CS-684 Project Report
Rocky Field Mapping and Ploughing Plot

Team Name : Innovation
Team No : 11

133050084 Pradip Tivhale
133059010 Pankaj Gaikwad
133050072 Shyam Kumar
133050056 Naveen Kumar

November 13, 2013

Contents

List of Figures

List of Tables

1 Introduction

1.1 Existing System

At present there is no existing system that helps in the detection of rocky area on the field and plot the location of the rocky area on a GUI. As ploughing is one of the important agricultural activity. The ploughing tool can get damaged due to rocky area.

1.2 Proposed System

The proposed system has to identify the rocky area, plot the location of the rocky area on a GUI and plough the remaining field. It is one kind of application of minesweeper algorithm. Instead of mines, we are trying to identify rocks in the ploughable field.

1.3 Final System

The final implemented system can

- Detect the rocky area in a grid arena using the Reed switch mechanism
- Plot the location of the rocky area using GNU plot

1.4 Definitions, Acronyms and Abbrevations

1 FireBird: A robot indigenously designed at ERTS laboratory, IIT Bombay [1].

2 AVR-libc: Standard C library implementation by AVR System [2]

3 Rocky area: It usually refers to naturally occurring solid aggregate of minerals and/or mineraloid

4 GUI: Graphical User Interface (GUI) is the one which helps in depiction of location of objects

2 Problem Statement

Rocky Field Mapping and Ploughing Plot aims at implementation of an system that identifies the rocky area and plotting the rocky area . In additional it also has to plough the field using the shortest route and skipping the rocky area

3 Requirements

3.1 Hardware Requirements

- FireBird V Bot
- REED switches
- zigbee cards (Wireless Communication)
- Powerful Magnets
- 10k Ω resistor.
- Three port wire for connection

3.2 Software Requirements

- AVR Boot Loader
- AVR Studio
- Python 2.7.5 or higher for Windows machine
- GNU plot for Windows machine
- Terminal(BR@Y's or X-CTU) to record communication log

3.3 Functional Requirements

1. Identifying Rocky area
2. Localization and plotting of map with grid
3. Localization without grid (sub goal)
4. Decide the best path for ploughing

3.4 Non Functional Requirements

1. Field Area to be covered
2. Size of the grid
3. Mapping Mechanism
4. Energy Consumption
5. Cost

3.5 Design Constraints

1. **Grid Arena:** For the system design purpose, the grid based approach is being used. Hence, the 6X6 or 7X7 fixed size grid should be used. Restricting the grid size may be helful for the rigerous testing of the system. Also, the size of the grid to be covered should be much enough for which Bot's battery can last while traversal.

2. **Black Lines:** For the working of bot, it has to traverse the grid. hence it need some direction guidelines. Here, black lines of the grid serves this purpose. As the firebird V have whiteline following sensors, the grid should be either black lined on white surface or whitelined lined on black surface. Nothing else can be useful here.

3. **Detection Ways:** Fixing the detection mechanism on either sides of bot can make both ways detection possible. While placing the magnets for testing purpose, one should be aware of this fact, unless detection mechanism fails.

4. **Detection Direction:** The detection direction inherits from the traversal path. Hence it should not void the traversal within grid.

5. **Traversal Direction:** As the bot is following black lines, traversal in diagonal direction not possible. The bot can move only forward,backward,left,right directions. The North, West, South, East should be directives forward grid square positions.

4 Implementation

We have divided our complexity of design into 3 modules

Module 1: Design of the detection mechanism which identifies the rocky area using REED switch

Module 2: Make the BOT to follow the black line and cover the entire grid

Module 3: Get the position of the rocky area and plot the data

4.1 Rocky area Detection

To detect the rocky area we used REED switch .

Functioning: When the magnetic field is applied to the REED switch, the switch will be in open mode and voltage becomes zero . Using the micro controller pins(OC1C ie. Port B7 and Port B6) we detect whether the switch is open or closed. We designed our circuit on two PCB's so that it'll cover both the sides of the grid line in a single way increasing the efficiency of the BOT i.e more area covered will be in less duration. The circuit diagram is shown in figure

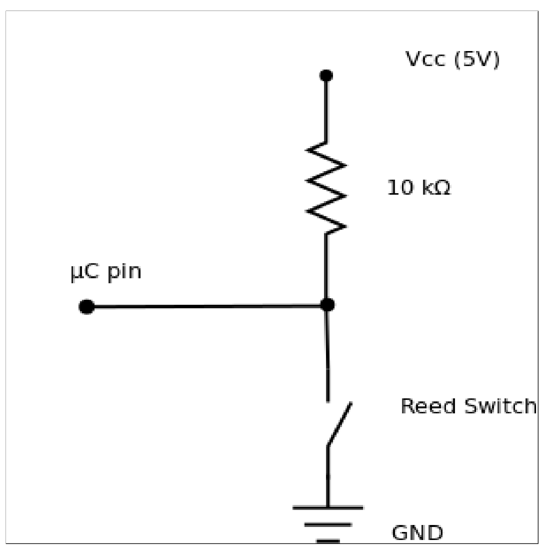

Figure 1: Rocky Area Detection Circuit

4.2 Grid Follower

To make the BOT cover the grid,we used black line follower to cover the entire arena. We used black cross lines as the check points which will be helpful in localisation of the BOT. For every checkpoint the position of the BOT increments by one in one direction and decremented in other direction. The position of the rocky area is sent using Zigbee communication

4.3 Plot

The data we got using zigbee communication is displayed contiguously on terminal , we used terminal logger to capture the log data . Then we used python script and GNU plot to create GUI which displays the position of rocky area
Here are some primitives of FireBird V which are used in our project

Primitive	**Use**
uart0_init(void)	Initialization of zigbee module
buzzer_pin_config (void)	Configuration of Buzzer pins
buzzer_on()	Turn the buzzer ON
buzzer_off()	Turn the buzzer OFF
adc_init()	Initialization of ADC hardware
adc_pin_config (void)	ADC hardware pin configuration
ADC_Conversion(unsigned char Ch)	Conversion of unsigned char into Hex value to display on LCD
lcd_port_config (void)	LCD hardware port configuration
print_sensor(row,coloumn,channel)	Function To Print Sesor Values at Desired Row And Coloumn Location on LCD
lcd_cursor(row, column)	Place cursor at given row and column on LCD display
lcd_string(data)	Write null terminated data on LCD display
motion_set (Direction)	Function to set wheels motion with given direction argument
motion_pin_config (void)	Function for configuration of motion pins
left_position_encoder_interrupt_init (void)	Enable interrupt 4 for left wheel encoder
right_position_encoder_interrupt_init (void)	Enable interrupt 5 for right wheel encoder
ISR(INT4_vect)	Interrupt service routine for INT 4
ISR(INT5_vect)	Interrupt service routine for INT 5
timer5_init()	Timer 5 initialized in PWM mode for velocity control
orient(int value)	Decoding the path variable value for appropriate grid traversal
follow()	Function containing instructions to follow while traversing grid

Table 1: Firebird V HAL Primitives

5 Design Diagrams

5.1 FSM and Statecharts

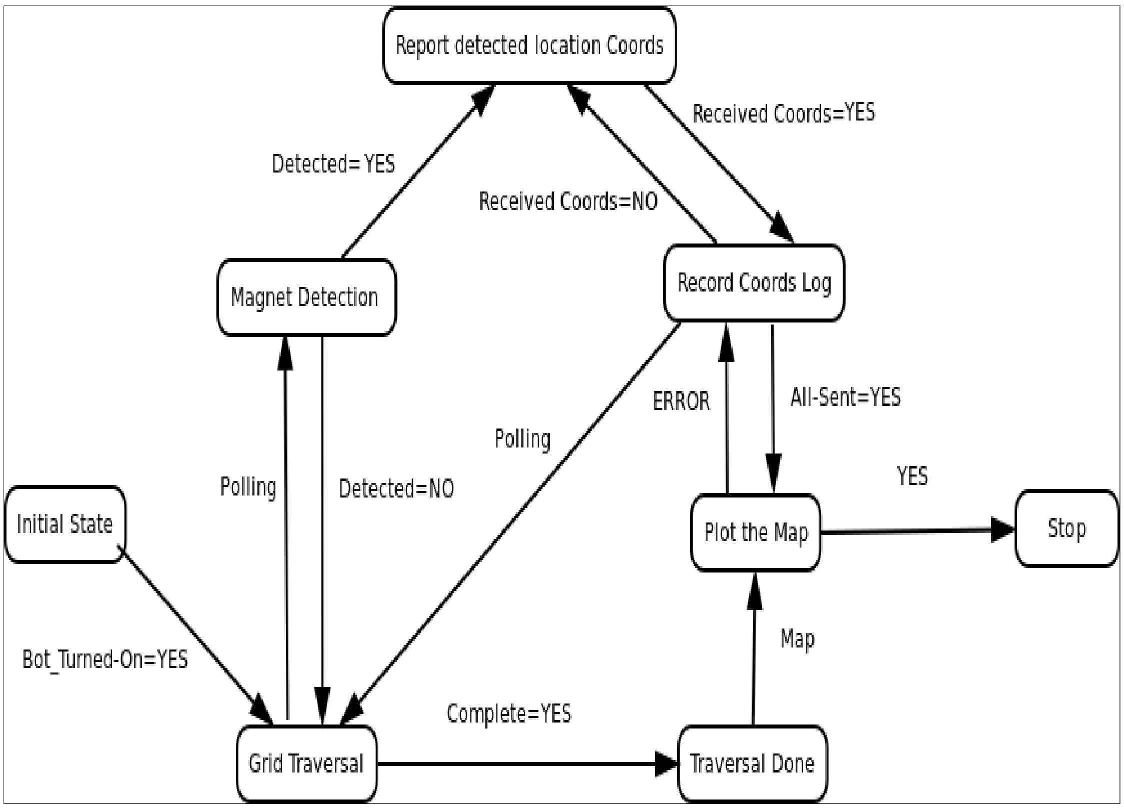

Figure 2: Finite State Machine for Detection & Mapping

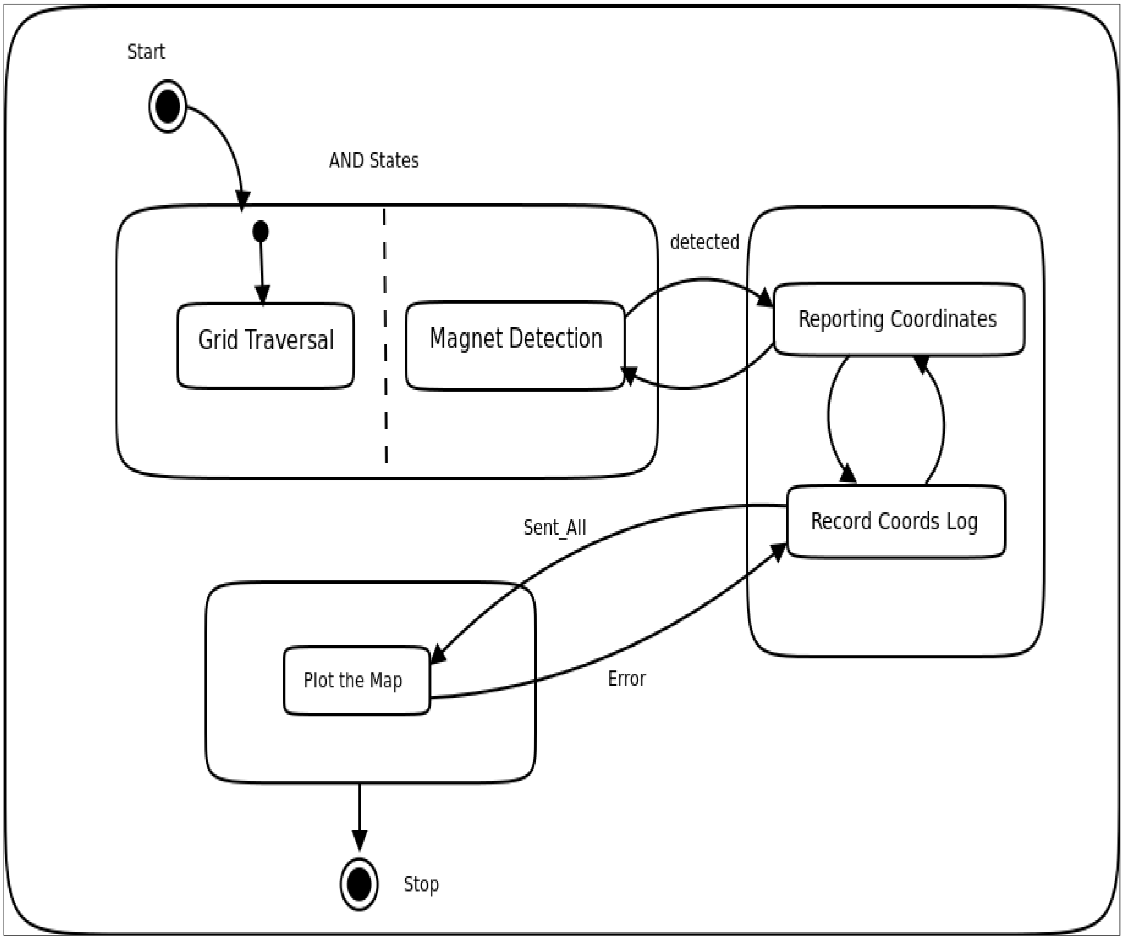

Figure 3: Statechart Diagram

6 Testing Strategy and Test Data

Here we tested our design continuously as we used Iterative and prototype based model
We used simple phenomena for testing ”Build and Test Module wise”

6.1 Testing Circuit

Initially we tested the correctness of the individual components followed by correctness of the circuit which is shown in 1 using the magnets.We tested all circuits which are located on both sides of BOT and checked for the message displayed on the LCD like ” Magnet detected on left hand side ” , ”magnet found on right”,”both sides”.While testing we calculated the sensing distance of the circuit which is found as 15 - 20mm.It depends on the power of the magnet

6.2 Testing Grid Follower

We tested our BOT to check whether it’s following the black line and covering the entire arena or not . Initially we tested the blackline follower sensors and noted down their readings . We found that the readings of the sensor changes from time to time as the readings depends on the intensity of the light, because of this BOT not following the black line when the intensity of the light is low . We adjusted the forward distance and turning angle so that the BOT will follow the blackline . We also tested for the checkpoints at the cross junction of the blacklines which is helpful in localisation of the BOT

6.3 Verifying the correctness of the PLOT

Verified the correctness of the map which is displayed on the machine by comparing with the exact location of the rocky area on the field by placing the magnets at the different positions on the 6x6 grid

6.4 Test Data

Our test data includes

- Magnets are treated equivalent to ROCKS in the field.
- Grid is used to depict agricultural field where ploughing is to be done.

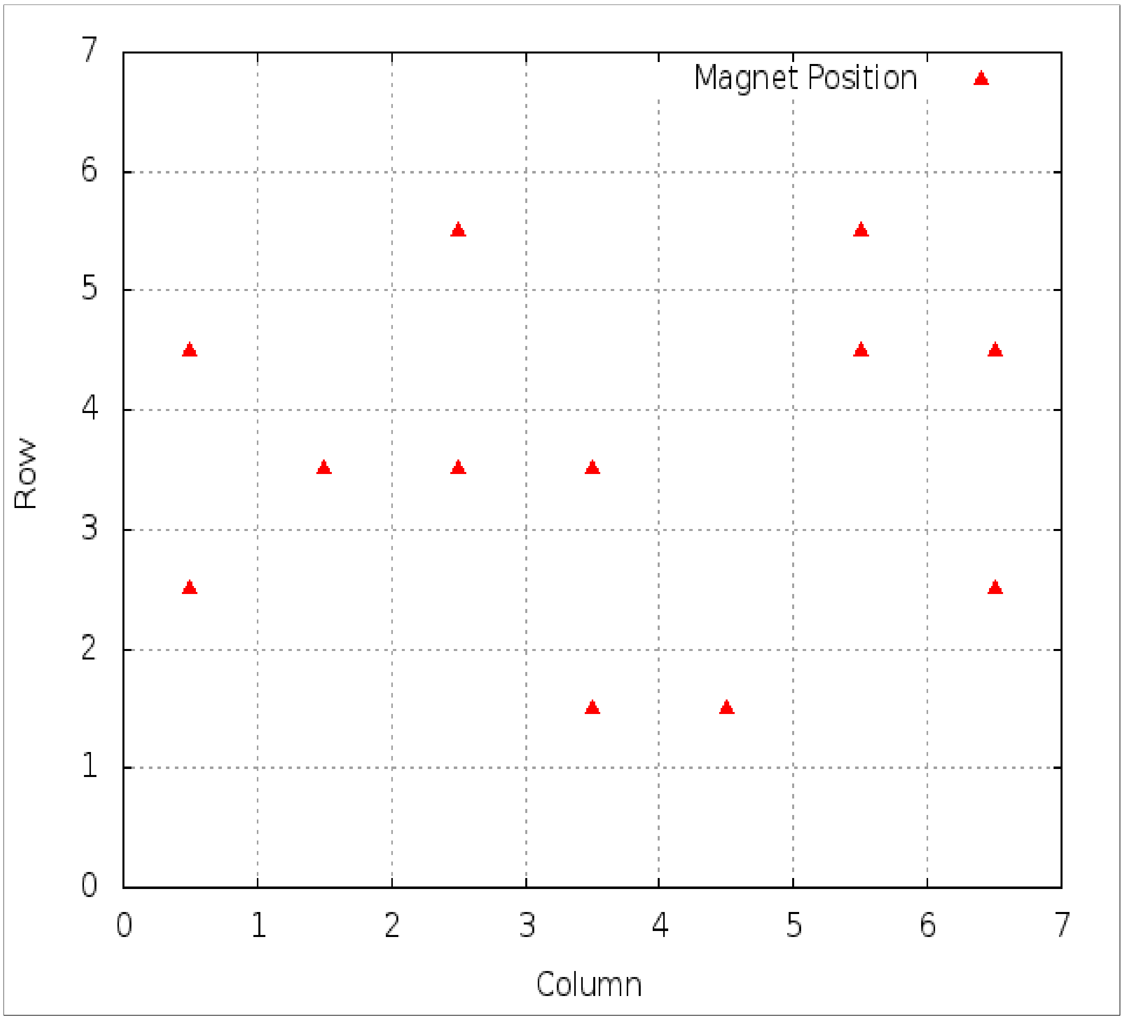

Figure 4: Test Result 1

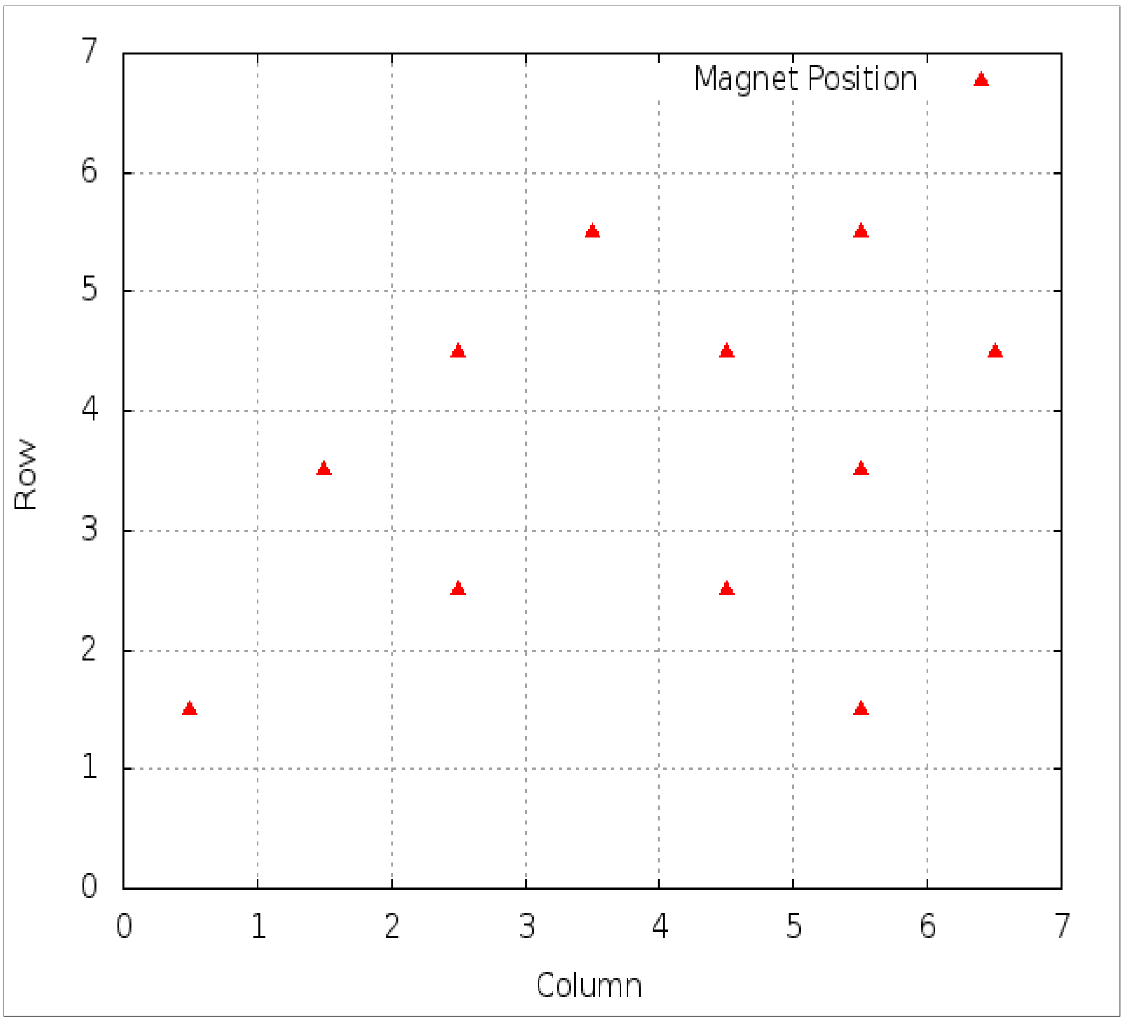

Figure 5: Test Result 2

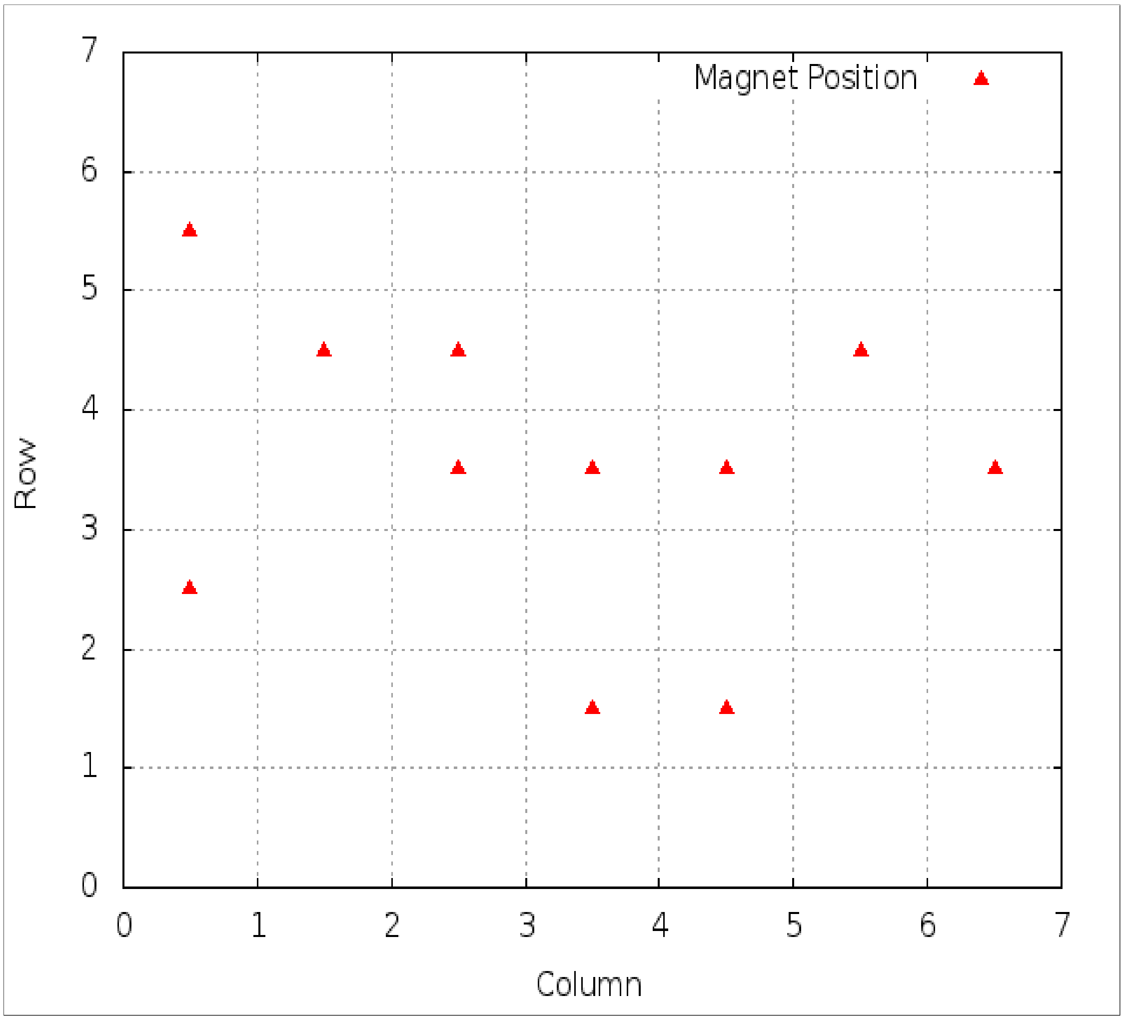

Figure 6: Test Result 3

7 Design Challenges

7.1 How to detect the rocky area?

1. **What type of sensor to be used ??**
   For detection of magnetic material or simply strong magnets, one can use magnetometer(magnetic field intensity measurement) , hall effect sensor , inductive sensors (electomagnetic induction principle),a simple switch(REED switch) with some circuit mechanism to detect the rocky area efficiently. We used a simple REED switch which is depicted in 1 as of its low cost and efficency to detect the rocky area even when the distance is 20 to 30 mm.

2. **Capability of the sensor to detect the metal/rocky area**
   Efficiency of the design depends on the ability of the circuit to detect the rocky area and its accuracy . Reed switch we used in our project can detect the rocky area which is at a distance of 15-20 mm. It also depends on the power of the magnets. More powerful magnets can increase the sensing range. But on an average, range can be limited upto 15 to 20 mm.

7.2 How to make the BOT to follow the grid?

1. **Thickness of the black line??**
   Thickness of the blackline also plays a major role as our BOT has to cover the entire arena depending on the thickness of the line and the cross junctions of the black line whiich are served as check points. One's design has to work even the thickness of the line is thin.

2. **Ability of the sensors to detect the black line**
   One should check the ability of the sensors to distinguish between the black line and white line

3. **Light conditions**(since ability of sensors depends on the Light intensity)
   Readings of the sensors varies depending on the intensity of the light hence one should check the ability of the sensors to distinguish between the black line and white line in different light conditions.The proper sensor calibration has to be done so as to make the system fleible independant of adverse lighting conditions.

7.3 How to get the data and plot the map?

1. **Data Structures to be used to store the location of the rocky area**
   To get the location of the rocky area we used zigbee communication and captured the data sent bythe BOT on terminal and used the terminal log to plot the data. One can use the file, array or different data structure to plot the data

2. **How to plot ??**
   One can use any scripting language like python or bash to collect the data from log of the terminal to store it in a file and then use GNU Plot or any other mechanism to plot the location of the rocky

8 Future Work

We have identified following possible improvements for future projects.

- **Identifying the exact location of the rocky area using Digital Image Processing**
  For more location accuracy one can use Virtual-GPS mechanism.In this, a camera should be mounted on top of bot,and it should continuously observe some stationary points. As the bot will move inside grid,these ponits will be captured by camera and corresponding values will be stored. Later on a map can be created using these values. This is an example of Simultaneous Localisation and Mapping(SLAM).

- **Plough the field using NEEDLE like mechanism keeping track of rocky position**
  The mapping and plotting can be done altogether. One should design a Needle like mechanism and attach it to our bot. The bot should plough non-rocky area and whenever the rocky area is been identifiedd bot should avoid this. It will save no.of traversals. Hencefourth, consumes lesser time for desired tasks.

- **Embed the sensing mechanism in existing projects which uses digging mechanism**
  If there is any proposed system which involves digging. Out sensor circuit can be embedded with the digging tool. In this way, the digger should

first sense field where digging to be happen. If there are some rocks in that portion, it should skip that area and dig somewhere else.

9 Conclusion

This system demonstrates that it is possible to map buried or underlying rocky area in field. Although real environment of field is rather different from the demonstration platform, this project should still function correctly as a prototype model. In real field, we cannot use grid with black lines because of the soil and the dirt on the floor. However, black lined grid for this project is mere abstraction of any mechanism that allows the Bot to move in field given some guidelines conditions. Here,grid lines serve the purpose of traversal inside the field. We can use GPS or virtual GPS system to move along straight line in real environment. This change merely needs modification at HAL layer code [4.3]. Rest of the system remains the same.

Real time Embedded systems can possess hardware inaccuracies and physical measurement faults.Still, upto certain constraint, the working system can be built. Module based development approach can lead towards complex real-time systems.

References

[1] E-yantra website http://www.e-yantra.org

[2] FireBird V Atmega2560 Robotic Research Platform Hardware manual.IIT Bombay and NEX Robotics Pvt. Ltd.

[3] FireBird V Atmega2560 Robotic Research Platform Software manual.IIT Bombay and NEX Robotics Pvt. Ltd.

[4] GNUPlot tool http://www.gnuplot.info