O Division 2 to Division 4: It will contain the sensor values and distance vector received from the robot ,like IR sensors , distance sensors.

  O Division 5: It will contain the value related to the voltage of the battery set up on the robot.

- Section III : It is in the middle of the user interface and contains tabs for various functionalities. The number of tabs is at least 3, as follows:

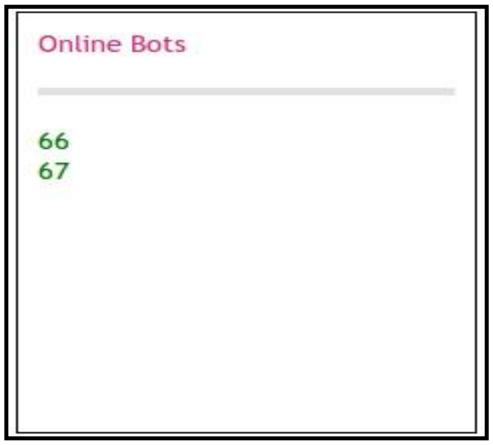
O Tab 1 [Motion Control]: It will help in controlling the motion of the robots using the key events .The velocity of the robots will also be controlled in this tab.

O Tab 2 [IO Port manipulation]: It will help to manipulate the IO port values of the robots.

O Tab3 [Batch Processing]: It will help the user to write batch scripts so that it can be executed on the robot.

LEFT column of the interface:

Description:

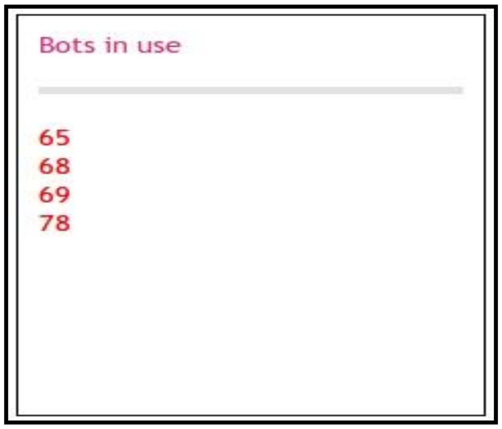


Online Bots

66
67

1. **Division 1:**

Online Bots: This division consists of the names of the robots which are online at that time. The robots come online after they register with the system. From this section the robots can be chosen and can be controlled. The robots which are available are visible to all the clients.

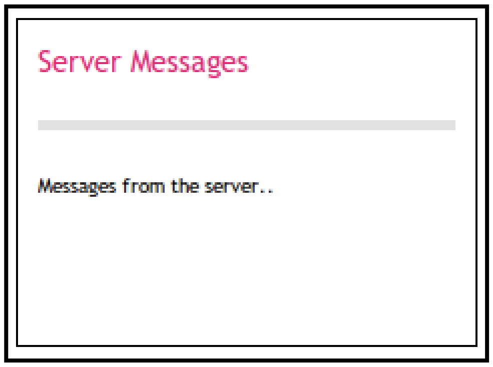


Bots in use

65
68
69
78

2. **Division 2:**

Bots in use: This division consists the name of the robots which are in use by the clients. No single robot can be controlled by two or more clients. And no other user can control other`s robot.


Server Messages

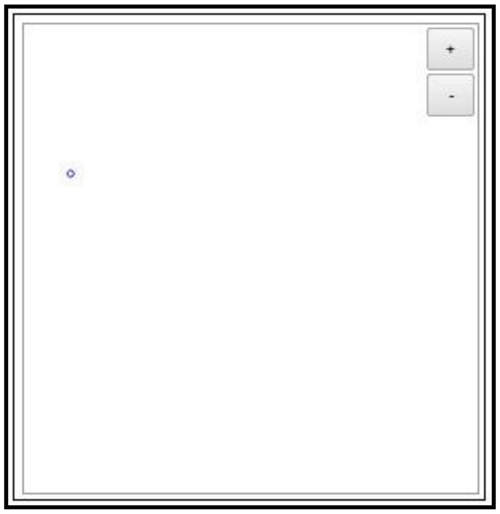
Messages from the server..

3. **Division 3:**

Server Messages: This division displays the critical messages. Messages like 'Obstacle ahead', 'Battery down'. 'Dead End' will be displayed.

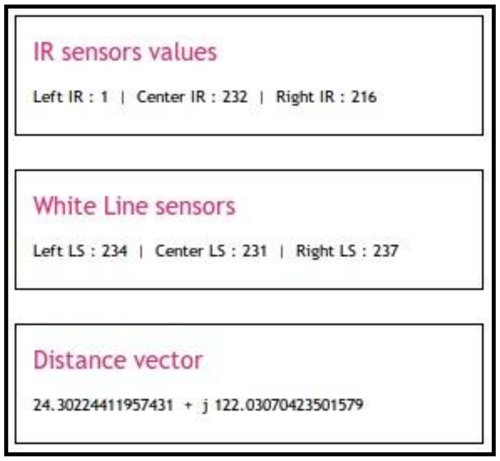
RIGHT column of the interface:

Description:



1. **Division 1:**

Map of the arena: The division includes the canvas element of the HTML5. The robot movements are mapped on to the canvas. The canvas displays the movement of only bots which are used by the client. The map can be zoomed and panned.


IR sensors values

Left IR : 1 | Center IR : 232 | Right IR : 216

White Line sensors

Left LS : 234 | Center LS : 231 | Right LS : 237

Distance vector

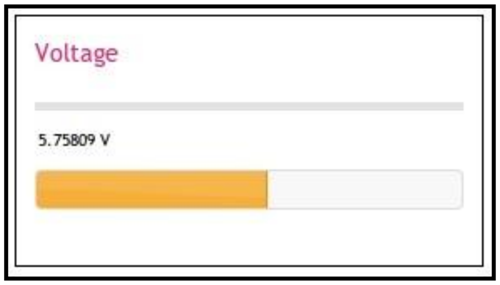
24.30224411957431 + j 122.03070423501579

2. **Division 2 – 4 :**

IR sensor values: This division displays the IR sensor values which are received from the robot. The values are monitored and accordingly messages are displayed.

White Line Sensors: This division displays the white line sensor values which are received from the robot.

Distance vector: The division displays the displacement of the robot in the vector form. Using this vector the robot will be mapped on the canvas.



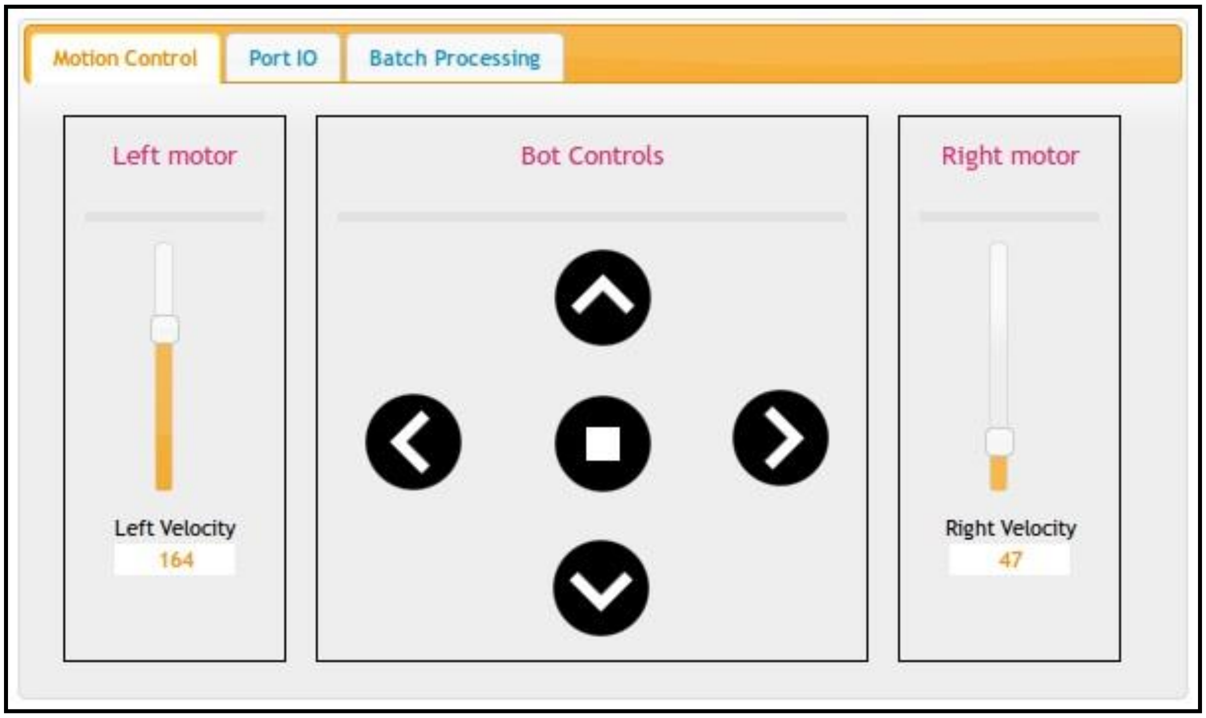
3. **Division 5:**

Voltage: This division displays the voltage level of the battery of the robot.

Middle column of the interface:

Description:

1. **Tab 1 : Motion-control:**

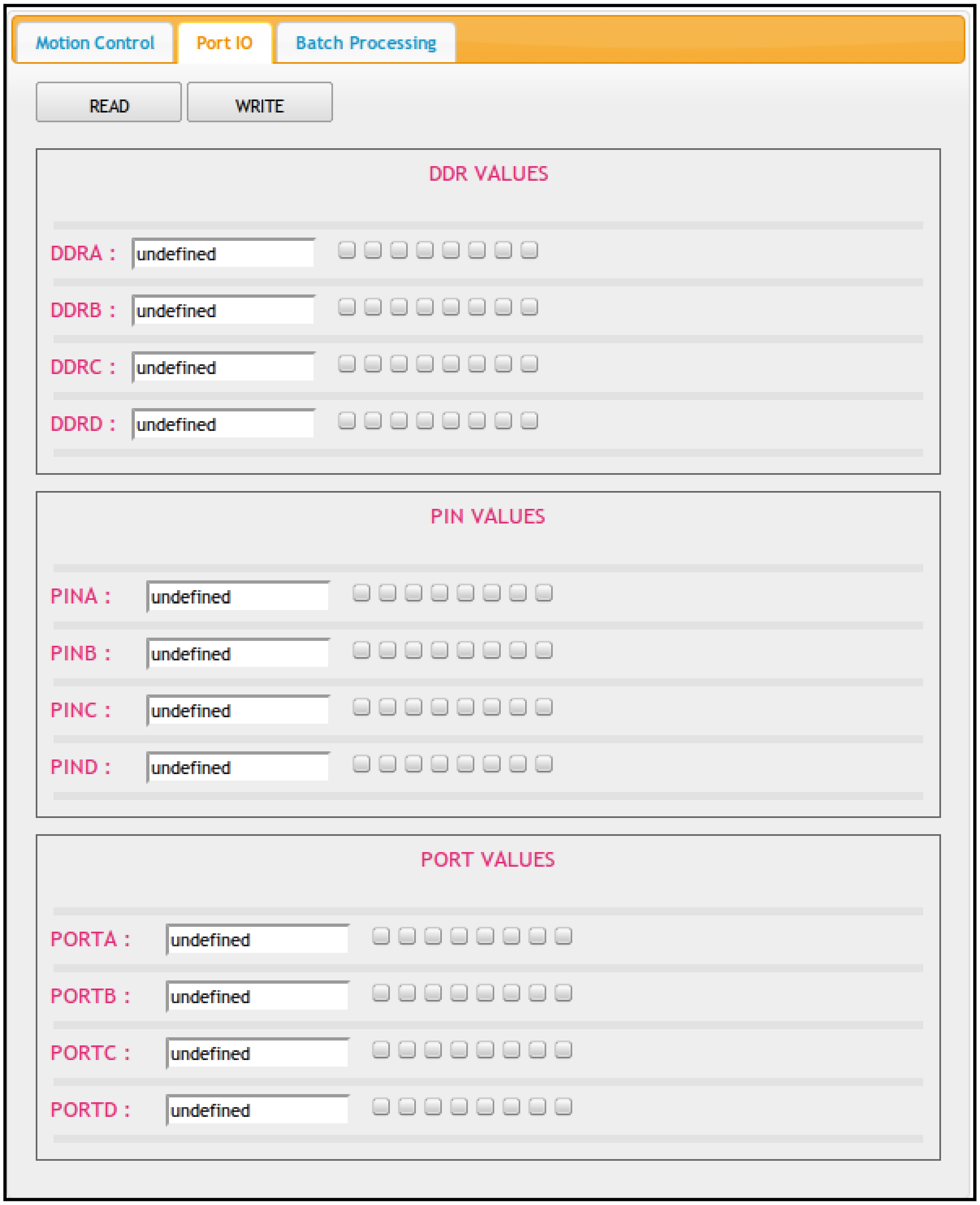


Left velocity: The slider is designed in Jquery which controls the velocity of the left motor.

Bot controls: The 5 buttons control the forward, backward, left, right motion of the robot. The middle button stops the robot.

Right velocity: The slider is designed in Jquery which controls the velocity of the right motor.

2. **Tab 2 : Port IO:**



DDR VALUES

DDRA :	undefined
DDRB :	undefined
DDRC :	undefined
DDRD :	undefined

PIN VALUES

PINA :	undefined
PINB :	undefined
PINC :	undefined
PIND :	undefined

PORT VALUES

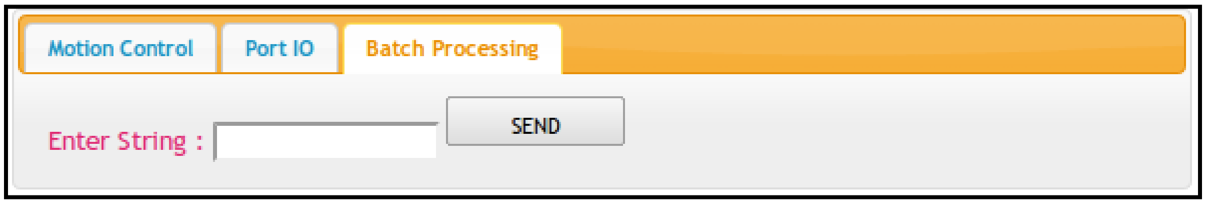
PORTA :	undefined
PORTB :	undefined
PORTC :	undefined
PORTD :	undefined

DDR values: This division maps the pins of 4 ports of SPARK V robot for setting the pins as Input or Output.

PIN values: This division maps the pins of 4 ports of SPARK V robot, for reading the port values.

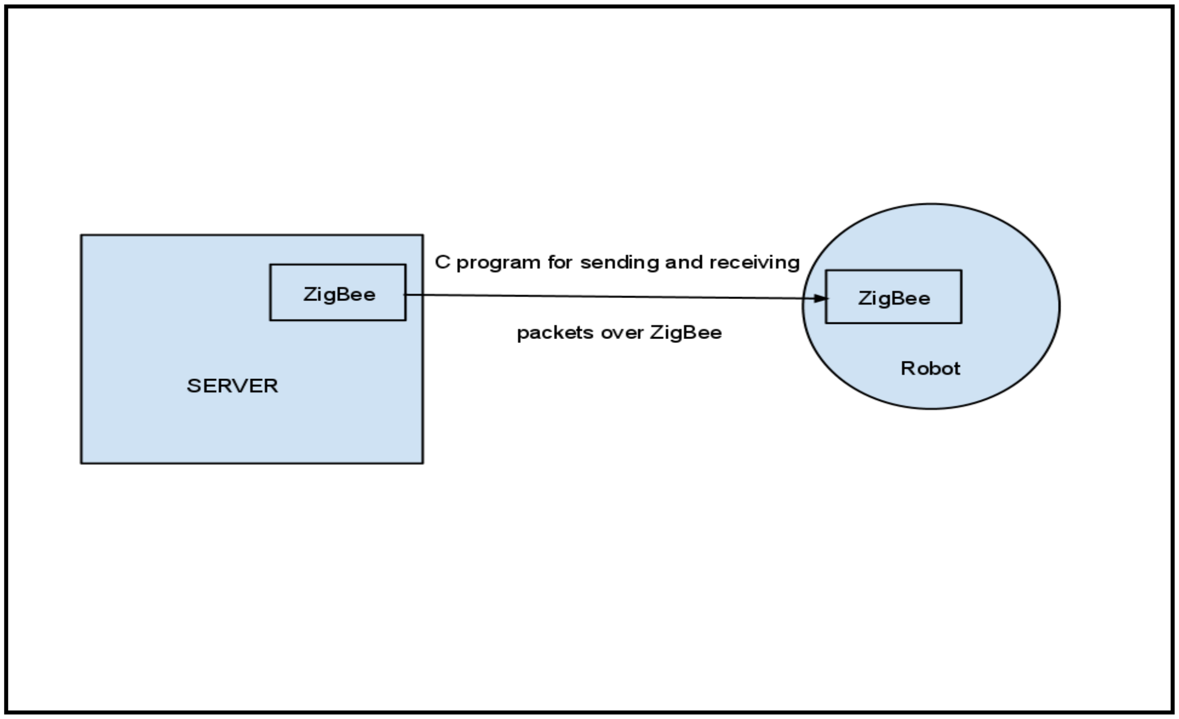
PORT values: This division maps the pins of 4 ports of SPARK V robot, for setting the port values.

3. Tab 3: Batch processing



LCD display: The text field takes the string to be printed on the LCD.

4.2 Communication between Apache web server and robot (SPARC V):



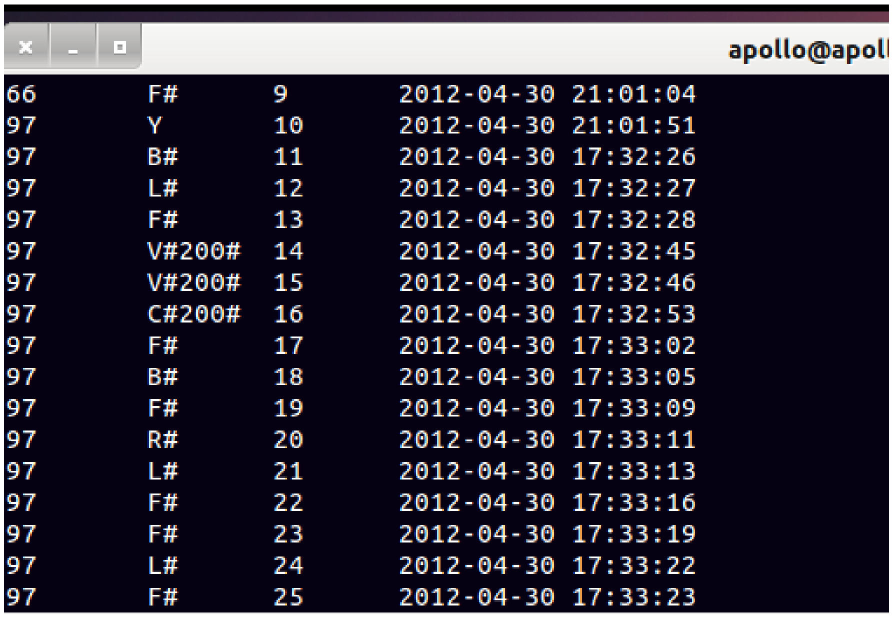
The fig shows the communication between the web server and the Zigbee associated robots.
At the server side, a daemon called SENDS (an executable file) sends the commands by extracting them from the MySQL database. The robot program monitors the packets sent by the server Zigbee and acts according to the content of the packet like forward action, backward action, switch on the buzzer, switch off the buzzer etc.
At the server side another daemon called RECEIVE (an executable file) receives the packets sent by the robot. The data from the packets is extracted and put into the MySQL database. The PHP script then read the database and displays it on the web interface. This daemon also registers the robot to the system. Register in the sense, the robot ID is entered into the database and is made available for the use for the users. It is not permanent registration, every time the robot is powered up after shutting down it has to register again to the system.

SEND daemon Algorithm :

1. Start
2. Initialize the serial port.
3. Connect to MySQL database.
4. Fire query to select the commands in the database.
5. Loop
    a. Send the command via the serial port after prepending the robot id to the command.
    b. Delete all the commands until the last command timestamp.
    c. Select the new commands after the last executed command timestamp by ordering.

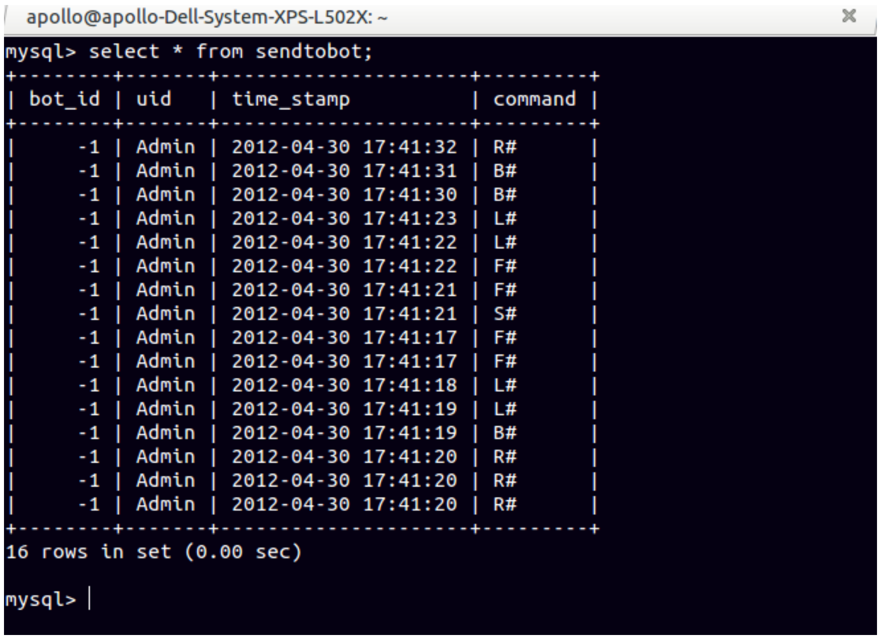
.

❖ Output of SEND daemon :



```
apollo@apol
66        F#      9       2012-04-30 21:01:04
97        Y       10      2012-04-30 21:01:51
97        B#      11      2012-04-30 17:32:26
97        L#      12      2012-04-30 17:32:27
97        F#      13      2012-04-30 17:32:28
97        V#200#  14      2012-04-30 17:32:45
97        V#200#  15      2012-04-30 17:32:46
97        C#200#  16      2012-04-30 17:32:53
97        F#      17      2012-04-30 17:33:02
97        B#      18      2012-04-30 17:33:05
97        F#      19      2012-04-30 17:33:09
97        R#      20      2012-04-30 17:33:11
97        L#      21      2012-04-30 17:33:13
97        F#      22      2012-04-30 17:33:16
97        F#      23      2012-04-30 17:33:19
97        L#      24      2012-04-30 17:33:22
97        F#      25      2012-04-30 17:33:23
```

❖ Database snapshot of table : sendtobot

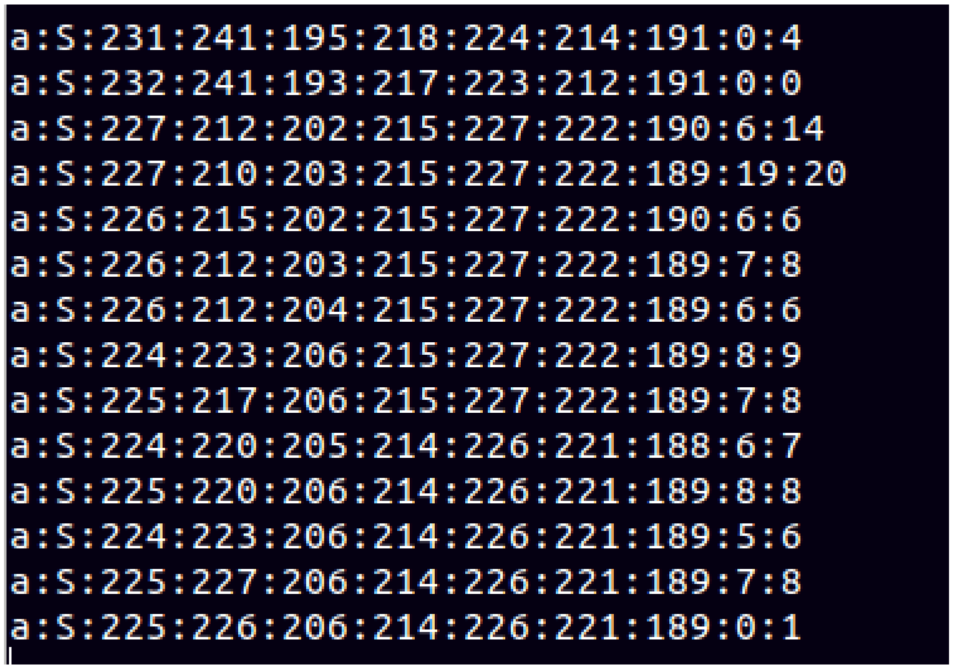


```
mysql> select * from sendtobot;
+--------+-------+---------------------+---------+
| bot_id | uid   | time_stamp          | command |
+--------+-------+---------------------+---------+
|     -1 | Admin | 2012-04-30 17:41:32 | R#      |
|     -1 | Admin | 2012-04-30 17:41:31 | B#      |
|     -1 | Admin | 2012-04-30 17:41:30 | B#      |
|     -1 | Admin | 2012-04-30 17:41:23 | L#      |
|     -1 | Admin | 2012-04-30 17:41:22 | L#      |
|     -1 | Admin | 2012-04-30 17:41:22 | F#      |
|     -1 | Admin | 2012-04-30 17:41:21 | F#      |
|     -1 | Admin | 2012-04-30 17:41:21 | S#      |
|     -1 | Admin | 2012-04-30 17:41:17 | F#      |
|     -1 | Admin | 2012-04-30 17:41:17 | F#      |
|     -1 | Admin | 2012-04-30 17:41:18 | L#      |
|     -1 | Admin | 2012-04-30 17:41:19 | L#      |
|     -1 | Admin | 2012-04-30 17:41:19 | B#      |
|     -1 | Admin | 2012-04-30 17:41:20 | R#      |
|     -1 | Admin | 2012-04-30 17:41:20 | R#      |
|     -1 | Admin | 2012-04-30 17:41:20 | R#      |
+--------+-------+---------------------+---------+
16 rows in set (0.00 sec)

mysql> 
```

RECEIVE daemon Algorithm :

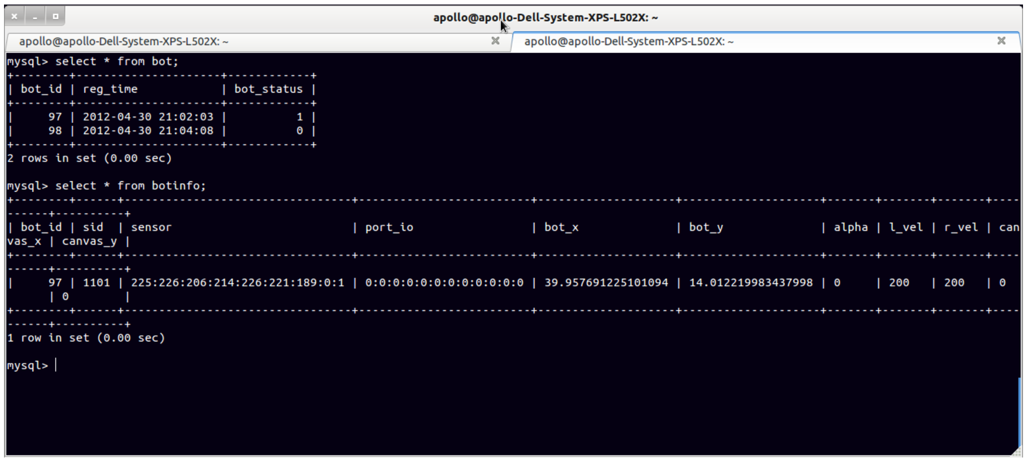
```
1. Start
2. Initialize serial port.
3. Connect to MySQL database.
4. Loop
        a. Register the robot.
        b. If  isBotRegistered  then
                i. Fire query to the database tables, bot,sendtobot and botinfo.
        c. Receive the robot info i.e the sensor and port values
        d. Update the database tables entries accordingly.
```

❖ Output of RECEIVE daemon :



```
a:S:231:241:195:218:224:214:191:0:4
a:S:232:241:193:217:223:212:191:0:0
a:S:227:212:202:215:227:222:190:6:14
a:S:227:210:203:215:227:222:189:19:20
a:S:226:215:202:215:227:222:190:6:6
a:S:226:212:203:215:227:222:189:7:8
a:S:226:212:204:215:227:222:189:6:6
a:S:224:223:206:215:227:222:189:8:9
a:S:225:217:206:215:227:222:189:7:8
a:S:224:220:205:214:226:221:188:6:7
a:S:225:220:206:214:226:221:189:8:8
a:S:224:223:206:214:226:221:189:5:6
a:S:225:227:206:214:226:221:189:7:8
a:S:225:226:206:214:226:221:189:0:1
```

❖ Snapshot of database table : botinfo , bot.

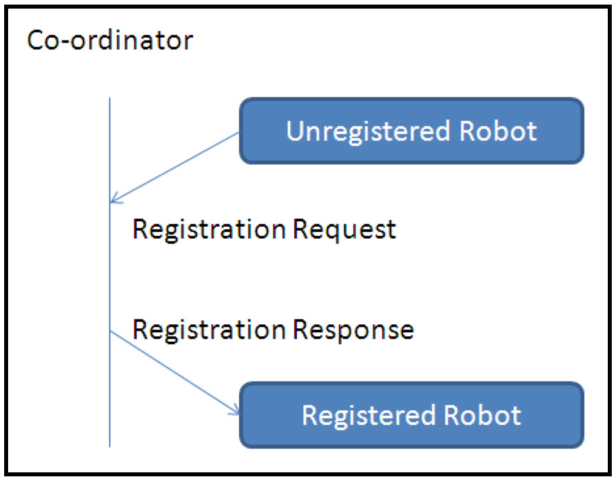


apollo@apollo-Dell-System-XPS-L502X: ~
apollo@apollo-Dell-System-XPS-L502X: ~
apollo@apollo-Dell-System-XPS-L502X: ~

```
mysql> select * from bot;
+--------+---------------------+------------+
| bot_id | reg_time            | bot_status |
+--------+---------------------+------------+
|     97 | 2012-04-30 21:02:03 |          1 |
|     98 | 2012-04-30 21:04:08 |          0 |
+--------+---------------------+------------+
2 rows in set (0.00 sec)

mysql> select * from botinfo;
+--------+------+---------------------------------+-------------------------+--------------------+--------------------+-------+-------+-------+----
------+----------+
| bot_id | sid  | sensor                          | port_io                 | bot_x              | bot_y              | alpha | l_vel | r_vel | can
vas_x | canvas_y |
+--------+------+---------------------------------+-------------------------+--------------------+--------------------+-------+-------+-------+----
------+----------+
|     97 | 1101 | 225:226:206:214:226:221:189:0:1 | 0:0:0:0:0:0:0:0:0:0:0:0 | 39.957691225101094 | 14.012219983437998 | 0     | 200   | 200   | 0
      | 0        |
+--------+------+---------------------------------+-------------------------+--------------------+--------------------+-------+-------+-------+----
------+----------+
1 row in set (0.00 sec)

mysql> |
```

4.3 Transfer of packets between server and robot:

For the transfer of the packets from coordinator and robot the Zigbee modules are set in the broadcasting mode. In this mode each and every robot (Zigbee) sends data to the coordinator which is routed to the other robots (Zigbee). In broadcasting mode there is no acknowledge received at the data link layer.

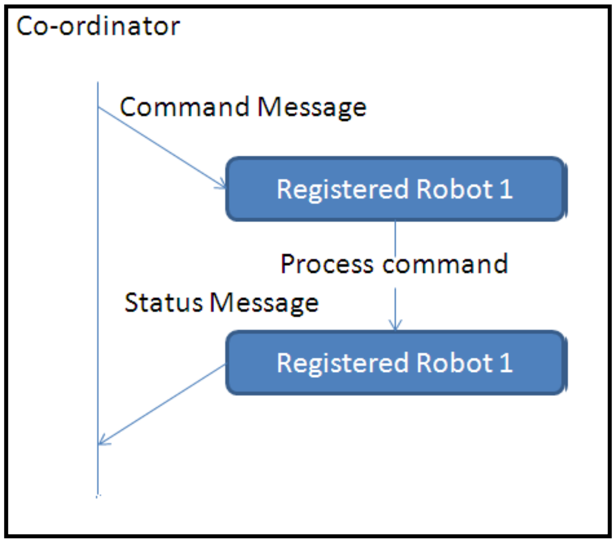


The **fig** shows the communication between the robot and the coordinator indicating the registration process of the robot with the system. The unregistered robot sends the registration request to the coordinator. The coordinator then replies with the registration response and the robot gets registered. Now, if two robots send the registration request simultaneously, the bots are registered individually depending upon whose request is received first.

Each and every robot sends its unique id as a registration request. The unique id is of the form "@X", where '@' is the forbidden character and X is robot id ranging from a-z. Now whenever robot receives the '@' character it resets its counter value. After the counter value is overflowed the registration request is sent again. This process continues until the acknowledgement from coordinator is received.

Packet type	**Command Format**	**Command Description**
Registration Request	@X	The request contains 2 characters '@' and 'X' .@ is the forbidden character and X is any character between 'a' and 'z' which is the robot id.

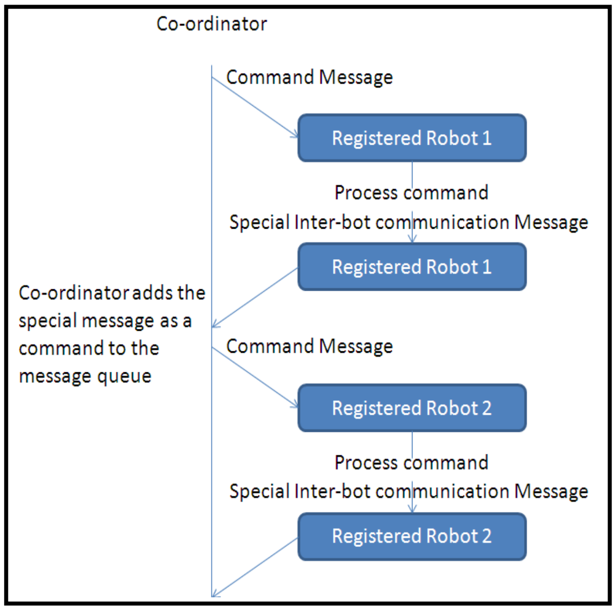
Registration Response	Y	The response indicates that the robot request is processed as a result of which the entry is put into the database.



The **fig** shows coordinator sending the command messages to the registered robot. The command is then processed by the robot and it functions according to the message contain. After executing the command the robot sends a status message to the coordinator.

Packet Type	**Command Format**	**Command Description**
Command message	X#	The message contains the action to be taken and the delimiter #.The action is interpreted by the robot and it is executed at the robot side.
Status message	S:[sensor values]# P:[port values]#	The status contains the sensor or the port values.’S’ and ‘P’ indicates that the values are sensor or port values. Followed by S or P is ‘:’ indicating the delimiter between command and the

		values.The values comes next.At the end is the # delimiter.



Co-ordinator
Command Message
Registered Robot 1
Process command
Special Inter-bot communication Message
Registered Robot 1
Co-ordinator adds the special message as a command to the message queue
Command Message
Registered Robot 2
Process command
Special Inter-bot communication Message
Registered Robot 2

The **fig** shows coordinator sending the command messages to the registered robots. The command is then processed by the robots and it functions according to the message contain. After executing the commands the robots sends status messages to the coordinator. The command is prefixed with the robot id. This helps the robots to identify the messages intended to them. Due to this the robots do not executes the wrong command.

Packet type	**Command format**	**Command description**
Command message	botID:X#	The message contains the bot id and the action to be executed. # is the delimiter.
Status message	botID:S:[sensor values]# botID:P:[port values]#	The message is same as for the single robot. Only change is the robot id is prepended with the command.

List of Commands used in the System :

Command	**Description**
F#	Move the robot forward
B#	Move the robot backward
L#	Move the robot left
R#	Move the robot right
S#	Stop the robot
Z#	Start the buzzer of the robot
N#	Stop the buzzer of the robot
P#	The robot sends the port values to the server
W#	Write the port values on the robot.
D#	Write the string on the LCD.

5.Testing Strategy and Data:

5.1 Interface Testing:

Test Case number	Test case name	Test case description	Testing Strategy
1	Canvas testing	The movement of the registered robot is replicated on the canvas.	The robot is started. When the arrow keys are pressed corresponding changes occurs in the canvas.
2	Robot Registration	The robot registers to the system so that it can be used by the user at the client side. Here the robots send the packet to the server and the sever will register the robot to the system.	Testing can be done by starting the robot. After the robot starts it sends the registration request which causes the server to send an acknowledgement on successful registration. As a result interface is updated with the newly registered robot id.
3	User authentication	Before the client can use the system, the user at the client side will be authenticated by the system .If not authenticated the robot can be used by any person in the world.	When the user is authenticated, user is redirected to the home page of the web application. Else user is redirected back to the login page.
4	Simple robot movement	The keyboard arrow keys as well as the images of the arrow keys on the interface	The motion control tab of the interface will be used to test this case.

		will be used to control the motion of the robots.	
5	Port I/O	The IO port pins of the robot can be set / reset from the system.	The port I/O tab contains two buttons READ and WRITE. When the read button is pressed the checkboxes and the text field are populated with the port values of the robot. When the write button is pressed the port values which are changed in the text field are set on the robot.
6	Battery voltage	This division shows the battery voltage using the Jquery progress bar UI element.	Whenever bot is registered its current battery voltage is displayed.
7	Batch processing	A script will be given to the robot and it will execute it , eg : Printing the characters on the LCD.	Input the string in the text field Batch Processing tab. Click on the SEND button. The text will be displayed on the LCD of the robot.
8	Multiple robot registration	Number of robots will register to the system so that they can be used by the users at the client side. It is same as test case 2, only difference is; it is done for many robots.	Start more than one robots simultaneously. If the interface shows the number of bot id equal to number of bots in the arena then test is successful.
9	Sensor values	This division displays the	Whenever robot is registered

		sensor values of the robot as simple text.	and any movement of the robot occurs then it displays the sensor values as simple text.

5.2 Robot/Communication Testing:

Test Case number	Test case name	Test case description	Testing Strategy
1	Packet transmission via Zigbee Module	Sending and receiving the packets via the Zigbee module at the two ends of the communication line i.e. the server side and robot side modules communicate by sending and receiving the packets.	X-CTU can be used for testing the packet transmission via the Zigbee module.
2	Motion configuration	To test the port at which the motors are connected.	The PORT values can be set for the robot and tested with simple functions to test the motion configuration. The functions are forward, backward, left, right.
3	ADC configuration	To test the port at which the ADC is connected.	The sensor values are converted by the ADC.

			This can be tested by checking the sensor values printed on the interface.
4	TIMER configuration	To test the timer configuration of the robot.	The registration is done user the timer. So if the registration is done successfully then the timer is configured perfectly.
5	Multiple robot registration	Number of robots will register to the system so that they can be used by the users at the client side. It is same as test case 2, only difference is; it is done for many robots.	Start more than one robots simultaneously. If the interface shows the number of bot id equal to number of bots in the arena then test is successful.
6	One client controlling multiple bots	A single user can control many robots together. It will be like , a user (say A) will control the motion of the robots .These robots won`t be controlled by the other users (say B,C,D,...) when the robot will be controlled by A.	This can be tested by opening the web application in different tabs of the web browser and selecting the required robot to carry out the task. This successfully completes the test case.
7	Multiple clients controlling multiple bots	Many users will control many robots, keeping in mind that M (users):1(robot) relationship won`t be allowed.	This can be tested by multiple client after logging in to the system and selecting multiple robots.

6. Discussion of System:

6.1What worked according to the plan :

- Feasibility of the architecture :

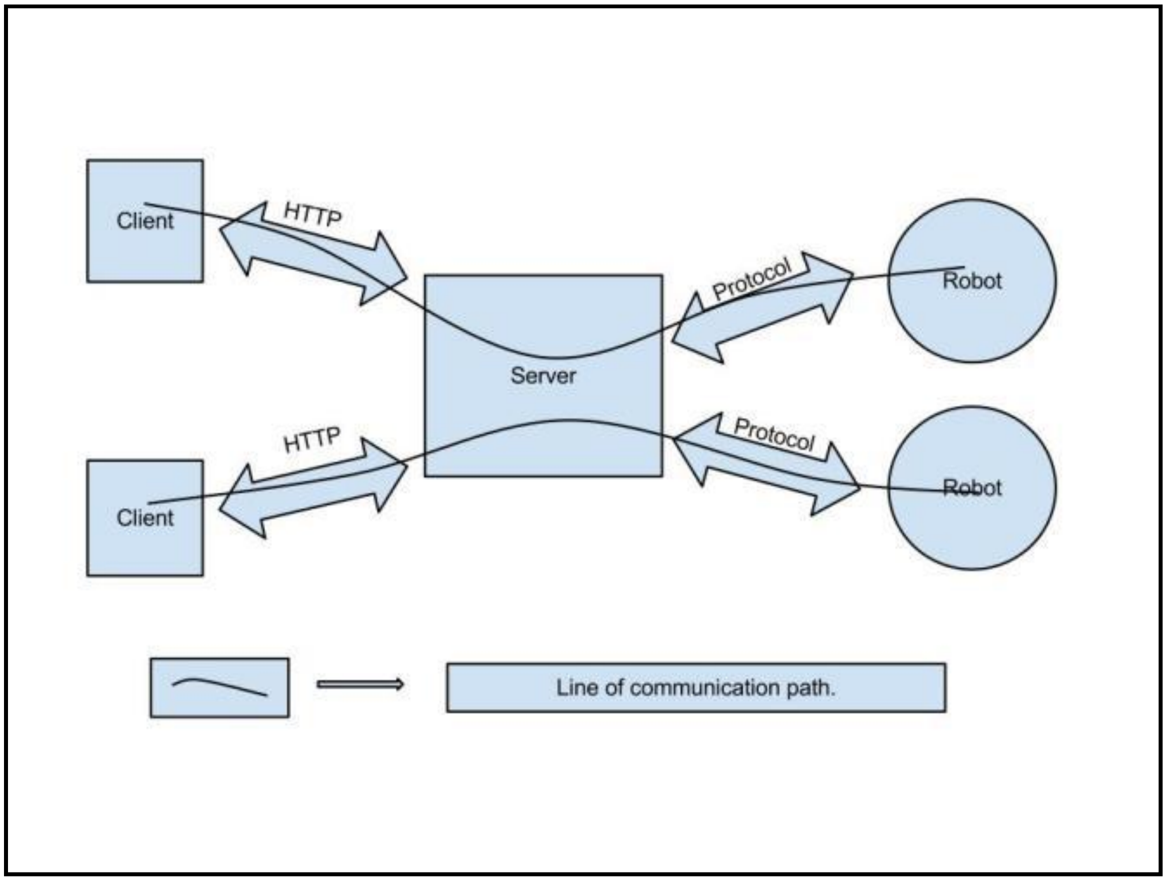


Fig. represents the architecture for the proposed system. As seen in the figure, there can be multiple clients as well as multiple robots. At a time one robot can be controlled by one client only.

All clients communicate with the web server using HTTP. The data packet is then propagated to the robot via designed protocol. The reverse communication from the robot to the client happens in the same way, from robot to web server via protocol and further from server to the client via HTTP. The practical implementation has been conceived by using Zigbee as the protocol of choice while communicating between server and robot. For this, a Zigbee module has been used at server side and robot side.

- Zigbee communication for single robot :

The communication of the server Zigbee module and the module connected to the robot successfully done. The server communicates with the robot using the peer to peer to architecture, also with the broadcast mode enabled.

- Zigbee communication for multiple robot :

The robots communicate with the server module. For this the broadcast mode is used ,where the server module is the co-ordinator and the rest are the end device. All the robots send the data which is routed via the co-ordinator to the other robots.

- Interface as planned :

The interface is designed as per the plan. The interface contains various divisions which are used for different purposes,such as the available robot div,used robot div,server messages div and others.

6.2 What did not worked as per plan :

- Plotting of robot position on Canvas is not accurate :

Due to improper alignment of the wheels of the robot, there is some error in the readings of the encoder values.

- Unnecessary transmission of packet :

The system required that the communication should happen between the server Zigbee and the intended robot. But since the broadcast mode is used the packets are unnecessarily transferred to all the robots.

- Interpreter :

The interpreter was introduced in the system design, which would interpret the code on the robot.We have studied 2 interpreter ‘Little C’ and NanoVM, but due to time restriction we could not complete the implementation of the same.

7. Future Work:

- Implementing the concept mentioned above for all the different robotic platform available.
- Enabling batch scripts support in different programming languages.
- To support media streaming over the robot.
- 3D mapping of the arena.

8. Conclusion:

The project makes use of programming techniques, MAC techniques, interfacing with micro-controllers and their programming, compiler design and web designing. Spark V robot was successfully controlled over the internet. They can be also controlled from remote location. Also the port-IO manipulation and onboard interpretation was accomplished for the same.

9. References:

[1] (2012) IO port programming. [Online] Available:
http://www.faqs.org/docs/Linux-mini/IO-Port-Programming.html

[2] (2012) Serial communication. [Online] Available:
http://www.easysw.com/~mike/serial/serial.html

[3] (2012) XBee / XBee Pro datasheet. [Online] Available:
http://www.nexrobotics.com/images/downloads/Manual_xb_oem-rf-modules_802.15.4.pdf

[4] (2012) Java Virtual Machine specification. [Online] Available:
http://java.sun.com/docs/books/jvms/second_edition/html/VMSpecTOC.doc.html

[5] (2012) The NanoVM-Java for the AVR. [Online] Available:
http://www.harbaum.org/till/nanovm/index.shtml

[6] N. Baker, "Zigbee and bluetooth strengths and weaknesses for industrial applications", Computing & Control Engineering Journal, vol. 16, pp.

[7]http://www.foresight.gov.uk/Previous_Projects/Intelligent_Infrastructure_Systems/Index.htm

[8] "Remote Sensing: A Supplemental Tool for Vehicle emission Control," EPA 400-F-92-017, and Environmental Protection Agency.

[9] P. Kinney, ZigBee Technology: Wireless Control that Simply Works, White Paper dated 2 October 2003.

[10] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava, " Tree-Based Data Broadcast in IEEE 802.15.4 and ZigBee Networks" , IEEE Transactions on Mobile Computing, 5 (11),

2006, pp. 1561-1574.

[11] G. Ding, Z. Sahinoglu, B. Bhargava, P. Orlik, and J. Zhang, “ Reliable Broadcast in ZigBee Networks” , 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'05), Sep. 26-29, 2006, Santa Clara, USA. pp. 510520.

[12]Tian-Wen Song, Chu-Sing Yang, “A Connectivity Improving Mechanism for ZigBee Wireless Sensor Networks” IEEE, 2008 Computer Society.