Mobile Robot for Warehouse Automation

Hasan Habib, Saad Waseem, Abdul Hanan Ghafoor and Dr. Waqar Ahmad
Department of Electrical Engineering
National University of Computer & Emerging Sciences
Islamabad, Pakistan
hasan.habib@live.com, swaseem192003@gmail.com, ahannan.ghafoor@gmail.com,
waqaar.ahmad@nu.edu.pk

***Abstract*—Mobile Robot for Warehouse Automation is design of a system comprising of Mobile Robots and a Control System to move and deliver racks from one place of warehouse to another in an efficient and effective manner. The control system assigns the task to the Mobile Robot from a remote Server Room. The Robot navigates its way through a grid system based on RFID tags placed on the warehouse floor at equidistant positions. Each RFID tag represents a unique location in the warehouse which is virtually mapped in Control system software.**

***Index Terms*— MOBILE, ROBOT, AUTOMATION, CONTROL, RFID, NAVIGATION**

I. Introduction

THE "Mobile Robot for Warehouse Automation" is to implement and devise a system for warehouse automation. The robot system is based on a very unique and advanced navigation technique. The idea is to equip the warehouse floor with High Frequency RFID (Radio Frequency Identification) tags, a mobile robot equipped with a RFID reader is able to navigate its way through the warehouse to lift, place or deliver racks from one place of the warehouse to other by making use of a very intelligent and Powerful Navigation Algorithm Software fed into a Remote Server.

II. Problem Statement

Traditionally, goods in a warehouse or any other distribution area are moved around using conventional techniques such as Conveyer systems, Forklifts, Automated Storage and Retrieval Systems, Guided vehicles, etc. Most of these techniques require human input and effort this creates inefficiency as it increases inflexibility of material handling systems and also increases cost of order fulfillment. While the other automated systems though efficient but requires a large Capital and Maintenance investments.

In MRWA's approach, items are stored and placed on portable racks. The Warehouse Floor is equipped with RFID tags and each tag represents a unique location in the warehouse. When an order is entered into the Control system by an Operator, the software locates an idle Mobile Robot, Calculates the shortest possible path to the location and directs the Robot to place or retrieve the item. The mobile Robot navigates its way around the warehouse by following a series of RFID tags on the floor. When the Robot reaches its location, it slides underneath the rack and lifts it off the ground. The robot can then either carry the rack to some new location or to a human operator to pick or place required items.

III. Flow Diagram of Complete Concept

At the very beginning the operator assigns a task with the source and destination locations on his remote Server. The Server generates a shortest path to the destination pick node and transmits it to the Mobile Robot.

The Robot now having received the path, scans its current position and compares with the path assigned and starts moving accordingly. The robot repeats this step until it reaches its destination.

Once the robot reaches its destination it lifts the rack up after this the robot communicates with the server to assign it with the destination drop node path. The server now transmits the destination drop node path.

Once having received the destination drop node path the robot compares it with its scanned position at each new position until it reaches its destination [1].

After reaching its destination drop node the robot drops the rack, and this completes one process cycle.

The Following Figure 1 provides a flow chart view of the Concept.

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456".

The next few paragraphs should contain the authors' current affiliations, including current address and e-mail. For example, F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@ boulder.nist.gov).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).

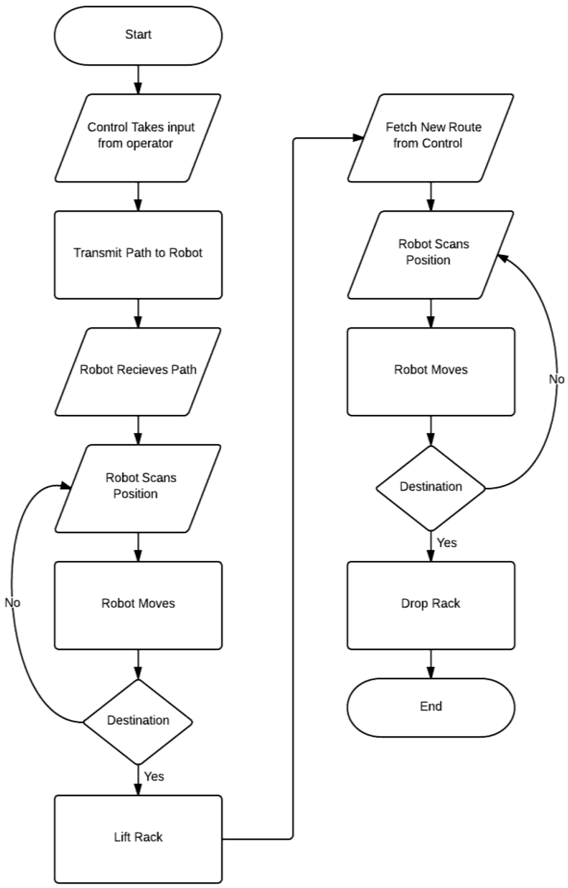

Figure 1 Flow Diagram

IV. Technical Design & Description

The robot system consists of three parts:

- The RFID Equipped Warehouse Floor
- The Mobile Robot
- The Control Part

These parts are further classified according to their design and tasks. The following Figure shows the main block diagram of the robot system.

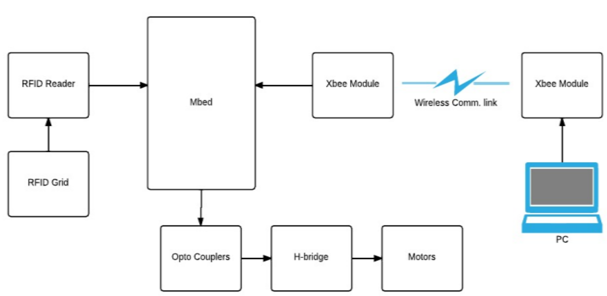

Figure 2 Main Block Diagram

A. The RFID Equipped Warehouse Floor

The Warehouse or any other distribution facility or place first needs to be equipped with RFID tags at exact equal intervals, this factor basically depends upon the size of Racks and the size of Warehouse.

The Basic reason behind the use of RFID based floor is to virtually map warehouse locations. Every RFID tag has a globally unique ID. These ID's are used to virtually map unique locations in the warehouse.

The RFID tags are to be placed in a grid structure on the facility floor such that each RFID tag is equidistant to its neighboring node. For instance for a RFID tag with ID 1 needs to be equidistant to node 2 with ID 2 at its right side and to node X at its bottom that has an ID X.

Figure 3 provides a 3 dimensional grid view for understanding the placement of RFID tags.

The RFID tags are to be placed at each intersection of two or more than two lines.

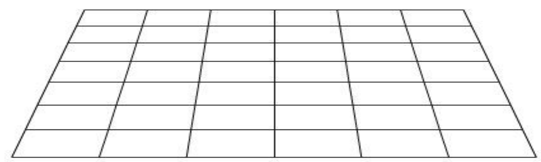

Figure 3 Three Dimensional view of a Grid

For our robot we have used a grid of 25 nodes that is a 5x5 formation.

1) RFID Tags

Radio-frequency identification (RFID) is the wireless non-contact use of radio-frequency electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information. Some tags are powered by and read at short ranges (a few meters) via magnetic fields (electromagnetic induction). Others use a local power source such as a battery, or else have no battery but collect energy from the interrogating EM field, and then act as a passive transponder to emit microwaves or UHF radio waves (i.e., electromagnetic radiation at high frequencies). Battery powered tags may operate at hundreds of meters [1].

We have used Passive RFID MIFARE tags within the HF 13.56 MHz range. These RFID tags provide a line of sight detection range of up to 1.5 inches. Figure 2-3 shows the RFID 13.56 MHz Mifare cards used for the floor Grid.

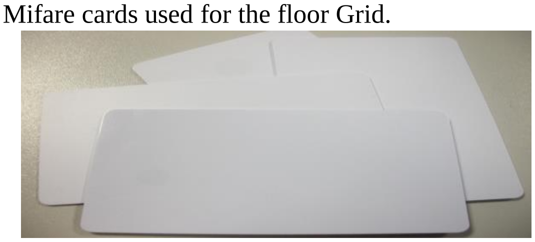

Figure 4 RFID MIFARE 13.56 MHz cards

B. *Mobile Robot*

The Robot part consists of both the hardware and mechanical structure. It includes the Robot structure, Motors, Lift Mechanism, RFID Sensor, Mbed, Wireless communication module, H-bridge and the Robot direction Algorithm [3].

1) *Robot Structure*

The Robot structure is made up of 4mm Acrylic sheet (a plastic polymer). It provides a light weight low cost alternative to aluminum and other metallic sheet structures. A thickness of 4mm was selected to avoid bending of sheet under some external force or weight.

Secondly the Robot Mechanically is a Circular differential type Robot. With two Wheels along the central axis and one Omni directional (ball caster) wheel.

For providing the Circular structure, the Acrylic sheets were precisely cut to circles with a diameter of 14 Inches each. There are two Acrylic sheets one that provides the base of the Robot structure and one to provide a platform for the lift mechanism.

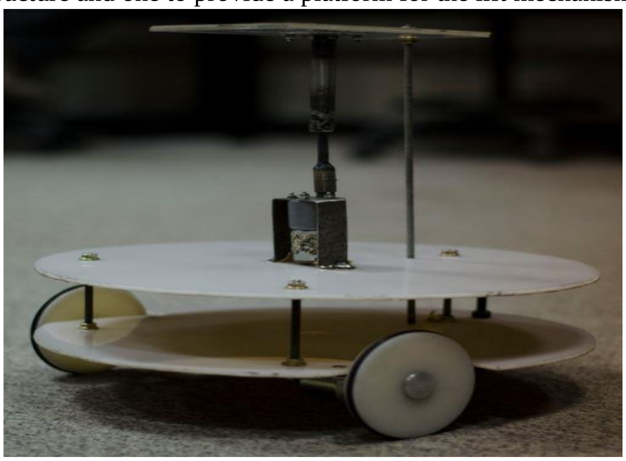

Figure 5 Robot Structure

2) *Omni Directional Wheel*

An Omni directional wheel also known as a ball caster was used to provide stability and 360 degree ease of turning to the Mobile Robot. The wheel consists of an adjustable rod through which it is mounted to the Robot base. At the bottom of the rod there is a hollow jacket which holds a metal ball caster, which is free to rotate just like a ball bearing. The height of each tire is 1.5 inches from ground to the base which is further adjustable. Figure 6 shows the actual Omni wheel used [2].

Figure 6 Omni Directional Wheel

3) *Teflon Tires*

Custom build Teflon tires where used to provide rotational motion to the Robot. The Tires are made up Teflon material 3.25 Inches in Diameter. The tires are coupled to their respective Dc motors and fixed along the ends of the central axis. A rubber belt is provided on the circumference of the tires that ensures grip on slippery surfaces.

4) *DC Motors*

Two 12 V Dc Geared motors of 43 Rpm where used for the movement of the Robot. Each motor is 12V Dc motor which takes 0.9 A to 1 A at full load. Another similar geared motor with higher Rpm of 96 Rpm was used for the lift mechanism. The motor speed can be further varied using PWM. Figure 2-7 shows the Motor used.

Figure 7 12 V DC Geared Motor

5) *Cork Screw Lift Mechanism*

The Robot is equipped with a lift mechanism to lift up racks in the warehouse. The Robot needs to slide down a rack and then lift it up from under it. This requires a lifting technique mounted over the robot structure of small size so that the Robot can peak under a rack and the lift it up.

For this a cork screw action lift mechanism design was used. As the name suggests, the lift mechanism consists of a rotating screw that holds a bolted lift on it. The clockwise and anticlockwise rotation on the screw lowers and raises the lift respectively. The screw is rotated by using a specially designed mount that is coupled to a Dc motor. The Dc motor needs to be of high RPM so it can lower or raise the lift within seconds. A supporting rod is attached to the lift base and the Robot base so when the screw rotates the lift base stays stable and doesn't rotate.

Figure 8 Cork Screw Lift Mechanism

6) *H-Bridge*

An H bridge is an electronic circuit that enables a voltage to be applied across a load in either direction. These circuits are used in robotics and other applications to allow DC motors to run forwards and backwards [2].

H bridges are available as integrated circuits, or can be built from discrete components.

In our circuit we have used an integrated H-Bridge IC L298N. The IC provides two H-Bridge per IC with a maximum current rating of 2A. Another H-Bridge IC L298N was used to provide rotational control for the lift mechanism motor.

The H-Bridge IC was coupled with Opto-couplers EL-817 at the input, and fly back diodes at the output.

Opto-couplers are used to isolate the Microcontroller from the H-bridge, while Fly back diodes are normal diodes that are connected across output terminals to isolate the motors from the H-bridge this eliminates the chance of IC damage by the back EMF generated by the motors.

A Voltage regulator LM7805, was also used to provide 5V logic supply voltage to the H-bridge.

Figure 2-9 shows the Circuit Design for one H-bridge. The figure includes LM7805 Regulator, EL817 Opto-Couplers and L298N H-Bridge IC.

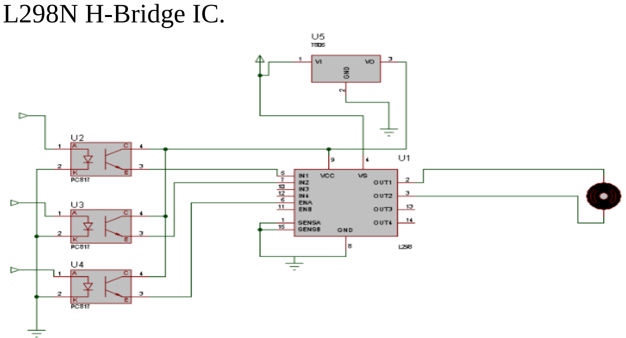

Figure 9 H-Bride Circuit

7) RFID Sensor

To read RFID tags we require a RFID tag reader of that specific frequency Range. Since we used High Frequency HF passive RFID tags of 13.56 MHz we require a RFID reader of the same Frequency.

We used Strong link Mifare SL040 USB RFID reader for reading the RFID tags placed on the facility floor. The reader is a HID Human Interface Device. A HID device is a plug and play device that is basically designed and configured to use with personal computers.

To interface the HID RFID Sensor we first had to implement driver software on the microcontroller so the device can be integrated with it. For this we used a special library for Mbed "USB host Keyboard".

This library provided driver headers for a HID keyboard device. We used this library and wrote some special functions to interface the RFID reader to the Mbed Microcontroller. Figure 2-10 shows the HID SL040 RFID reader. The reader has dimensions of 65x46 mm and provides a detection range of 1.5 inches line of sight.

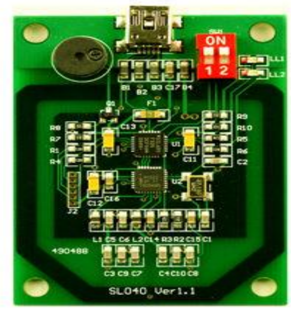

Figure 10 SL040 HID RFID Reader Module

To interface the module physically with the Mbed some special configuration technique has to be used. In case of normal sensor module the connection are simple as that of a serial communication link. But since the module was a USB HID device we had to connect the device to Mbed USB D+, D- pins. The D+ pin from the module is connected to the D+ pin of Mbed and the D- Pin of the module to the D- pin of the Mbed through a pull down network. A pull down resistive network is required for the Microcontroller to act as an input. The +5 V of the module is connected to Vout pin of Mbed that provides 5V regulated voltage this is enough to power up the reader module. The Gnd terminal of the reader module is connected to the common ground. Figure 10 shows the physical connection Diagram.

8) Xbee Module

To communicate with the Control Server the Robot requires a hardware and medium for communication. The communication medium cannot be wired since the Robot requires high mobility. So the medium of communication should be wireless.

Xbee Module is hardware module that provides HF wireless link for fast and error free wireless communication. It work on the UHF band of 2.4GHz and provides speeds upto 250Kbps.

We have used Xbee series 1 module for our robot. It provides communication range of upto 330 ft or 91 meters line of sight. It uses the IEEE 802.15.4 protocol for PAN communication shows the Xbee S1 module.

The connection of Xbee Module is very simple. The Din pin of Xbee is connected to serial Tx pin of the Mbed, the Dout pin of Xbee is connected to the serial Rx pin on the Mbed. The Xbee Vin is connected to 3.3V pin of the Mbed to provide regulated 3.3V supply to Xbee. The Gnd terminal of Xbee is connected to a common Ground. Figure 11 shows the Xbee S1 Connection to Mbed.

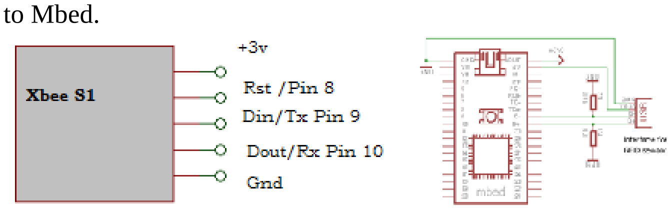

Figure 11 Xbee S1 Connection Diagram

9) Li-Po Battery

To provide uninterruptable power on the go to the Mobile Robot we used a 12v Li-Po battery. Lithium ion polymer batteries are rechargeable dry cell batteries. Li-Po batteries are usually composed of several identical secondary cells in parallel to increase the discharge current capability, and are often available in series "packs" to increase the total available voltage.

These batteries though expensive but are long lasting and requires little or no maintenance at all. The battery used provided a backup time of almost 3 hours once fully charged. Figure 12 shows 12V Li-Po battery used for the Robot.

Figure 12 12V Li-Po Battery

10) Mbed Microcontroller

The Mbed Microcontroller is the main part or the brain of the Mobile Robot it handles and controls all the hardware of the Robot through Software.

The mbed NXP LPC1768 Microcontroller in particular is designed for prototyping all sorts of devices, especially those including Ethernet, USB, and the flexibility of lots of peripheral interfaces and FLASH memory. It has a built-in USB FLASH programmer. This makes easier for the user as to burn new code into the Microcontroller the user just has to copy and paste the code similar to the file transfer in a USB.

It is based on the NXP LPC1768, with a 32-bit ARM Cortex-M3 core running at 96MHz. It includes 512KB FLASH, 32KB RAM and lots of interfaces including built-in Ethernet, USB Host and Device, CAN, SPI, I2C, ADC, DAC, PWM and other I/O interfaces. The pin out above shows the commonly used interfaces and their locations. Note that all the numbered pins (p5-p30) can also be used as Digital In and Digital Out interfaces.

The mbed Microcontrollers provide experienced embedded developers a powerful and productive platform for building proof-of-concepts. For developers new to 32-bit microcontrollers, mbed provides an accessible prototyping solution to get projects built with the backing of libraries, resources and support shared in the mbed community.

a) Official Mbed.h (C/C++ libraries)

The Mbed.h library provides the C/C++ software platform and libraries to build your applications.

One of the most concerned functionalities we used out of Mbed.h was Digital I/O it enables the user to input and output data from Mbed I/O pins. The second most used function was the timer "wait" function. This function provides a wait for a specified period of time. It also provided us with built in Xbee module functions such as readable and writable.

Digital I/O

Digital Out - Configure and control a digital output pin.

Digital In - Configure and control a digital input pin.

Pwm Out - Pulse-width modulated output.

Time and Interrupts

Wait – wait for a specified time.

Xbee Communication

Xbee.readable( ) – This reads wireless signal information.

Xbee.writable( ) – This enables the user to send information a wireless signal.

b) USBHostKeyboard.h Library

This library was used to interface the HID RFID reader module to the Mbed. This library provides software translation for HID Keyboard device drivers for Mbed.

The most concerned functions provided by this library were

- KeyBoard Connected – Detects and connect HID Keyboard Device.
- KeyBoard Disconnected - Detects and Disconnects HID Keyboard Device.
- Event Detected – Detects any input Event
- OnKey – On reception of an Event scans and detect the RFID tag.

11) Coding for Mobile Robot

Mbed provides a C++ based coding environment to its users. It provides an online compiler where the user can write and compile his code. For our Mobile Robot the Brain is the Mbed controller that controls, assigns and monitor task to all hardware components of the Robot.

The code written for controlling the Robot is of around 500 lines (appendix). It starts from trying to connect and configure the HID RFID reader module. Once the HID device is connected to the Mbed the Mbed keeps reading Xbee for any task that the operator may assign to the Robot [6].

The instructions from the Operator are received in a form of array. The starting numbers in the array represent the shortest path to the destination and the last number defines the task that the robot needs to carry. A number 27 represents that the operator wants the Robot to pick up the rack from the destination node. While a number 28 at the end represents that the operator wants the robot to pick the Rack from its source node and drop it on the destination node. A number 1 represents simple movement from source to destination.

Since we are using a 5x5 grid the maximum number of nodes in a path are 17 I.e. the path is (1-2-3-4-5-10-15-14-13-12-11-16-21-22-23-24-25) when nodes (6,7,8,9,17,18 ,19 20) are obstacles. So the maximum length of the array size for a 5x5 matrix will be 18 i.e. 17 path nodes and one end number for task definition. Figure 14 explains the concept.

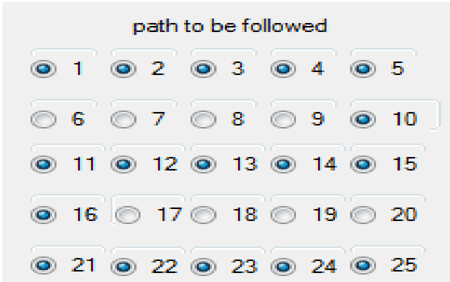

Figure 13 Path and array size

Once the path and the task has been received the Robot responds. The Mbed Compares its initial position by scanning the RFID tag with the first node of the array. If the Source node matches the code proceeds else it waits for the operator to rectify the source and destination nodes.

After successful comparison of the first node, Mbed identifies the next node from the the path received and applies another "Direction Algorithm" to locate the exact side of the node placed in the Real world. The Mbed accordingly provides 90,-90,180 degree turns for moving to the right node. The robot moves to the next node and again confirms it by scanning the RFID tag and then matching it with path iteration. This process continues till the robot reaches its destination node.

Once the Robot Reaches its destination node. The Mbed performs the assigned task i.e. lower or raise the lift. This Completes one process cycle, the Robot now again goes into its scanning mode to receive any new orders from the operator. The Figure 15 shows a flowchart which describes the style and coding technique used to control different hardware components of the Robot.

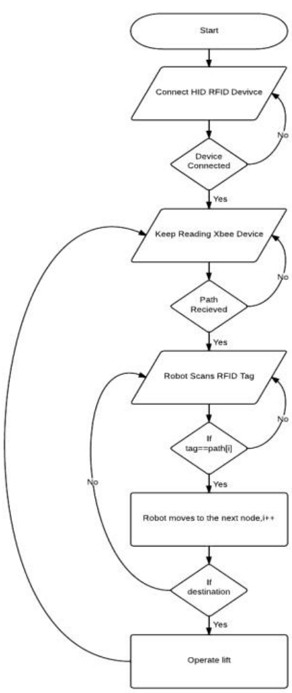

Figure 14 Mbed Code Flowchart

12) The Direction Algorithm

Once the Robot receives the shortest path array from its operator the robot does not knows its next location in the virtual world but the Robot doesn't know the exact location of the next node in the real world. For instance a path array denotes the next location to be node 2 from its initial position 1. For this particular instance the robot does not know if node 2 is present on its right side or left side or at top or bottom. To solve this problem a Direction algorithm was designed and implemented on the Mbed to locate the exact location of nodes in the Real world.

For our particular 5x5 RFID grid the 4 next node possibilities for the robot are:

- Next node location=Current node location +1
- Next node location=Current node location +5
- Next node location=Current node location -1
- Next node location=Current node location -5

Since we are not using diagonal movement the maximum no. of next node possibilities for any NxN grid will remain 4.

For the first time a Robot is put into the facility floor the Robot is set and oriented at a reference direction. We have used node 1 as a reference point and west direction as a reference direction. Each node increases by 1 when we move from node 1 towards west. Similarly movement towards south will increase each node by 5.

So now if the next node in the path array is 2 the robot knows it has to move straight to reach 2 and incase if the next node in the path array is 6 the robot knows It needs to make a -90 degree or right turn before moving straight to reach node 6. Figure 16 shows the Tag placement and the reference directions.

Figure 15 Direction and Tags placement

13) Overall Circuit Diagram

The Combine Circuit Diagram for the Mobile Robot is shown In Figure 2-20. The power supply for the H-bridge is taken from the 12V Li-Po battery while the Mbed is powered from a different source. The Mbed then provides voltage supply to both RFID sensor module and the Xbee module.

The supplies are set different to provide isolation and ensure the Mbed stays safe from any back emf produced by the motors. The H-bridge for lift mechanism is not shown in the diagram.

due to limitation of space.

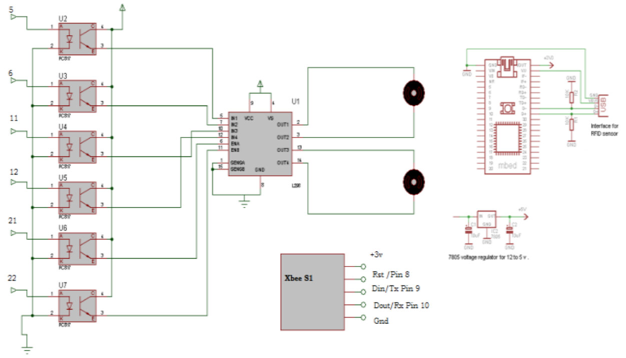

Figure 16 Overall Circuit Diagram

Figure 18 shows the actual circuit for the Mobile Robot.

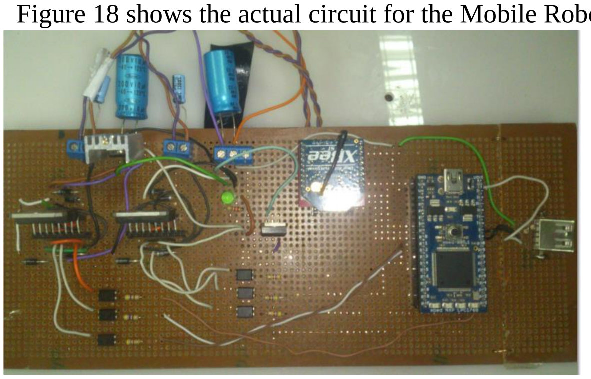

Figure 17 Actual Circuit of the Robot

C. Control Part

In an automation system, usually thousands of machines are working to get the work done fast and efficiently without any error. Same is the case with MRWA. This scope of this robot can be expanded to include thousands of robots in order to improve the efficiency. Since all the robots are expected to transport racks to different parts of the warehouse, it can be said that each robot will have different source and destination point. They should also follow different path in order to avoid collision if they are all to work at the same time. if only one robot works at one instant, while all other robots wait for their fellow bot to complete its task, this will make the automation process in the warehouse extremely inefficient and pointless as this will be very time consuming. So in order for the robots to work in a Swarming action, there should be some intelligence between them to make sure all the robots in the warehouse are working on different tasks at the same time. This will also help prevent assigning of one task to different robots which will create the confusion among the robots and due to which the whole system will fail.

Now there are two ways to accomplish this intelligence.

Every robot exchanges information with every other robot in the warehouse.

Implement a fixed and isolated control server, which tells the robots of their task.

1) First Approach

The first approach, as described before, involves the robots to exchange the information with one another. This information exchanging can be done by Xbee wireless modules which work on radio frequency and distinguish each other with distinguished private area network (PAN) and specific PAN-ID, IDs allotted to each module within the private area network just like the IP addresses. The problem with this approach is that even if the fastest possible Xbee module is used to transfer that information, it will create time delay as robot 1 will first transfer information to the robot 2, which will then transfer to robot 3 and so on. This information exchanging is done until the last robot is reached which may be thousandth robot. The information may include the path, source and the destination of the robot. If there are thousand robots working, this information package will contain information about each robot, making the package heavier which means it will take a large amount of time to send and receive, making the automation system inefficient.

2) Second Approach

The approach which is applicable in this case and the one we have used is to implement a control server. The control server is wirelessly linked to all the robots using Xbee modules. In this case the server will only transfer information to a specific robot to which it wanted to assign some task thus eliminating the need to send information to all the robots in the warehouse. The control server also has a database which gets update after the robot picks or places a rack. The control server takes the path decision by consulting the database first. So an algorithm can be implemented on the control server to choose a path that enables the robot to do the least amount of work to reach the destination in the least possible time. The database on each robot is also cumbersome as robot will require extra memory device which will increase the cost of the automation system which is also undesirable.

The control section consists of three main parts:

- Navigation Algorithm
- Gui Interface
- Wireless link

The explanation of each part is done in the following section.

a) Navigation Algorithm

(1) Objective

The aims and purpose of the navigational algorithm is to give intelligence to the robot. since the purpose of the warehouse is to store items and in our proposal of warehouse automation, the floor of the warehouse will be lined with RFID cards and the items to be stored will be placed on the racks on the RFID cards. Also since the movement of the robot is based on the RFID cards, so there are multiple paths present between the source , where the robot is standing initially, and the destination, the node from the robot has to pick items or deliver them. The important question is now which path to choose and what should be the criteria of selecting the path. The navigational algorithm comes handy for this problem. Not only it provides the robot with the intelligence of selecting a particular path, but also it helps to avoid obstructions and obstacles. It also helps the robot to take the directional decisions. Also it tells the robot to decide when to turn its motors, what should be the speed of them, in which direction and for how long they are to be moved. In short, the navigational algorithm is the brain of the robot similar to human brain whose function is to control human's body parts, the navigational algorithm controls the robot's parts.

As told before, the purpose of the path calculation algorithm is to choose the best possible path for the robot. The criterion for selecting the path is as follows:

The path must have the least number of nodes

The path must contain the least number of turns.

The first criterion is used to ensure that the path selected would be the shortest path. And the second criterion is used to ensure that the robot has the minimum processing delay due to the path. Since more turns in the path means that the robot has to stop for each turn, make a turn on the same axis and then moved forward to the next node. Stopping and turning cause delay and therefore should be avoided.

(2) Shortest Path Algorithms

There are number of shortest path algorithms available such as

- Bellman-Ford algorithm
- A* search algorithm
- Dijkistra algorithm

All the algorithms decide the path based on the distance between the two nodes in order to reach the destination node. Since all the RFID tags are placed in the form of the grid in which the nodes are equally spaced, therefore these algorithms fail at this point. In order to work out an algorithm which can select a shortest path from a grid of equally spaced nodes, a fusion of dijkistra theorem and A* search algorithm, which is actually the extension of dijkistra algorithm, is used.

(a) Floor Map

The floor consisting of nodes is virtually mapped first in the code as it gives information about the nodes to the algorithm. It tells the algorithm about the interconnection of nodes with each other. This is achieved by imagining the nodes as the elements of a 2 dimensional Boolean array. The rows of the array represent the total number of nodes present on the floor. The columns of the array describe the nodes the row nodes are connected to. So for 25 nodes on the floor, the array made will be of 25x25 size as shown in the Figure 2-22.

```
= {{0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//0
{1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//1
{0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//2
{0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//3
{0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//4
{1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//5
{0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//6
{0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//7
{0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//8
{0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},//9
{0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0},//10
{0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0},//11
{0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},//12
{0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0},//13
{0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0},//14
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0},//15
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0},//16
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0},//17
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0},//18
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1},//19
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0},//20
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0},//21
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0},//22
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1},//23
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0},//24
```

Figure IV-18 A 25x25 Array for each node

Take for example row 0. Now it contains all zeros except of 2 ones, which are on the column 1 and column 5 position. This means that the node 0 is connected to node 1 and node 5 in the grid. Similarly row 1 contains 1 in 0,2 and 6 column location which means that the node 1 is connected to the node 0, node 2 and node 6.

Another approach for mapping the floor is the use of linklists. The linklist are structural nodes which carry data and a pointer which points to the next node. These nodes are used as simple variables in the program. However in this case, linklist provide a difficult approach to work with because for different type of interconnection between the nodes, different structure programs are required. There are usually 3 types of interconnection.

2 nodes interconnection in which the nodes are connected to 2 nodes

3 nodes interconnection in which the nodes are connected to 3 nodes

4 nodes interconnection in which the nodes are connected to 4 nodes

After describing the interconnection, the node instances are to be created. Since nodes instances are created using different structure programs. They cannot be grouped in an array; therefore the algorithm code cannot be generalized. So in this case it is better to use the array matrix approach as compared to the linklist approach.

(b) Theory

Dijkistra theorem is the most commonly used shortest path algorithms. It works by constructing a path tree from the first vertex to the every vertex in the graph. It then calculates the shortest path by considering the total distance from one vertex to other vertex. Algorithm, for example, starts from the initial position and look for the connected nodes. Here in this case the initial node is node 1. Figure 2-23

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Figure IV-19 Algorithm Explanation 1

Then the algorithm looks for the connected nodes. Figure 2-24.

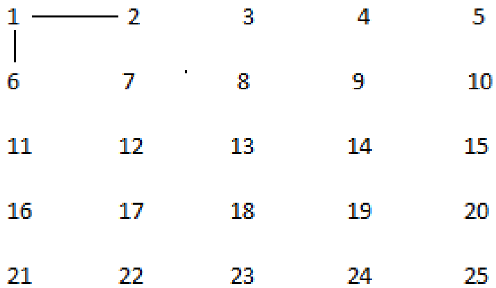

Figure IV-20 Algorithm Explanation 2

For that the algorithm check for the 1's in the array, and save the corresponding column locations into another array. algorithm also saves the iteration number as the cost to reach a certain node from the initial node. After finding the connected nodes, in this case node 6 and node 2, the algorithm then finds the nodes connected to nodes 2 and node 6 and saves the costs to reach those from the initial location i.e. node 1. This is shown in the diagrams below. Figure 2-25 to 2-32.

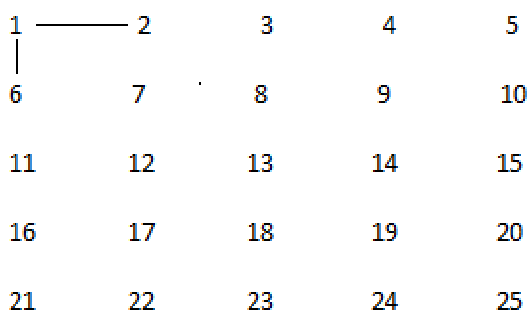

Figure IV-21 Algorithm Explanation 3

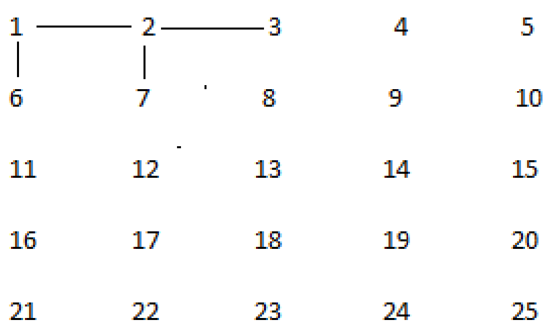

Figure IV-22 Algorithm Explanation 4

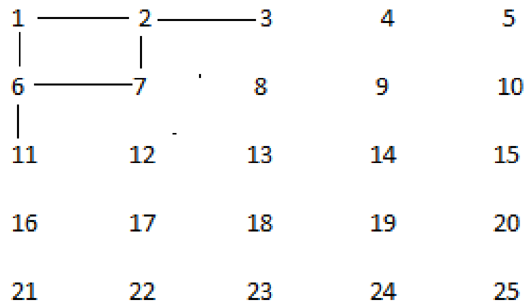

Figure IV-23 Algorithm Explanation 5

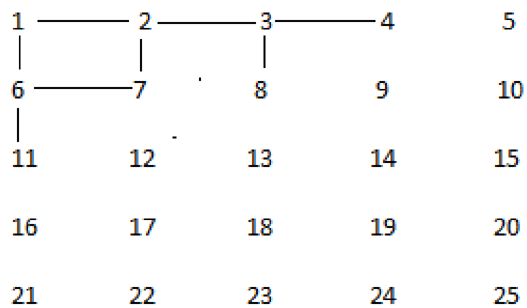

Figure IV-24 Algorithm Explanation 6

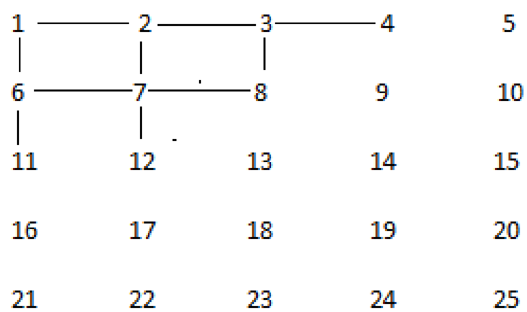

Figure IV-25 Algorithm Explanation 7

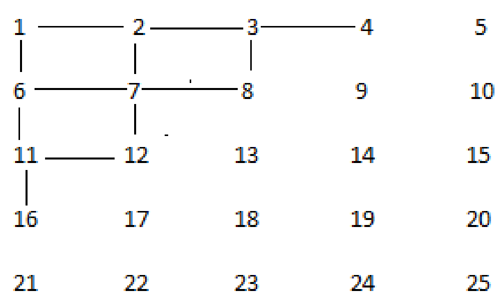

Figure IV-26 Algorithm Explanation 8

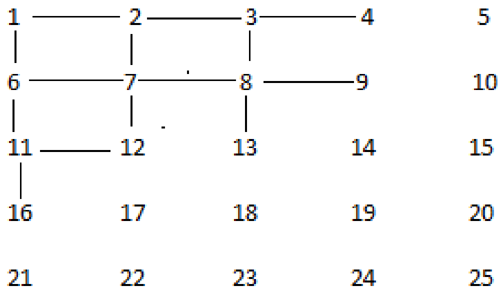

Figure IV-27 Algorithm Explanation 9

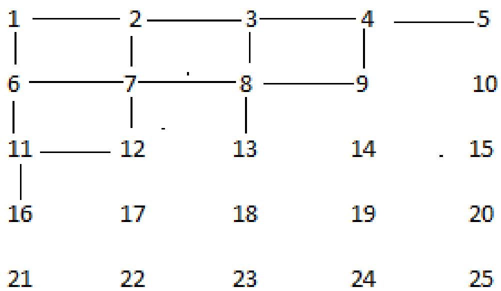

Figure IV-28 Algorithm Explanation 10

The information about the distance from the node 1 to other nodes is saved in another array. that array also includes the information about the nodes which come in the path.

(c) Obstacle Detection

Obstacle detection is also the main part of the shortest path algorithm. The obstacles create hindrances for the robot and may affect the robot's path for reaching the destination. Also if the obstacles are avoided on the run time when the robot is already following a path, the resulting new path will be bigger.

There are 2 types of obstacles

- The racks on which the items are stored.
- Other robots or any person which a robot may find in its path.

Only first type of obstacle detection is addressed here. Since the control server is giving commands to the robot about which rack to be picked up and which rack to be placed at a node, it knows all about the racks which might act as obstacle for certain robot's path. The control server updates its database as racks are picked and placed. The node on which there is a rack is taken out of the floor grid in the program. So for example if there is rack on node 2, the node is taken out of the floor maps by putting a zero in corresponding locations in the array which means, now, that node 2 is now not connected to node 1, node 3 and node 7. Now the row 2 and column 2 will both be zero. As it is not connected to other nodes, the algorithm will not include this in its path. Figure 2-33 shows the concept.

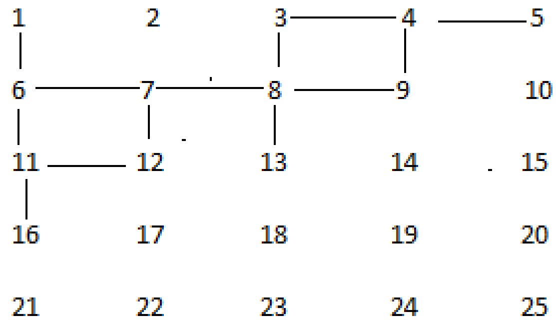

Figure IV-29 Obstacle Detection Case

b) GUI Interface

The GUI program provides the interface between the user and the control server. It provides the interacting mechanism for the user through which user can communicate to the control server. The user in this case will be the person controlling the control server or in simpler terms, the person who wants to pick a rack from one location and place in the other location of the warehouse. It is made on C++ forms application on Visual Studio 2008 express edition. Figure 2-34 shows the GUI interface.

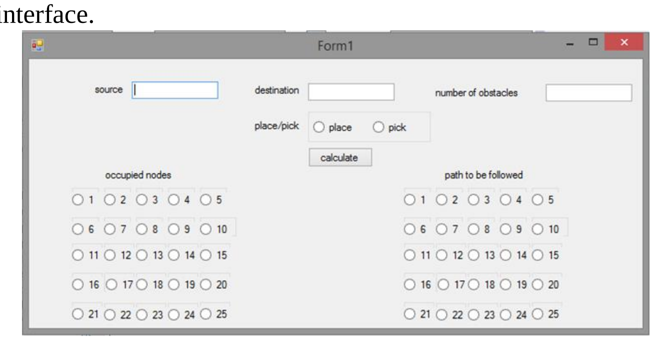

Figure IV-30 GUI Control Interface

Here the user are provided with 4 options , source node, destination node, number of obstacles and whether that item is to pick or place.

As this is a prototype, the GUI also gives an option to the user to select occupied nodes. Occupied nodes actually indicate the nodes on which there are racks and because of that, they have to be taken out of the system. When the user hit the calculate button the path to be followed will be displayed on the right side of the GUI interface and the path is sent to the robot via wireless link.

Figure 2-35 shows the functional view of GUI.

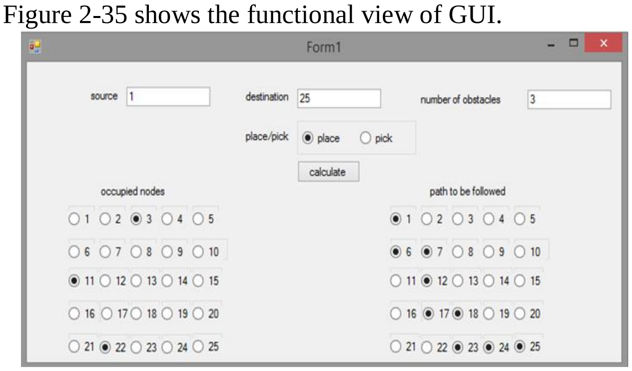

Figure IV-31 Functional View of GUI

c) Wireless Link

The wireless link in the automation system is done by using the Xbee module. Xbee is a wireless information exchange module which works on the frequency of 2.4 GHz and has the data rate of 250 kbps. The different modules within the private area network created by the Xbee connected to the control server are differentiated by the PAN-IDs. The Xbee modules are configured by the XCTU software by Digi. The Xbee modules are USB compatible modules. Which means the information will be exchanged serially. In the control part the serial port will be opened first. The path to be sent is stored in an array whose size is 15. The size of the array depends on the maximum number of nodes that can be present in a path which in this case will be 14. The last index of the array contains the option for the place or pick i.e. in order to pick a rack the number 26 will be sent and in order to place an object the number 27 will be sent. The baud rate used for serial communication will be set as 9600 which is a standard for every USB compatible device. Also since the array is of int data type and the serial communication supports char and string data types. The array was first converted to char by incrementing each value in the array by 48 which is done by consulting the ASCII table.

To interface the USB Xbee module with C++ , Serial communication code for C++ was used.

3) Windows app

A windows phone based application was also designed for the purpose of assigning task to the robot via a windows phone. The app was made on C# platform and it used TCP/IP connection to the Control server through a network connection. The app provided the liberty to the operator to command the robot from other than the control server. Figure 2-36 shows the Gui interface of the app during testing on a softphone.

Figure IV-32 Windows app Gui interface

V. Results

The robot was able to move objects from one place to another. The Control Part was used to assign path and task to the Robot. The Control Server was equipped with a C++ GUI software interface for providing an interactive and user friendly platform for the operator to use it with ease. At its back end the software had advanced "Navigation algorithm" as defined in the previous chapter. The Navigation algorithm calculated the shortest possible route to the destination. The Control Server was serially linked with a Xbee Usb module. C++ serial Communication code was used to interface the Xbee with the Control software.

Once the path was calculated and transmitted to the Mobile Robot, the robot used a series of steps including the "Direction algorithm" to move accordingly in real world warehouse floor until it reaches its destination. The steps followed by the robot to fulfill the task assigned are already mentioned in chapter 2. The following Figure 3-1 shows the Robot processing a task.

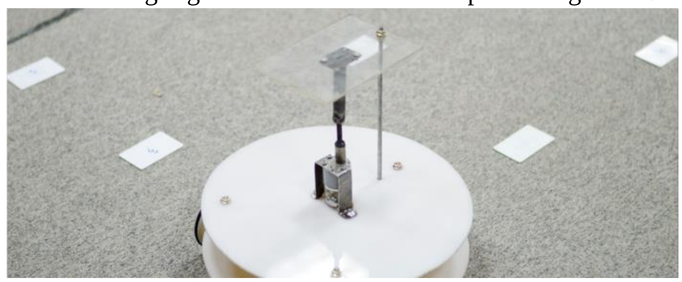

Figure V-1 MRWA Navigating on the Floor.

Though the robot was completely functional it still had some precision problems. The precision problems as mentioned in Chapter 2 forced the Robot to leave its path and never reach the next node. As already mentioned a considerable amount of time and effort was spent for rectifying the problem but there still remains a chance of error.

Work in Future can be done to completely eliminate the fault by use of high precision IMU Sensors and encoder motors.

Future Enhancement & Applications

A. Recommendations

Since our main focus was to design and implement a complete prototype solution. Design of Navigation Algorithm was the focal point. For this only a small amount of attention was given to the hardware design of the robot. We used low cost, light weight and lower strength materials for our robot structure. This was enough for a prototype model, but to be realistic as the robot is made up of acrylic sheet it cannot handle load above 3 KG. Furthermore the low cost design also contributes to the overall precision error. So our recommendations for anyone interested in improving the robots are:

- Professional Design
- Precision
- Swarming

1) Professional Design

Since our model is a prototype design, it has its own limitation. Slow speed and low load carrying capability, Furthermore the low cost mechanical design also contributes it part to the precision movement error. So we would recommend that a new professional design of Robot to be implemented. We will recommend use of Aluminum to provide high strength and tolerance to load. Use of Aluminum sheet will enable the user to use the Robot in real world applications. With increase of weight high power motors will also be required. So for that High power DC geared encoder motors can be used. The new design should be precise by a single cm as very small dissimilarities in the Robot mechanical structure contribute a large error [4] .

2) Precision

The only problem we had to face in our robot was the precise movement issue. A small error can grow so large that it completely removes the robot from its path.

We recommend that the user might use IMU sensors for exact angle measurement for rotation and movement. Also encoder motors can also be used along with it to calculate exact number of steps for a rotation. Other techniques such as use of PID controller can help to make the robot precise.

3) Swarming

Last but the most important recommendation is to provide swarming facility.

Swarm intelligence is inspired by the insect colonies, the way they interact with each other and perform certain tasks. Swarm robotics is the application of SI (swarm intelligence) principles to the control of groups of the robots. SI is an artificial intelligence based technique to study collective behavior in self-organized, decentralized systems. SI systems include population of simple robots or agents interacting locally with each other and with their environment (arena). Examples of systems like this can be found in nature, including ant colonies, honey bees, bird flocking, animal herding, and fish schooling.

As the cost of robotic hardware has come down and availability has gone up, there has been growing interest in robotic systems which are composed of multiple simple robots rather than one highly-capable robot. This tradeoff reduces the design and hardware complexity of the robots and removes single point failures, but adds complexity in algorithm design. The challenge is to program a swarm of simple robots, with minimal communication and individual capability, to perform a useful task collectively [4].

Introducing Swarm intelligence in MRWA will increase further efficiency and effectiveness of the Warehouse, since now more than one Robot will work in coordination to complete different tasks in parallel [4].

4.2 Applications

The MRWA robot has a number of applications. The main purpose and application of the robot as the name suggests is provide warehouse Automation, MRWA is more efficient as compared to traditional warehouse automation techniques.

Traditionally, goods in a warehouse or any other distribution area are moved around using conventional techniques such as Conveyer systems, Forklifts, Automated Storage and Retrieval Systems, Guided vehicles, etc. Most of these techniques require human input and effort this creates inefficiency as it increases inflexibility of material handling systems and also increases cost of order fulfillment. While the other automated systems though efficient but requires a large Capital and Maintenance investments.

MRWA provides low Capital investment warehouse Automation solution and eliminates cost of human resource by 95%.

Other applications include:

- Radioactive Material Handling.
- Online-shopping.
- Library management.
- Localization

VI. Conclusion

Mobile robot is able to move goods from one place of warehouse to other by navigating itself using RFID cards placed on the floor grid. Robot is controlled by central control unit, which guide robot about loading, unloading, shortest path and point of destination. Further studies are needed to find mechanism which allow Robot to read RFID tags faster, automation in software can also be done to make it intelligent. Advance mechanism of tracking using wireless technologies can be used to detect movement of robot.

VII. References

1 E. Guizzo, "Three Engineers, Hundreds of Robots, One Warehouse," *IEEE Spectrum,* 2 July 2008.

2 E. Ackerman, "Robots Use RFID to Find and Navigate to Household Objects," *IEEE Spectrum,* p. 2, 2014.

3 "mbed LPC1768," MBED, [Online]. Available: https://developer.mbed.org/platforms/mbed-LPC1768/. [Accessed 4 Jan 2014].

4 K. S. a. M. A. N. F. Arvin, "Design of a Differential-Drive Wheeled Robot Controller with Pulse-Width Modulation," in *Innovative Technologies in Intelligent Systems and Industrial Applications,* Sunway, 2009.

5 K. V. T. Sowjanya, "Cascaded H-Bridge with Single DC Source and Regulated Capacitor," *International Journal of Advanced Science and Technology,* vol. 73, pp. 89-102, 2014.

6 "Wheel Control Theory," Robot Platform, [Online]. Available: http://www.robotplatform.com/knowledge/Classification_of_Robots/wheel_control_theory.html.. [Accessed 5 Feb 2014].

7 N. Smith, "Picky Robots Grease the Wheels of e-Commerce," *Business News Daily,* 2 June 2011.

8 A. S. R. W. R. N. N.R. Hoff, "Two Foraging Algorithms for Robot Swarms Using Only Local Communication," in *IEEE International Conference on Robotics and Biomimetics*, New Jersey, 2010.

9 M. Dorigo, "Swarm Intelligence and Swarm Robotics," FNRS Research Director IRIDIA Université Libre de Bruxelles, Bruxelles, 2009.