## Course Project Documentation

# BE Project 2011-2012

# Autonomous Room Mapping Robot

## TEAM : 2

Submitted by

**Ankit Trivedi**
**Ankita Shrotriya**
**Rahul Patil**
**Prathamesh Akolkar**

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Robots are becoming more and more common in our daily lives showing up in the form of everything from children's toys, to robotic vacuum cleaners, to home security robots; robots have been doing automated tasks in factories for decades. With the ever-increasing speed and power of digital systems coupled with the continuously expanding field of robotics, it is becoming more practical to build custom robotic systems with a degree of flexibility and freedom that was once impossible, giving robots the ability to communicate wirelessly or to act autonomously.

All researchers are also interested in robots as a way to understand human (and not just human) intelligence in its primary function - interacting with the real world. Due to the wide and advanced applications of robotics, we planned to do our project in robotics field.

## 1.1 Research on the project

At the start of the VIIth semester, we had started working on cleaning application of the robot, named as **'Autonomous Vacuum Cleaner'**.

Initially, we worked on the literature and IEEE papers of the project, and we found the working model of the project. We worked on the different algorithms for cleaning and navigation purposes.

The cleaner was supposed to be work in five different operating modes for cleaning.

1. Square spiraling
2. Wall following
3. Room crossing
4. Dirt detection
5. Back to dock (automatic charging point for the robot)

But, this cleaner had certain limitations.

- **Unaware about the surrounding environment.**
  The system was inefficient to detect the different types of obstacles, if kept in a heavily crowded room. It was unable to detect the perfect size and dimensions of the room eg. Small, large and medium.
- **Inefficient navigation**
  It does not ensure the maximum coverage of an area for cleaning purpose. It is gullible for multiple passes for the same section of the room which results in wastage of battery power for cleaning the same area repeatedly.

Due to these limitations of the algorithm, we are exploring the idea of room mapping. Hence, we are working on the project called ***'Autonomous Room Mapping'.*** The robot will automatically scan the surrounding environment by calculating different parameters and will simultaneously display the map (boundary) of the arena.

# Chapter 2

# Problem Statement

The goal for an autonomous robot to be able to construct (or use) a map or floor plan and to localize itself in it. Robotic mapping is that branch of one, which deals with the study and application of ability to construct map or floor plan by the autonomous robot and to localize itself in it. It is an intelligent robot because it will itself locate the area which has to be mapped and navigate itself as per predefined algorithm. As everything is automated in nature, the manual efforts are reduced for calculating different parameters for mapping.

In our project, we are using Firebird V Atmega2560 platform. Along with it, the different components such as the servo motor, sharp IR sensor,zigbee wireless module with an operating frequency of 2.4GHz. Initially, the robot will start mapping process from any corner of an arena prescribed by the user. The start point of the robot will be designated as block position (0,0) where robot will start its operation. Initially, the robot will be facing NORTH. At block position (0,0),the servomotor shaft will rotate in a semicircular orientation from $0^\circ$ to $180^\circ$. The sharp IR sensor mounted on a shaft of a servo motor will take 22 different samples at different angles. These samples in a form of range will be sent to the computer via zigbee wireless module. Thus, the robot will navigate, scan and map the arena as per the predefined algorithm.

# Chapter 3

# Requirements

1. **Hardware Requirements**

   (a) **Firebird V(Atmega2560):** Required for mapping.

   (b) **ZigBee(XB24):** Serial communication from bot to host pc.

   (c) **Servo motor(VS1):** Scanning with resolution of $9^\circ$.

   (d) **Sharp IR Sensor(GP2D12):** Measuring distance between bot and obstacles.

2. **Software Requirements**

   (a) **AVR studio:** To program instruction onto a given bot.

   (b) **MATLAB:** Displaying the integrated map.

# Chapter 4

# Implementation

## 4.1 Arena Description

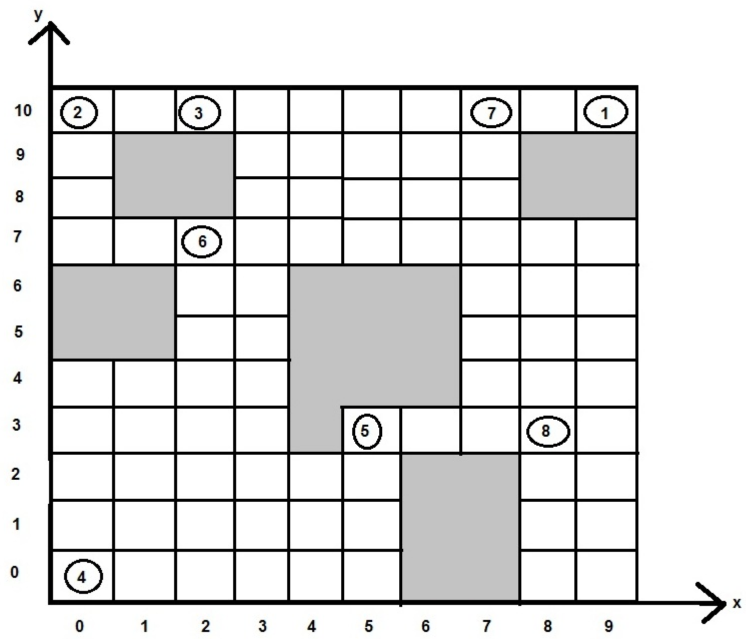



Figure 4.1: Overview of an arena

Initially, for demonstrative purpose, we have the surrounding area(ARENA) of a particular room. This arena will be virtually created in the memory of the robot. This arena will have dimensions of 140 sq.cms with 49 blocks in total. The arena contains a square blocks with the side of 20cms each. The arena will have different obstacles of different size placed randomly. We can take an example of a living room having different obstacles like sofas, chairs, bed etc. placed at different locations.

## 4.2 Working

The robot will perform mapping process based on two operating modes.

1. Scanning and Mapping
2. Navigation

### 4.2.1 Scanning and Mapping

The scanning process will be done in two steps are as follows:

1. 180 ° Scanning
2. Matrix Generation

180 ° **Scanning**

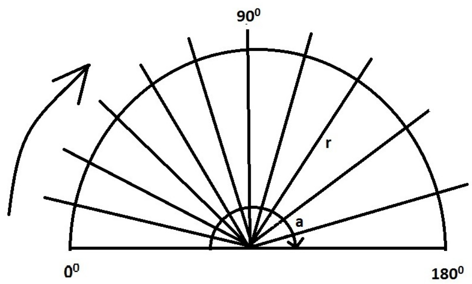



Figure 4.2: Schematic diagram of 180 ° scanning

The robot will start the operation from any one corner of an arena. For addressing the blocks in an arena, we assign the values for blocks from 0 to N in both horizontal and vertical direction defined as x and y respectively where N=0,1,2,,n as shown in the diagram . The robot will start mapping from the (0,0) block position. At this position, the robot will display the (x, y) coordinates as (0,0) and the direction as NORTH by default. Then, scanning process will start in each and every block from the starting block. The servo motor shaft will rotate from $0^\circ$ to $180^\circ$ in a semicircular orientation. The servo motor will take 22 different angles ranging from $0^\circ$ to $180^\circ$. Simultaneously, the sharp IR sensor mounted on a servo motor shaft will take 22 samples at this 22 different angles obtained by servo motor. This data is in the form of the range (r) and angle (a) will be send to the computer via zigbee wireless module operating at 2.4GHz.

where,
r=distance of the ultrasonic sensor from the obstacles in cm.
a=angular position of servo motor shaft in degrees.

Data sent from a robot will be serially given to the host PC.This data will be captured by matlab function code in host PC where predefined algorithm is constructed. According to this predefined algorithm, the matlab code function will perform the conversion from polar to cartesion co-ordinates by using the formula :

$$p = x + rsin(a) \tag{4.1}$$

$$q = y + rcos(a) \tag{4.2}$$

where,
p & q are integrated Cartesian co-ordinates.

Thus, in this way the data coming from robot will be integrated into one single map that will be displayed on matlab.

### Matrix Generation

In matrix scanning, the robot will select three samples at 0° , 90° and 180° as per predefined algorithms. The range calculated at these angles will be divided by 20 to determine the number of blocks in the forward, left and right direction. Thus, matrix will be generated by assigning '1' to the free space block and '0' to the block in which obstacles are placed. Based on this information, the robot will decide its navigation path whether it has to move in left, right or forward direction. The robot will move in a wall following mode.

Depending on the different combinations of 1's and 0's, the following table can be illustrated.

Table 4.1: Direction of a Robot

| Left(L) | Center(C) | Right(R) | Direction of a robot | Block Position |
|---|---|---|---|---|
| 0 | 0 | 0 | (turn right $\times 2$) + $Forward$ | 1 |
| 0 | 0 | 1 | turn right + forward | 2 |
| 0 | 1 | 0 | forward | 3 |
| 0 | 1 | 1 | forward | 4 |
| 1 | 0 | 0 | turn left + forward | 5 |
| 1 | 0 | 1 | turn left + forward | 6 |
| 1 | 1 | 0 | turn left + forward | 7 |
| 1 | 1 | 1 | turn left + forward | 8 |

$$\underset{\text{Current Matrix}}{\begin{bmatrix}1&0&0\\1&0&0\\1&1&1\end{bmatrix}} = \underset{\text{Previous Matrix}}{\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}} \ \textbf{OR}\ \underset{\text{Generated Matrix}}{\begin{bmatrix}1&0&0\\1&0&0\\1&1&1\end{bmatrix}}$$

Figure 4.3: Logical OR of Matrix

In this way, the robot will navigate the entire arena by avoiding the obstacles and at the same time, it will map the whole arena. At the starting point, a matrix is stored in the memory of the robot, which is initialised by zero. The robot will logically OR the current value with the previous value to give new matrix. This matrix will be sent to the host PC when robot changes its position from current block to new block. After coming to its original position the complete matrix will be sent to the host PC via zigbee.

- When the robot detects the wall/obstacle on left and front side of it, that is, when 'L' and 'C' both represents '0', the robot will take a right turn(block no 1 & 2) Ref to the figure no (4.1).

- When the robot does not detects the wall/obstacle on left side of it, that is, when 'L' represents '1', the robot will take a left turn(block no 5, 6, 7 & 8)Ref to the figure no (4.1).

- When the robot detects the wall/obstacle on left side while it does not detects the wall on the front side of it, that is, when 'L' and 'C' represents '0' and '1' respectively, the robot will move ahead in the forward direction(block no 3 & 4)Ref to the figure no (4.1).

### 4.2.2 Navigation

Navigation will be done in two steps:

1. Direction algorithm
2. (x,y)algorithm

**Direction Algorithm**

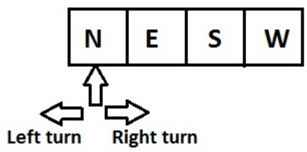



Figure 4.4: Direction pointer

Here, we first define a pointer called "NUM" in a program.This pointer will be initially pointing towards north at the start of the map. Whenever robot turns the direction,the pointer will be incremented or decremented by 'one' depending upon the direction it is facing. For every right turn,the pointer will incremented by 'one' and will move in east direction as shown in figure. Accordingly the robot will upgrade its current direction which will be simultaneously sent to the host PC for display.

### (x,y) Algorithm

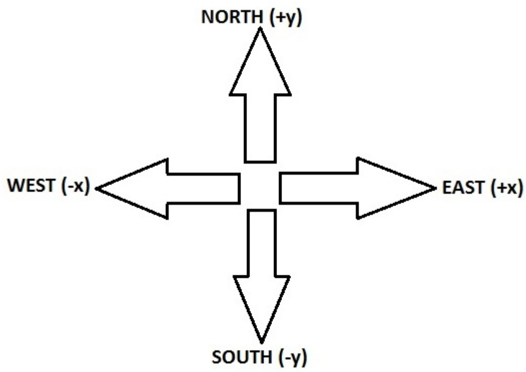



Figure 4.5: (x,y) Algorithm

Depending upon direction, the robot will calculate its current block position (x,y) value as shown in diagram.

- Whenever the robot will move in the east direction, it will increment its x co-ordinate value.
- Whenever the robot will move in the west direction, it will decrement its x co-ordinate value.
- Whenever the robot will move in the north direction, it will increment its y co-ordinate value.
- Whenever the robot will move in the south direction it will decrement its y co-ordinate value.

## 4.3 Flowchart

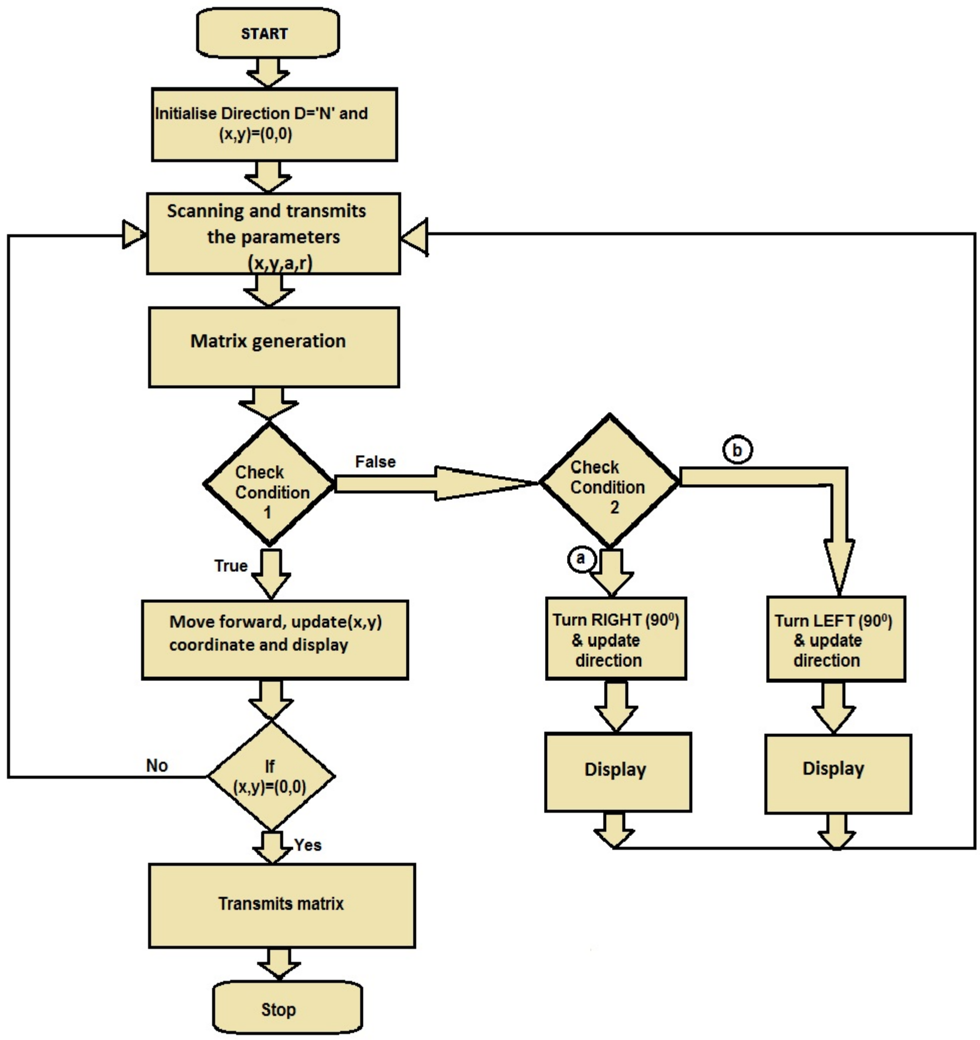



Figure 4.6: Flowchart

### Flowchart Explanation

- The robot will start its operation from any one corner of an arena. At this point, the robot will have (0,0) block position.
- Next, the robot will be facing towards 'NORTH' direction as per predefined algorithm by initializing its (x,y) as zero i.e.x=0 and y=0.
- The (x,y) block position will be transmitted to the host PC via zigbee wireless module.
- Afterwards , the robot will start scanning process. It will perform two types of scanning:
  - 180° Scanning .
  - Matrix Generation.
- The sharp IR sensor mounted on servomotor shaft will capture 22 different samples in the form of range (r) at different angles from 0° to 180°.
- The servomotor shaft will rotate in the interval of 9° such as 0,9,18,27,36 and so on.
- Thus , the parameters (x,y,a,r) will be transmitted for first block to the host PC.
- Then, the robot will take 3 sample at 0°, 90° and 180° to determine the number of blocks in left ,forward and right direction.
- Depending upon these samples the robot will assign '0' to obstacle/wall and '1' to free space block.

- Based on these 3 values ,the robot will check two condition are as follows:

  **Case 1:**

  1. First , the robot will check condition one, if it holds true, then the robot will move one step forward.
  2. If condition one is false, then the robot will check condition two.
  3. Accordingly, the robot will update its (x, y) co-ordinate as well as its direction.

  **Case 2:**

  1. If condition two is true, then the robot will move towards right direction by taking $90^\circ$ turn.
  2. If condition two is false,then the robot will move towards left by taking $90^\circ$ turn.
  3. Again, the robot will update its direction.

- Thus, the robot will display its all relevant information on the LCD display for every cases mentioned above.
- Now, the robot will follow the same procedure for the every block in an arena , until the robot will come at its original position i.e.(0,0).
- If the (0,0) condition is not satisfied , it will again start its scanning by transmitting its current direction and (x ,y) co-ordinate to the host PC.
- After wards , the robot will transmit the final matrix generated to the host PC.
- Finally, the robot will stop performing its operation.

# Chapter 5

# Testing Strategy and Data

### Arena Description

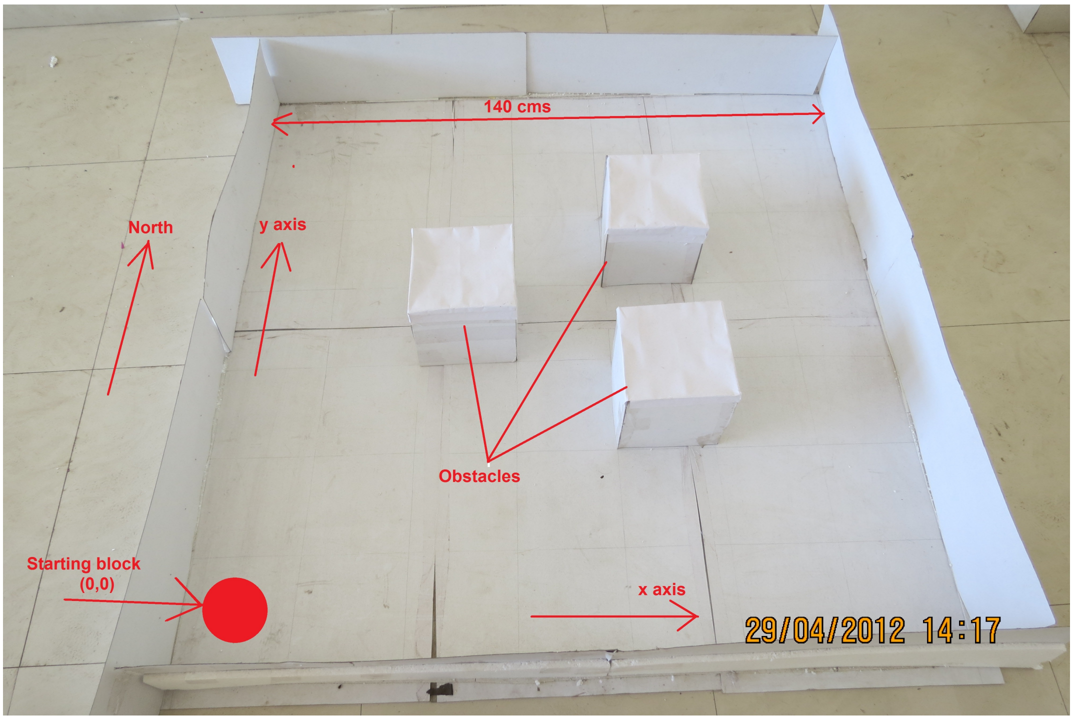



Figure 5.1: Arena Description

### Bot and Components used for Mapping

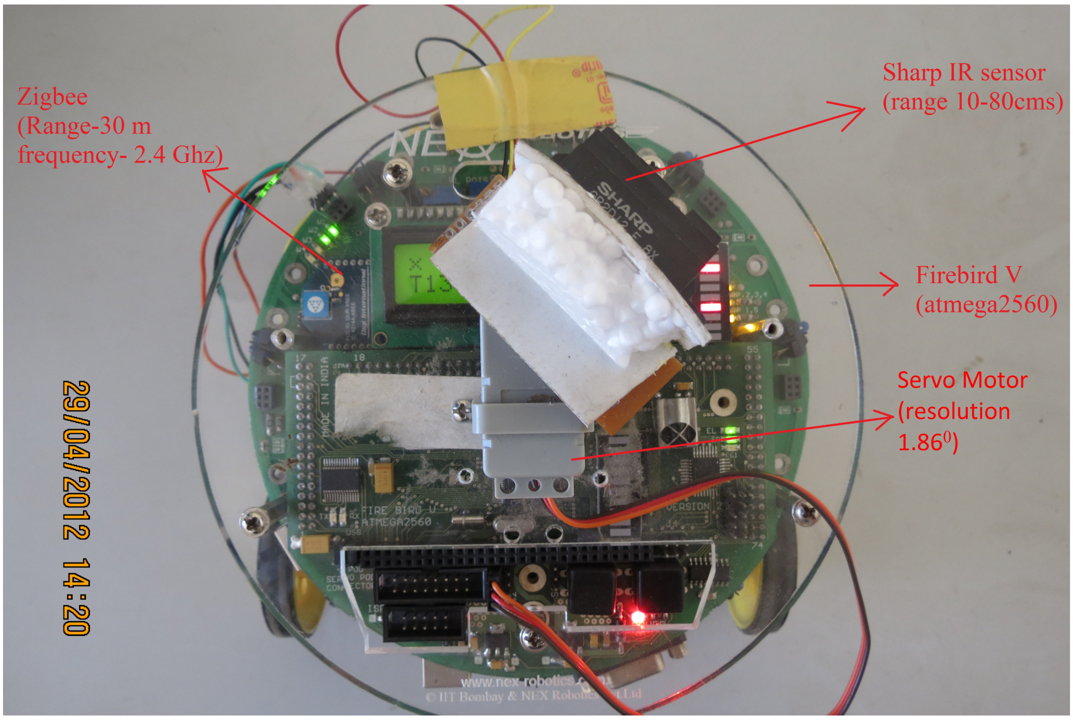



Figure 5.2: Bot and Components

### 180° Scanning

The robot performing 180° scanning using servo motor at different angles with the resolution of 9°. Accordingly, the sharp IR sensor mounted on servo motor will calculate distance from the obstacles as shown in the figure

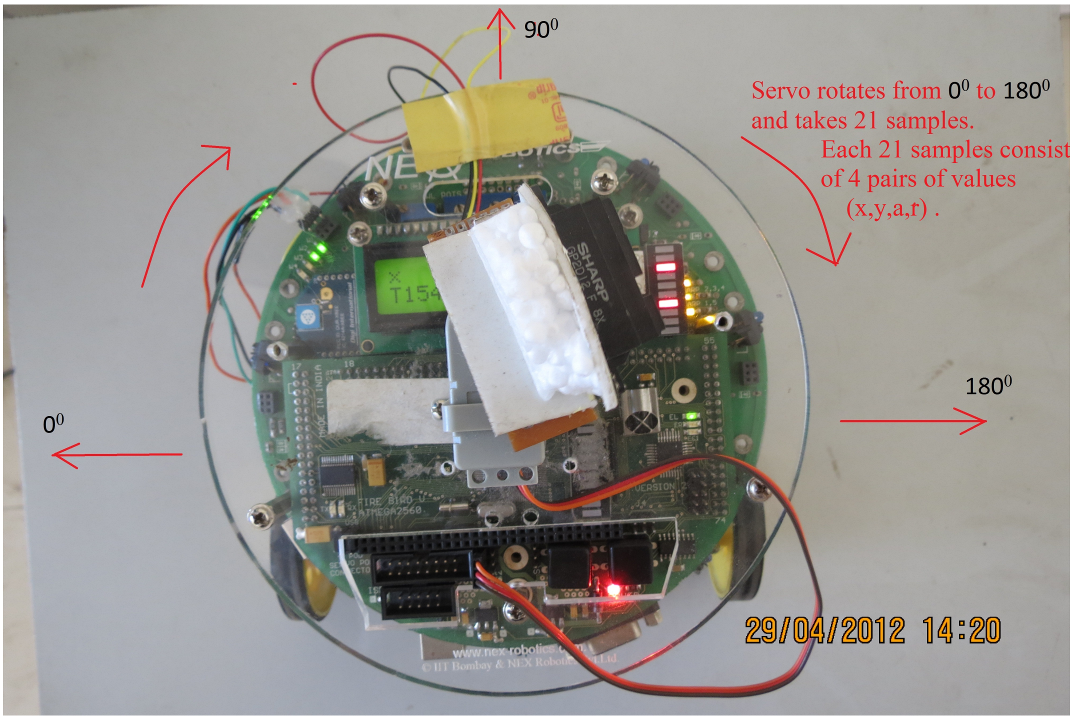



Figure 5.3: 180° Scanning

The 21 samples (x,y,a,r) transmitted from the robot through zigbee serial communication will be simultaneously received in x-ctu software in host pc as shown in the figure

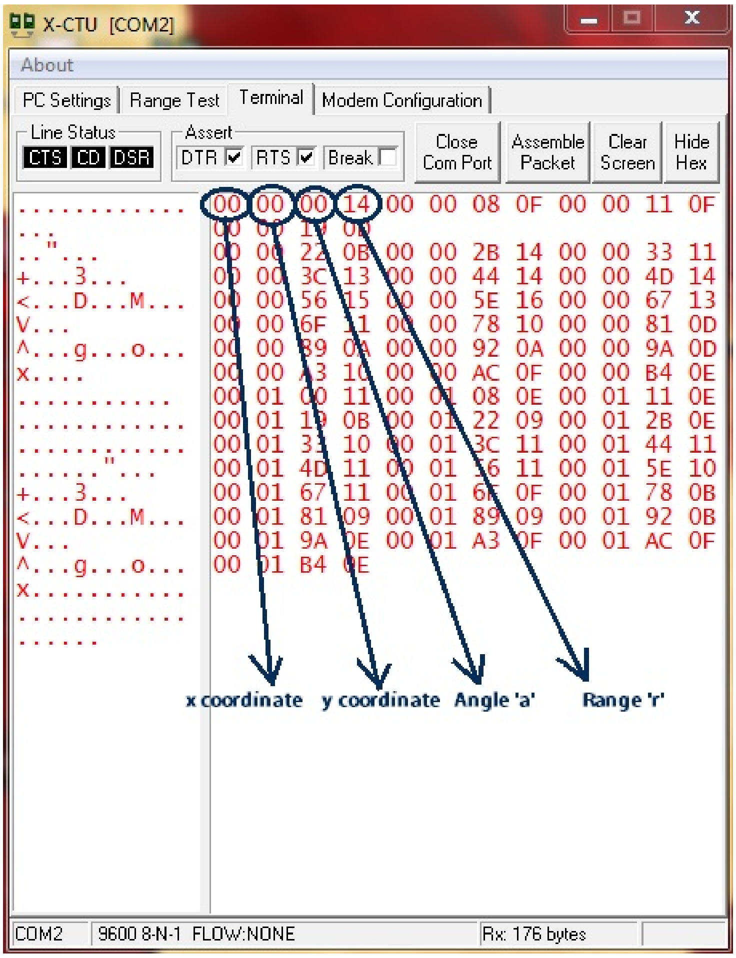
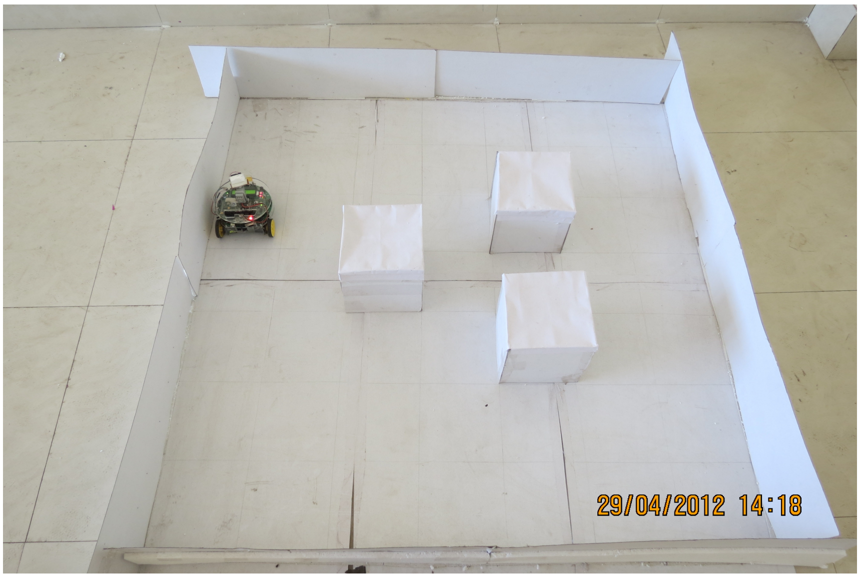



Figure 5.4: x-ctu software

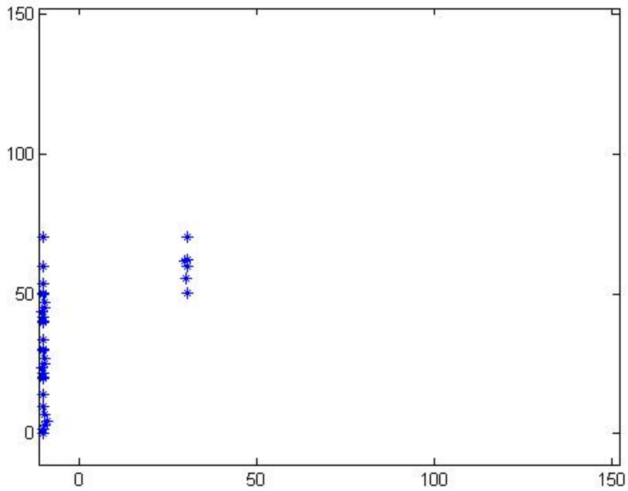
These 21 samples will be serially communicated to the matlab to plot the actual map of the arena.

### Map generated in matlab till block position (0,5)

Bot travelling from block position (0,0) to (0,5)



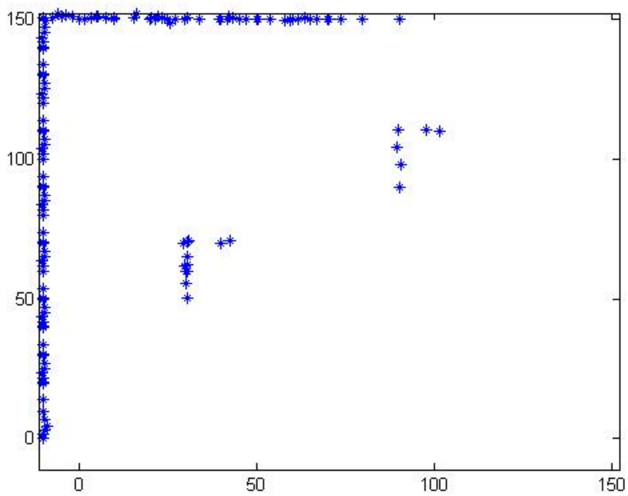

Corresponding map in matlab



### Map generated in matlab till block position (4,6)

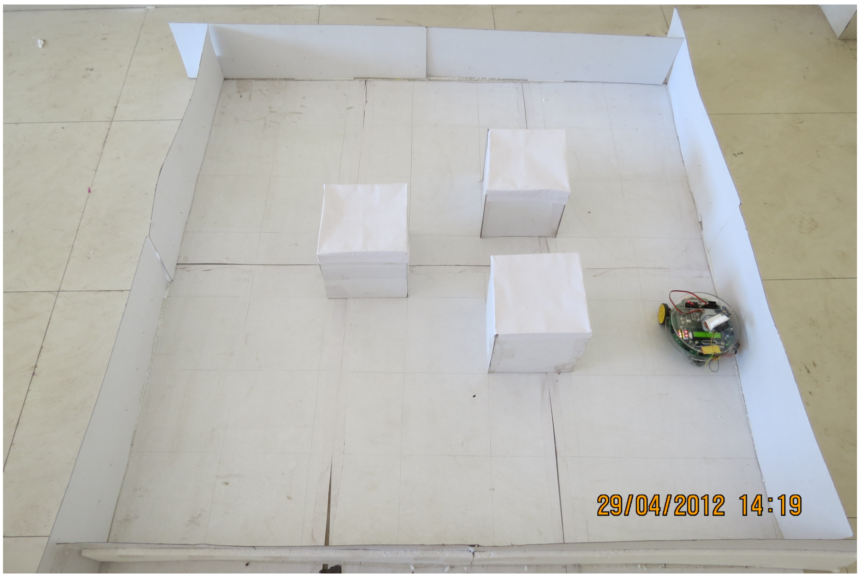
Bot travelling from block position (0,0) to (4,6)


Corresponding map in matlab

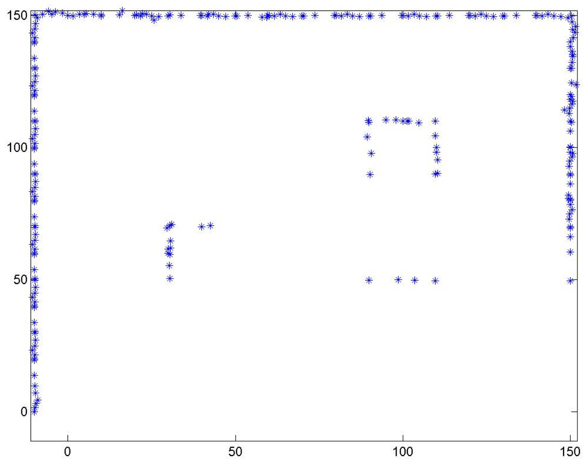


### Map generated in matlab till block position (6,2)

Bot travelling from block position (0,0) to (6,2)



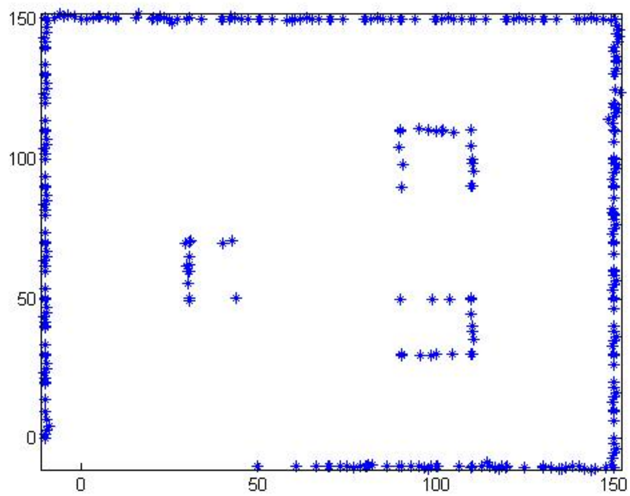

Corresponding map in matlab



### Map generated in matlab till block position (2,0)

Bot travelling from block position (0,0) to (2,0)

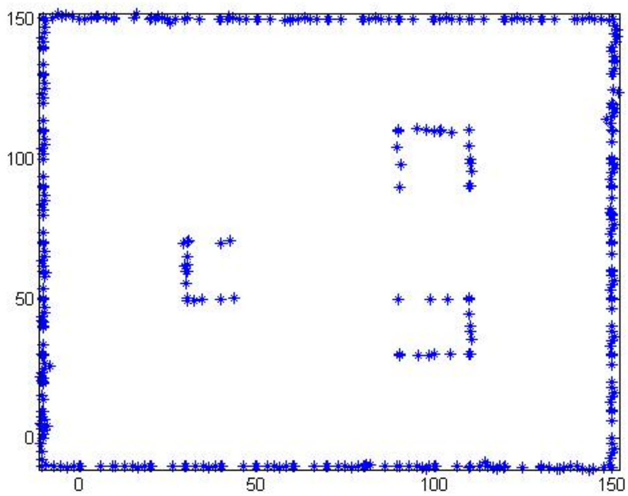

Corresponding map in matlab



### Map Generated till bot reach to original position(0,0)

Bot reached back to its original position block(0,0)


Corresponding map in matlab



# Chapter 6

# Discussion of System

1. What are worked as per plan?

   (a) Two scanning techniques:
   For mapping purposes , we are using two scanning techniques ie. 180 $^\circ$ scanning and matrix generation. Implementing these techiniques simultaneously , gives the more accurate and authentic map of an particular arena.

   (b) Increased resolution of servomotor:
   Mapping can be done efficiently , if number of samples are more . By using resolution of 9 $^\circ$ , we are getting as many as 21 samples for one block which are quite enough for good map.

   (c) Efficient navigation:
   For efficient navigation, the bot takes 3 samples at 0 $^\circ$, 90 $^\circ$ and 180 $^\circ$ to determine its navigation path. Accordingly , the robot will take decision either to move in left ,right or forward direction.

   (d) Self localisation:
   Bot always calculates its (x,y) co-ordinates which determines the actual position of bot in an arena .Thus, self localisation plays a crucial role for mapping.

   (e) Zigbee communication:
   Successfully able to communicate from bot to the host PC by zigbee for collecting the data in real time.
   The bot takes 4 samples (x,y,a,r) continuously. To map the area, integration of these samples is necessary .

   (f) Integration:
   The bot takes 4 samples (x,y,a,r) everytime and transmits it to host

PC simultaneously. To map the area, integration of these samples is necessary .

2. What we added more than that discussed in SRS?

   (a) Final Matrix map at the end:
   The bot was suppose to generate the matrix at every block and send it simultaneously to the host PC. Due to these , a delay was generated in the operation of bot which was not desirable. Therefore, we implemented the sending of matrix at the end of the operation.

3. Changes made in plan:

   (a) FIREBIRD V Atmega 2560:
   Initially, we were implementing our code on SPARK V Atmega16 . Due to the increased length of the program , we started implementing the code on Firebird where it was found to be successful.

   (b) Sharp IR sensor:
   Initially , we thought of using ultrasonic sensor for distance measurement from the obstacles. But , it gave the error reading due to its wide beamwidth. Therefore , we implemented sharp IR sensor which has a very narrow beam giving desired output.

# Chapter 7

# Future Work

Our project ”Autonomous Mapping Robot” is a prototype model based on the basic idea of mapping which can be implemented in different fields. Using these simple concept , it can be extended for advanced applications by making it a better product. Some of them are listed below:

- Magnetic compass can be mounted on robotic platform to get the desired alignment of the robot in a predefined direction.
- Using the concept of mapping , a particular area /room can be cleaned by autonomous room cleaner. For ex -product like Roomba,Navibot are available in market for performs the same functions .
- By mounting the camera on the servomotor , surveillance and security can be done using image processing.
- GPS wireless system can be installed for localizing the area of the robot .
- Android application can be used for mapping by using mobile controlled system.
- Laser beam can be used instead of sharp IR sensor which has a very large range in SONAR sensor system.
- Localisation of places can be done to carry heavy objects in warehouses .
- To cut the grasses in small farms and gardens by installing scissor type instrument on the robotic platform .

# Chapter 8

# Conclusion

An " Autonomous Mapping Robot" is an efficient product for mapping a particular room/area by localizing itself in an area. Also, it detects the obstacles and navigates itself in an area by avoiding the obstacles. Therefore ,mapping is done without human efforts which is plotted on matlab accurately. The data transmitted by robot will be sent to host PC via zigbee wireless module to give real time map of an arena.

Using these concept of mapping , the product can be used in different applications like room cleaning, security and surveillance system , mobile controlled mapping etc.

# Chapter 9

# References


- http://www.e-yantra.org/home/studio/presentations
- www.nex-robotics.com
- Book - AVR Microcontroller and Embedded Systems_Mazidi
- Firebird V Atmega2560 hardware & software manual
- http://www.irobot.com
- http://en.wikipedia.org/wiki/Zigbee.
- pdf document - Autonomous Robot Navigation of Corners with Uncertain Sensor Information Using Fuzzy Control.
- pdf document - autonomous mapping robot.