CS 684 Project Report

Team Number 1

FireBird VI based Feeding Robot

ASHOKKUMAR C
124050006

ROHIT GUPTA
124050002

SUBHASH KUNNATH
123050040

November 13, 2013

Table of Contents

1. Introduction

Green house is a place where plants are grown in a controlled environment. It maintains a favourable micro weather. Greenhouse provides ability to actively control temperature, humidity, resist pest attacks, precisely feed nutrients and water. In current situation there is a constant requirement of manual work for the maintenance of green house for seeding, watering, fertilizing, weeding, harvesting etc. Technological innovations are meant for making the human life easier. We can find the automation technology being widely used in manufacturing industry. But it has not been utilised in agriculture to its maximum potential. As the population of our country is growing year by year, we need to have improvement in our agricultural technologies to increase the yield and decrease the cost of production. This is the area where greenhouse and related automation comes into play. We try to solve a part of this problem by trying implementing a fertilizer feeding robot for the greenhouse environment.

2. Definitions, Acronyms and Abbreviations

- FireBird VI - a robot designed in collaboration with IIT Bombay and manufactured by NEX Robotics
- Archimedes' screw - It consist of a screw inside a hollow pipe. Rotation the screw inside will result in transfer of elements from one side of the pipe to the other side.
- L293D - a motor driver IC helps to control the DC motors connected to it

3. FireBird Bot System Architecture

Fire Bird VI uses LPC1769 ARM cortex-M3 as main microcontroller. It has an on-board Computer for complex operations like image processing, laser range finder based navigation, etc. On-board computer have a Intel Atom processor and runs ubuntu desktop 12.04 LTS.

The onboard computer is connected to FireBird VI via serial port. Fire Bird VI has 8 line sensors, 8 IR proximity sensors, 8 ultrasonic range sensors along with two gear motor with 360 ticks per revolution position encoders. Robot is also equipped with on board GPS, motor controllers, wireless communication components which supports Wifi, Xbee and Bluetooth. There are three different mode of motion control available in FB VI. Figure 1 shows the block diagram of Firebird robot VI.

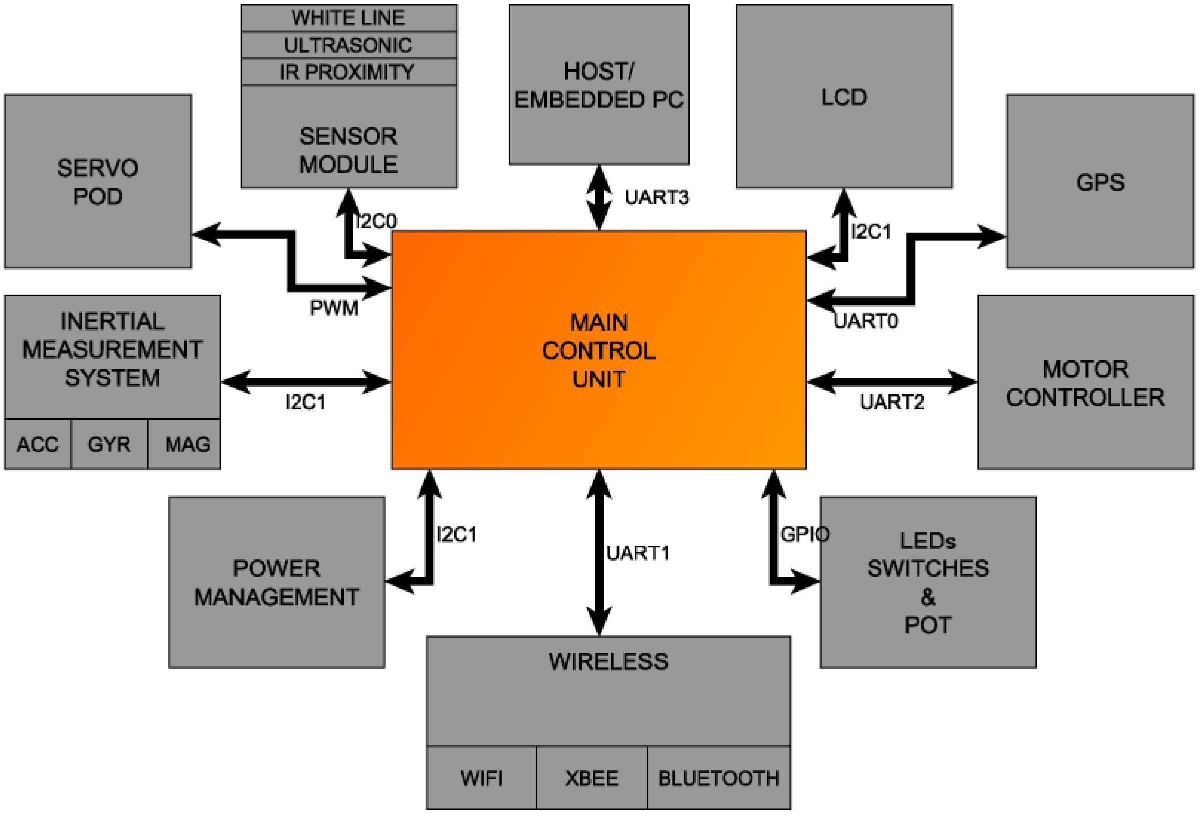


Figure 1: Block Diagram of Firebird VI

4. Problem Statement

The objective of our project is to design and build an autonomous robot which is capable of finding plants in a greenhouse environment and feeding precise amount of solid fertilizer to them.

5. Requirement specification

5.1. Hardware Requirements

- Fire Bird VI robot along with on-board computer
- Two webcams connected to on-board computer
- Two DC gear motors
- DC motor controller (L293D)
- Atmega32 microcontroller kit including the ISP programmer
- Two position encoders, one attached with DC motor
- Retractable arm mechanism
- Archimedes' screw
- USB - RS232 converters

5.2. Software Requirements

- ubuntu 12.04
- openCV
- python
- AVR Studio
- LPCXpresso v4.1.5_219

5.3. Functional Requirements

- Navigating the robot through greenhouse
- Identifying plant location (robot motion)
- Identifying root location of a plant (arm motion)
- Dispensing solid fertilizer (rotor motion)
- Stop dispensing on specified quantity (quantity control)

5.4. Non - Functional Requirements

- Power management.
- Nutrient (Fertilizer) storage level detecting
- Verification of correct operation by storing images before and after fertilizing

6. System Design

This section describes the hardware and software design of our Fertilizing robot. First section provides the overview of the system in terms of its components. The next section provides the activity diagram of the system. Following that is the state diagram . After that each of the modules are described in detail.

6.1. System Overview

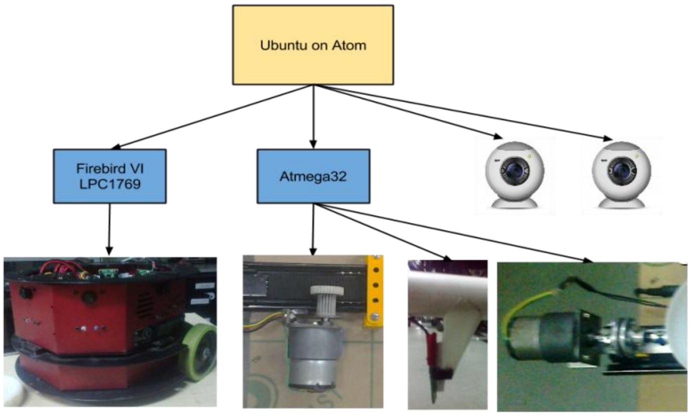


Figure 2: system overview

6.2. Activity Diagram

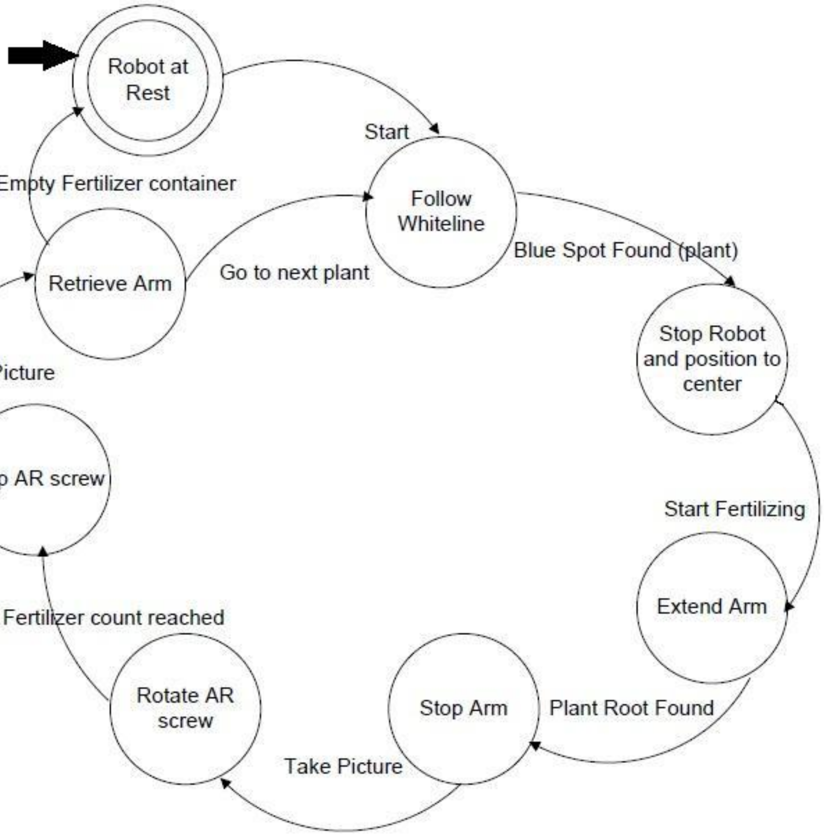


Figure 3: Activity diagram

6.3. State Diagram

This section shows the state diagram of the main system

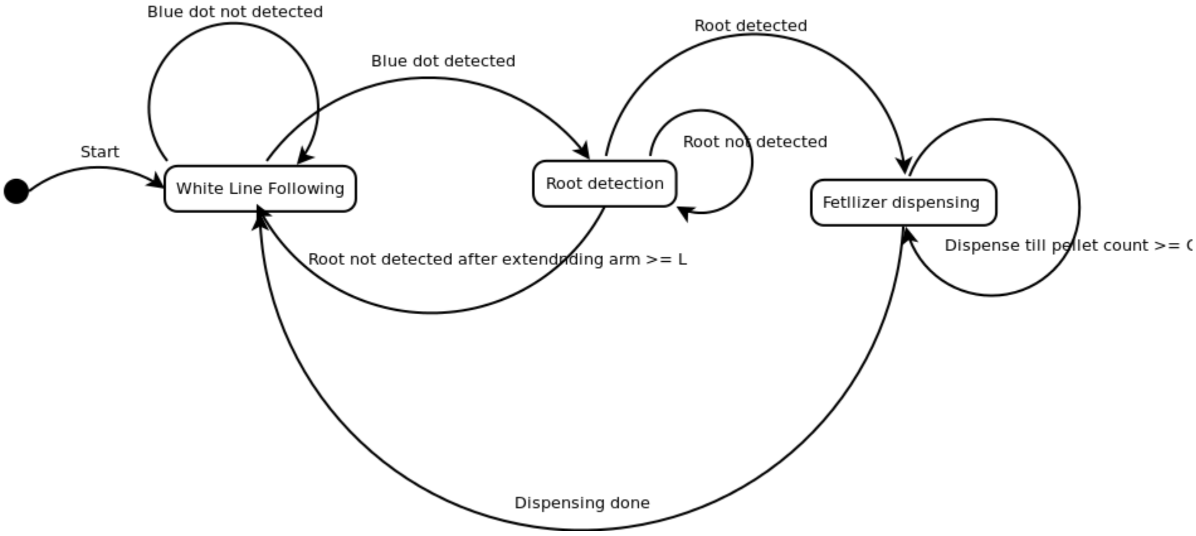


Figure 4: State diagram

6.4. White Line Following

White line following method is used for the navigation of the bot. FB VI has 8 white line sensors. The greenhouse floor is assumed to have tracks marked with white lines for the navigation of the bot. The controller script running on the onboard computer of the FB VI will read values from the white line sensors and make decision on movement direction and sends commands for motor control.

HARDWARE: White line sensor array of FB VI.

SOFTWARE: The python script running on the FB VI's onboard computer reads the white line sensor data and sends out appropriate motor control commands to achieve the white line following task.

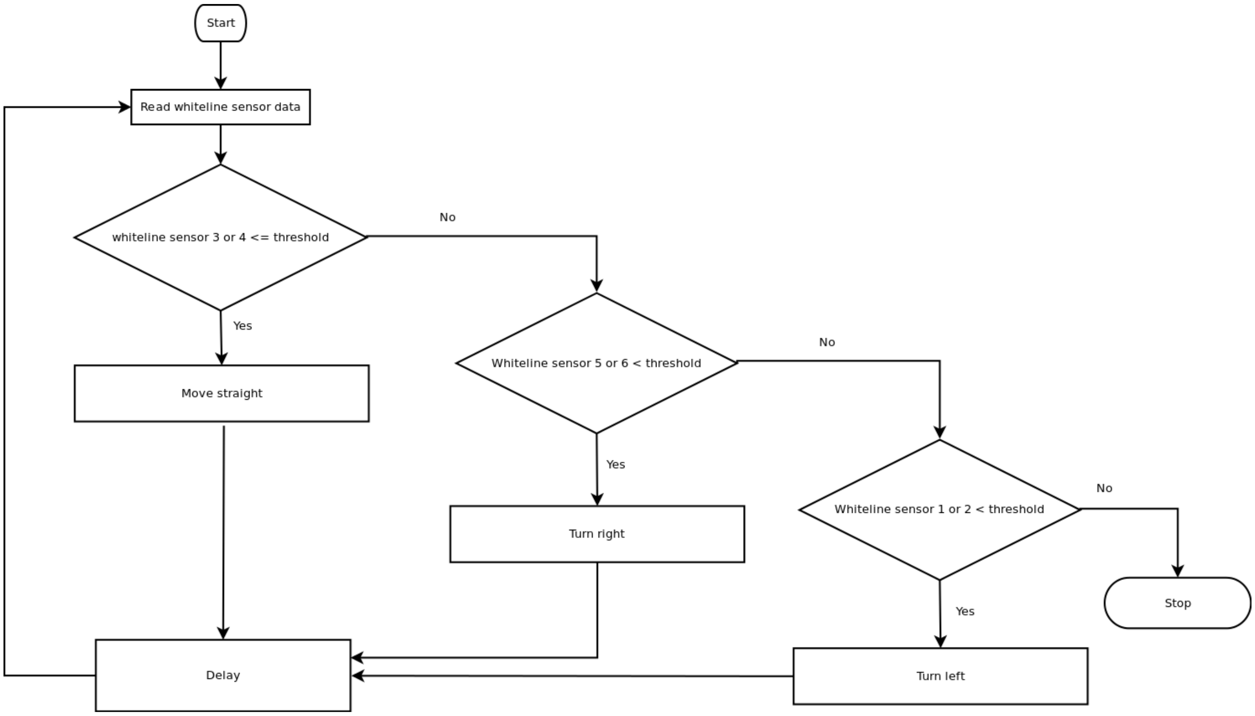

Start
Read whiteline sensor data
whiteline sensor 3 or 4 <= threshold
No
Yes
Move straight
Whiteline sensor 5 or 6 < threshold
No
Yes
Turn right
Whiteline sensor 1 or 2 < threshold
No
Yes
Stop
Turn left
Delay

Figure 5: White following - flow chart

6.5. Blue Dot Detection (For identification of plant location)

Locations of each plant are marked with a blue colored rectangle on the corresponding trough. An image from the bot’s primary camera is processed to detect the blue dot. If found the bot stops the movement and aligned itself exactly before the dot.

HARDWARE: 640x480 USB Webcam connected to one of the USB ports of the robot’s onboard computer.

SOFTWARE: The script should capture the image from the webcam and process it to find the blue dot. If found make the bot to stop and perform alignment maneuver .

DIRECTION CONTROL: For the direction control the H-bridge motor driver needs to be used.

SPEED CONTROL: Pulse Width Modulation (PWM) is used for controlling the speed of arm movement.

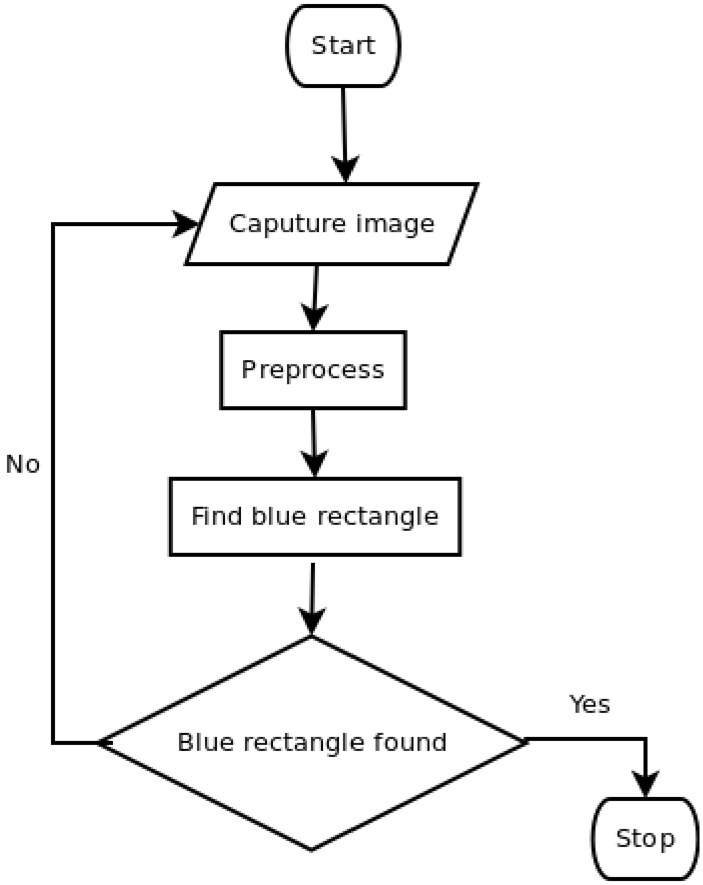


Figure 6: Blue dot detection - flow chart

6.6. Retractable Arm Mechanism

HARDWARE: Custom made retractable arm.

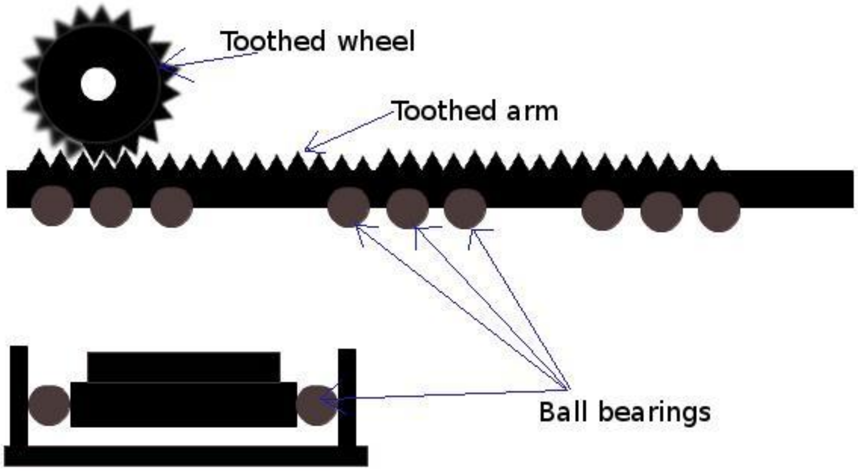


Figure 7: Retractable arm - mechanical design

CIRCUIT SCHEMATICS

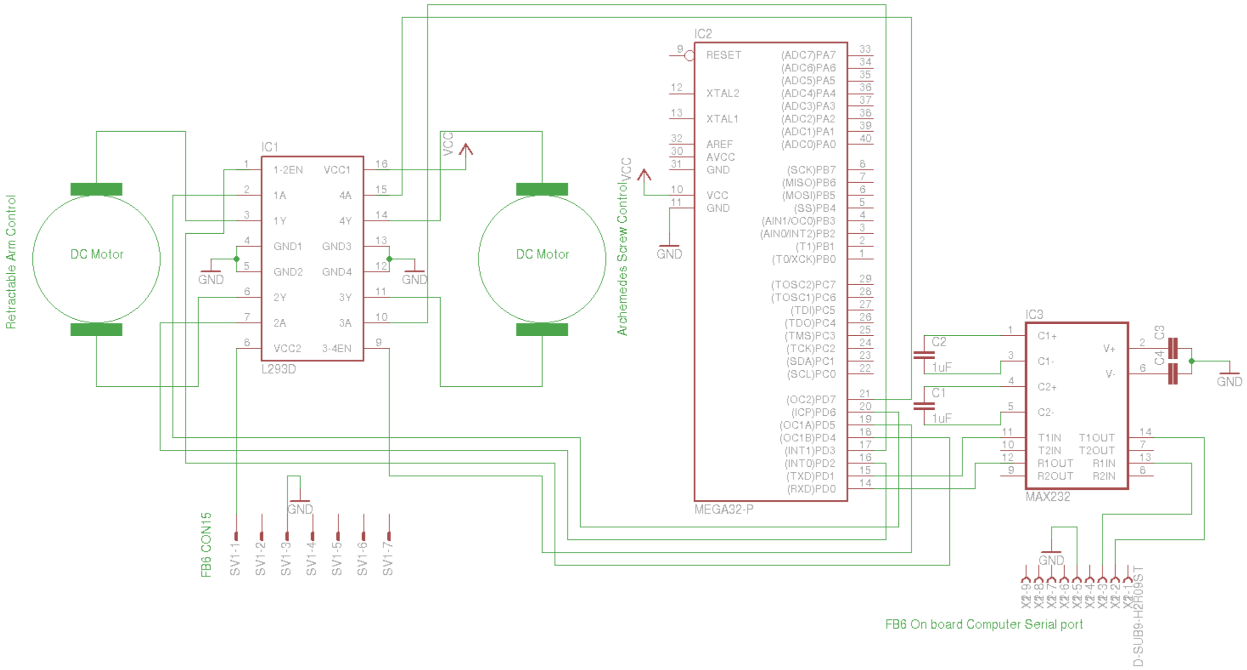


Figure 8: DC motor control circuit schematics

SOFTWARE: The avr board performs all the control operation for the retractable arm movement. The firmware implementation will listen to the UART commands, decode them and perform the corresponding operations. The script running on the onboard computer of the FB VI issues the commands to the motor control system of the retractable arm through UART.

SPEED CONTROL: PWM is used to control the speed of rotation of DC motor.

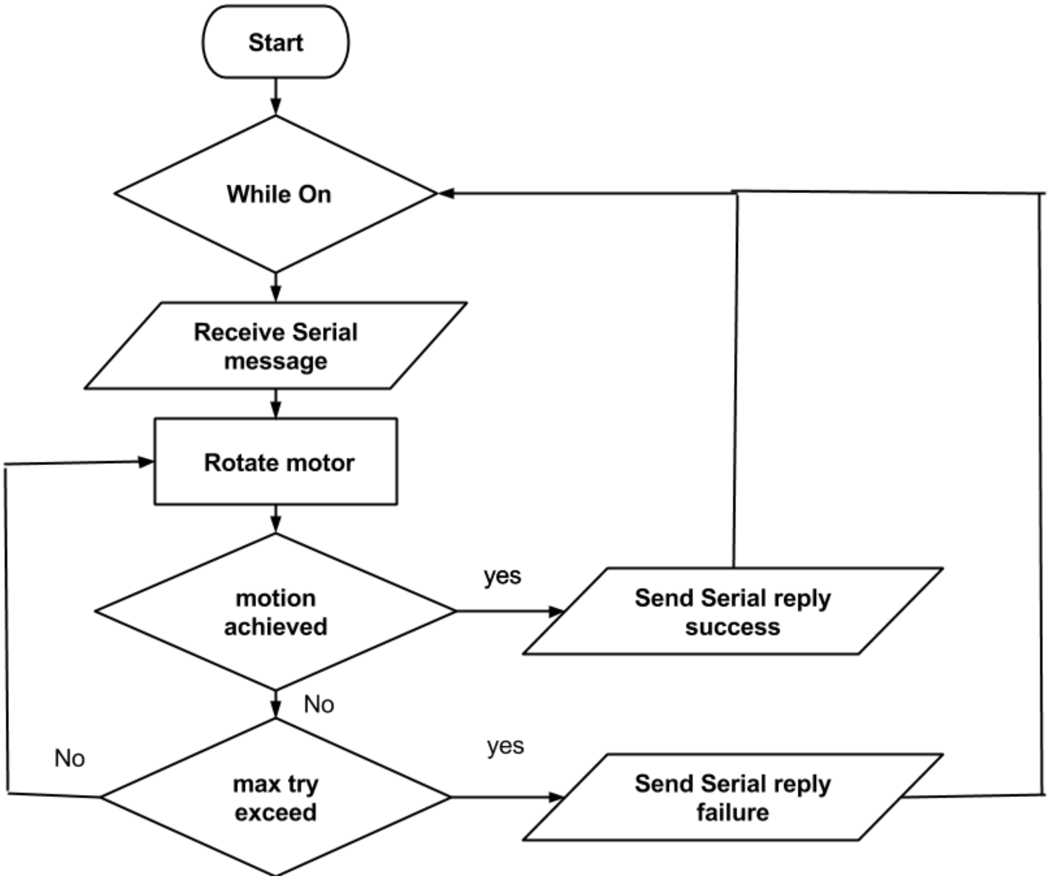


Figure 9: DC motor control - flow chart

Atmega32 Communication Protocol

The atom host connect to Atmega32 using the RS232 interface. The communication is always initiated by host device. The host should wait for first command to finish before issuing the next command. First byte of each command is always “N”,

The RS232 settings are as follows:
Baud Rate: 9600
Data Bits: 8
Parity: None
Stop Bits: None

Command: Host to Atmega32
Length: 4 bytes

Header	Motor + Direction	Speed	Distance
'N'	0x1 0x2 0x3 or 0x4	0x0 to 0xFF	0x0 to 0xFF

Motor + Direction details

0x1: move the motor 1 in forward direction
0x2: move the motor 1 in backward direction
0x3: move the motor 2 in forward direction
0x4: move the motor 2 in backward direction

Response: Atmega32 to Host
Length: 2 bytes

Header	Result
'N'	'S' = success, 'F' = failure

6.7. Root Detection

HARDWARE: 640X480 USB Webcam connected to the FB6 onboard computer.

SOFTWARE: Captures image from the camera. Then tries to match it with stored templates. If root not detected returns failure. Then the retractable arm of the robot is moved a little bit further and runs the detection algorithm again. If the root is detected, returns success signal.

6.8. Archimedes Screw

HARDWARE: Custom made Archimedes screw mechanism

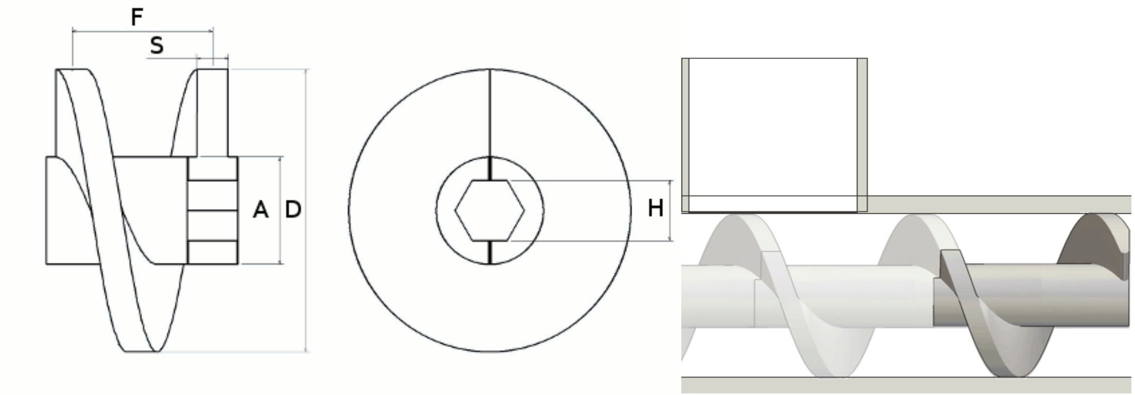


Figure 10: Archimedes screw mechanism - Mechanical design

SOFTWARE: Subroutine in AVR code which generates PWM and motor control signals. These signals are the input to the motor driver chip (L293D)

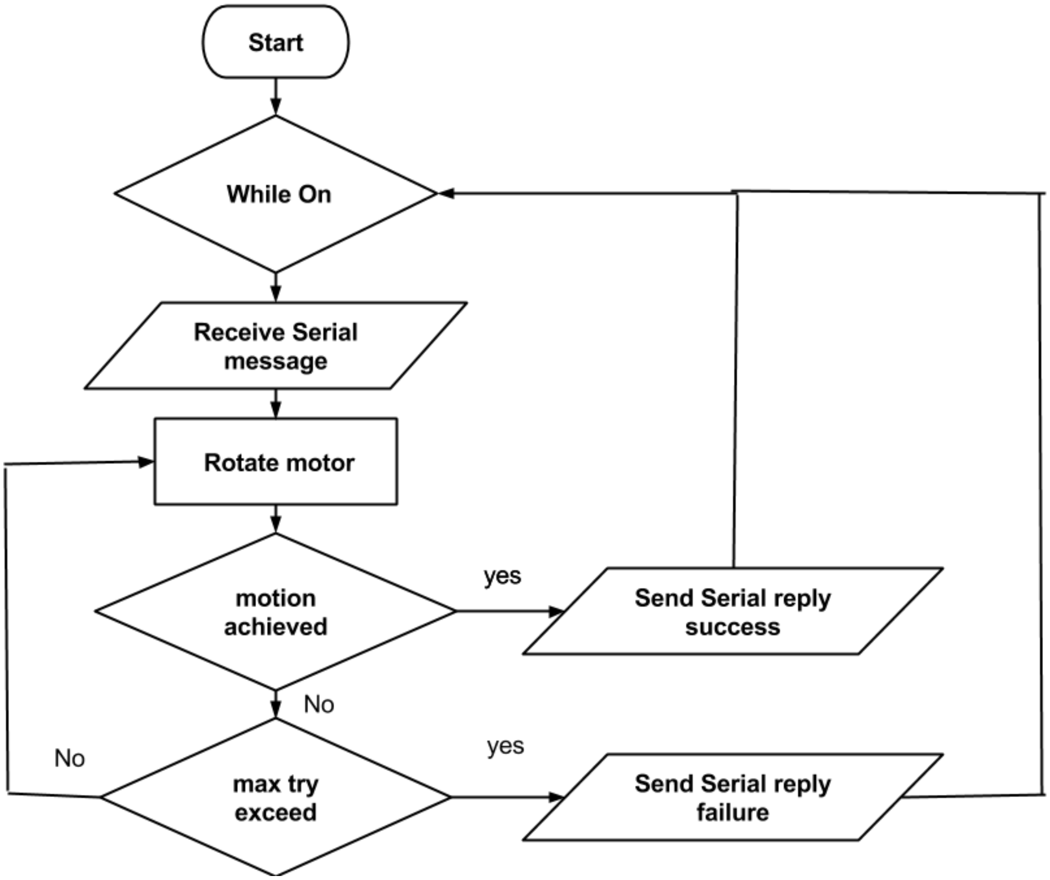


Figure 11: DC motor control - flow chart

6.9. Counting Fertilizer Grains

HARDWARE: IR transmitter receiver pair.

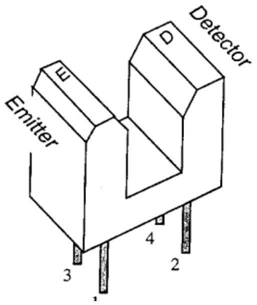


Figure 12: IR transmitter - receiver pair

CIRCUIT SCHEMATICS

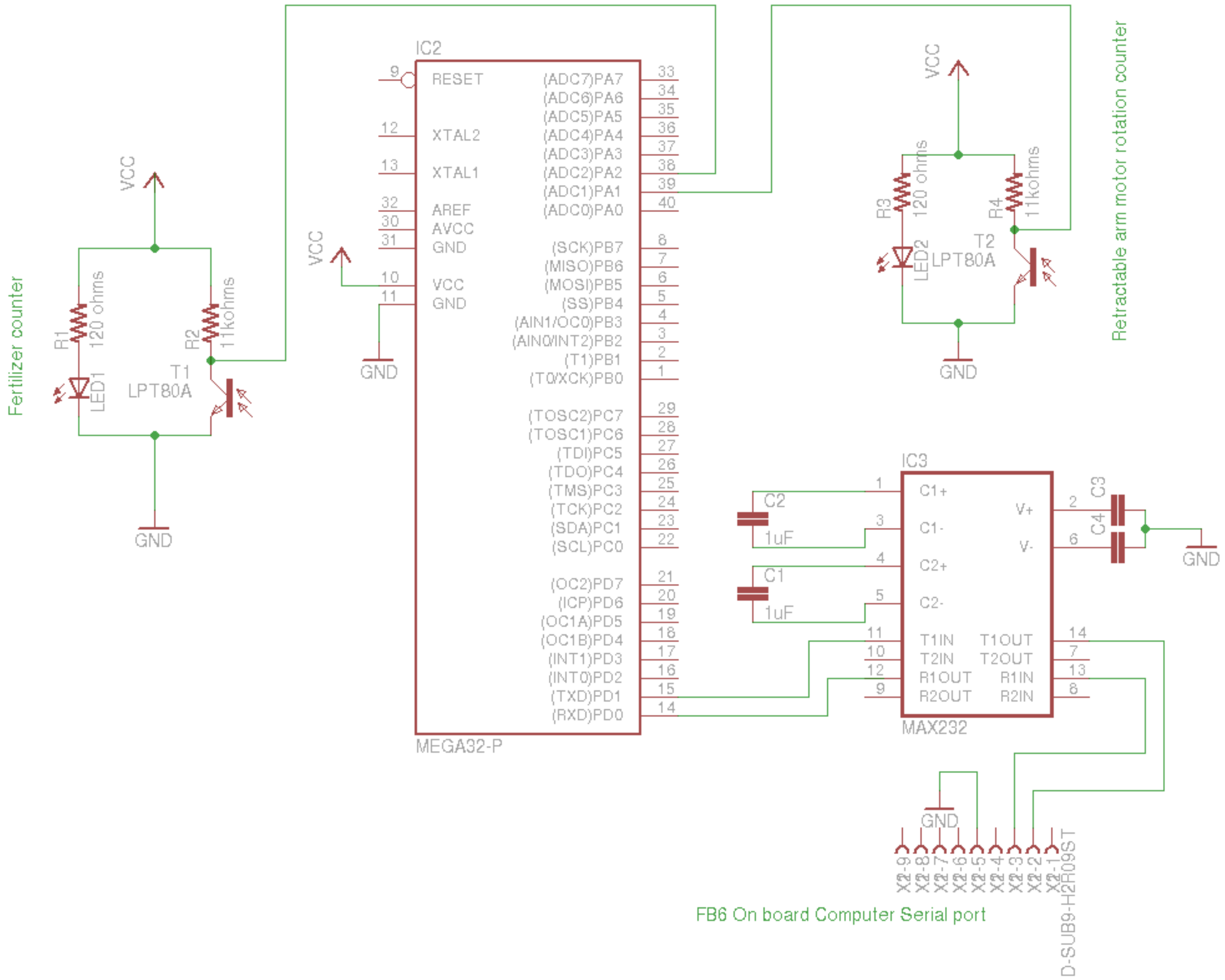


Figure 13: Fertilizer grain counter - circuit schematics

SOFTWARE: Subroutine in the AVR firmware which counts the number of IR beam cuts happened whiles the fertilizer disposal. If the number of pellets crossed the predefined threshold, motor rotation is stopped

6.10. Taking and Storing the Snapshots of Dispensing

The robot stores the images of the root position before and after fertilizer dispensing. This could be used for verifying the correct operation.

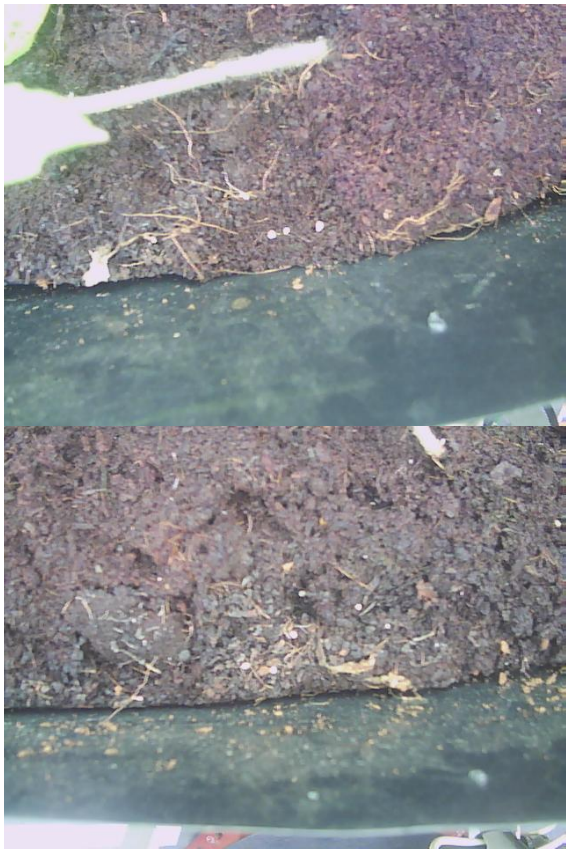
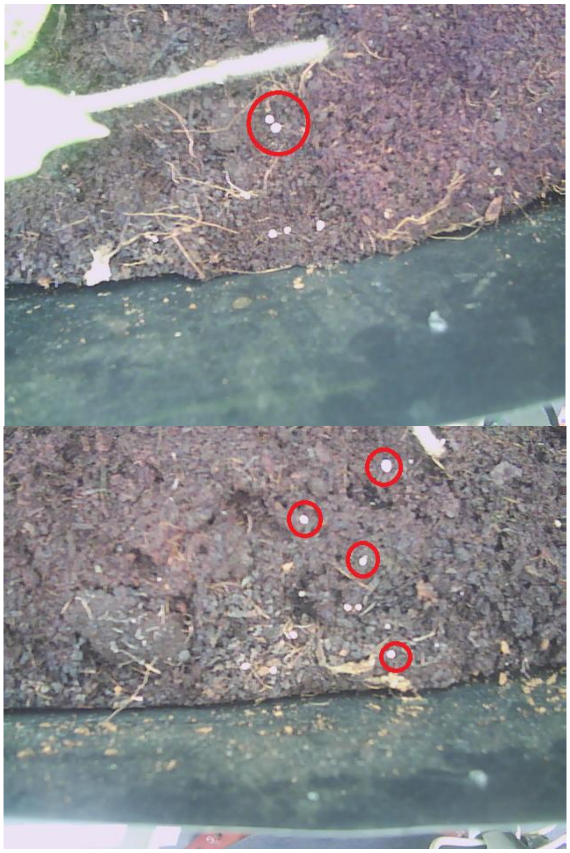
Before	After
	

Figure 14: Pictures taken by robot before and after fertilizing plant 1 and plant 2

7. Challenges

In this project we have came across many interesting and unique challenges. This section of report explains the difficulties and solution we came across

7.1. Making of Archimedes' screw

The Archimedes` screw mechanism deployed in the project is the fourth major model we made.

Model 1:

A plastic pipe is used as an axis of the screw, disk shaped plastics are curved out from a paint bucket using a divider and a hole of size equal to the axis outer diameter is made on all disks. A cut is made on each disk and inserted on the axis. Disks are connection with each other and glued on the axis as shown in the below figure

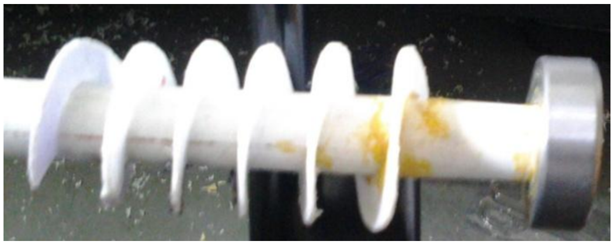


Figure 15 : Model 1

Then a ball bearing is connection on each end of the axis and inserted in a larger outer plastic pipe.

Problems:

- The inner circle of disks is not regular and created irregular fit.
- It is difficult to rotate inside the outer pipe, which is resolved after sanding.
- The space between the disks are not uniform
- There are several bends on the disk which may not result in smooth flow of grains
- It is too big to fit on a Fire Bird VI robot

Model 2:

A metal pipe of small diameter is used as axis and a plastic pipe is used as an outer covering layer. Small disks are curved out from the plastic sheets used for spiral binding. The diameter of the disks is equal to the inner diameter of the outer plastic layer which will cover the screw mechanism.

Small hole of size equal to the diameter of axis is made on every disk. A cut is made on every disk to break the ring shape. These disks are inserted on the axis and connections are made between edges of the disk to form spiral shape, then it is glued with the axis and inserted in the outer layer (pipe). Little sanding is required on disks to make the screw mechanism rotate inside the outer layer pipe

Problems:

- The disks are flexible, may not be strong enough to move the grains forward.
- Curving inner (small) circle was difficult.

- Axis is not steady during rotation because of flexible disks

* To increase the strength of disks, the above same process was repeated with disks made from lamination sheets (after lamination without any paper) and folder covers. Both are not satisfactory and faced similar problems.

Model 3

A plastic pipe is used as axis of the screw. Flexible transparent plastic hoses are used instead of continues disk to form spiral shape. Hose are made to spiral around the axis. Hoses of different diameter are tried.

Problems:

- Hose are not flexible enough to spiral around the axis with close intervals
- The inner space of hose is not regular after spiralling around the axis, sometimes it blocks entire passage

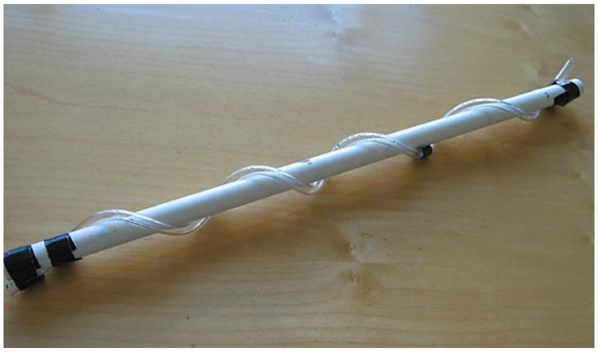


Figure 16: Model 3 - concept

Model 4

This is the final model which is used on the robot for dispensing. A metal pipe of smaller diameter is used as axis of the screw and a plastic pipe is used as outer covering layer. To make the spiral, multiple strips of smaller width is cut out from folder cover like shown in the figure (17) and made flexible by using a pencil. Initially we glued few strips together by holding it in spiral shape and with required thick, but it is difficult to do make such spirals separately and connecting it with each other cause extreme difficult and more thickness at the joint. So decision is made to make the spiral shape layer by layer on the axis.

The layer by layer approach gives advantages such as

- the thickness can be increased or decreased
- there is no sudden increase in thickness at joins of two strips
- easy to glue the strips separately
- since the strips are of same width, it gives expected finish and strength

To ensure the distance between the spirals, a blue insulation tape of suitable breath is initially spiralled on the axis with small and uniform space between each spiral. This space between each spiral is used as a guiding line and layer of stips are made on top ot it. In our setup it required eight layer to fix exactly inside the outer layer. To ensure smooth rotation inside the outer layer some portion of spiral required some sanding. Figure(17) show the final version of inner axis.

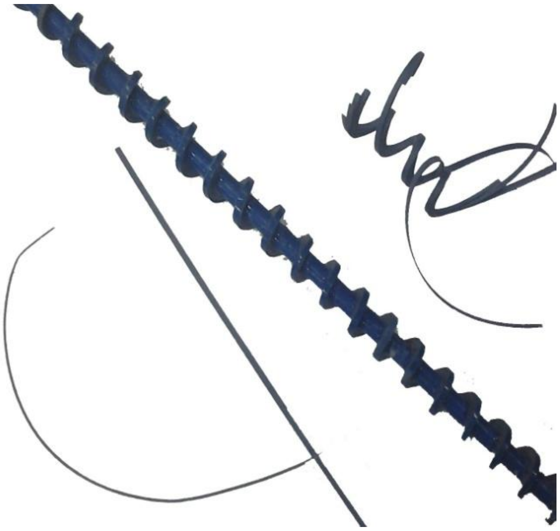


Figure 17: Archimedes` screw Axis finished version, plastic strip and glued spiral

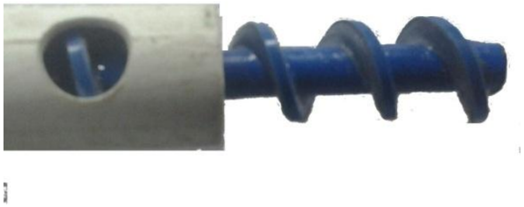


Figure 18: one end of Archimedes` screw model 4 with a hole for dispensing.

7.2. Making Retractable Arm

Retractable arm of the dispensing mechanism helps to drop fertilizer grains near to the plant root. The first attempt of implementation was to make the entire arm from aluminium sheets and bars. But at the initial point itself it has been found that creating a robust mechanism would be very difficult and time consuming. After searching for a similar mechanism in household objects (cd player tray, cupboard drawers etc.) we figured out that the cupboard drawer railings would be good choice for our implementation.

A toothed rubber sheet has been sticked on the railing; a plastic toothed gear has been attached to the DC motor and fixed all components as shown in the figure (20). An IR transmitter-receiver pair has also been attached to the motor axis for counting the rotation angle. With this count feedback the amount of rotation can be controlled by stopping the motor when required cous has been reached.

Figure 19: Cupboard drawer railings

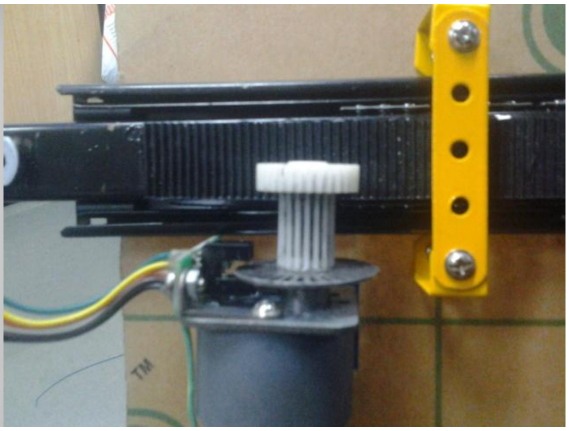


Figure 20: Retractable arm - DC motor

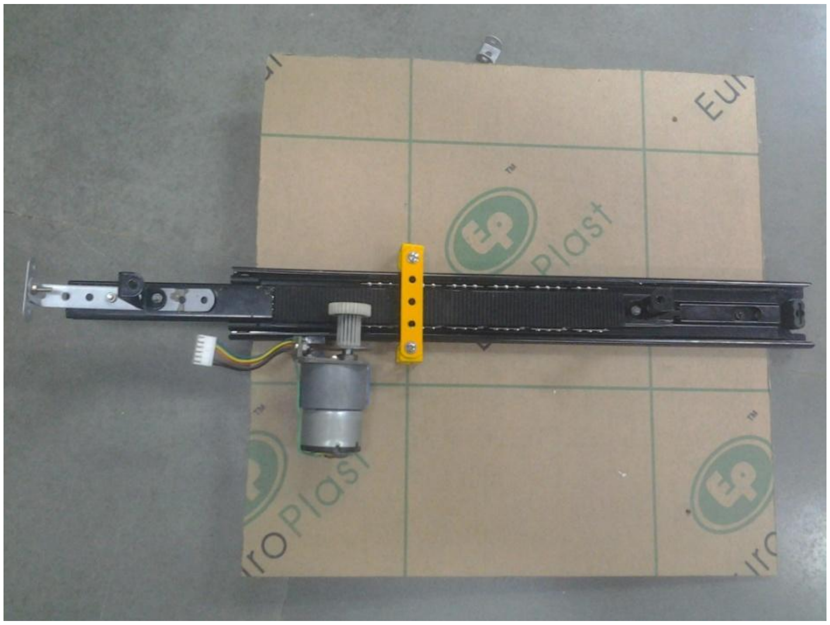


Figure 21: Retractable arm mechanism mounted on an acrylic sheet

7.3. Robot movement guiding

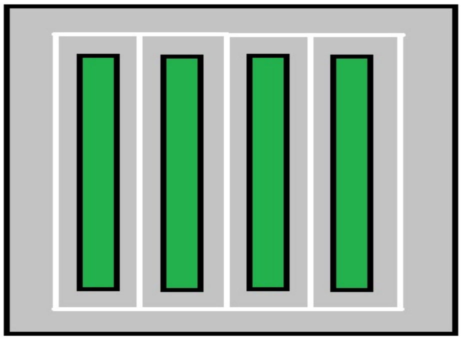


Figure 21: Greenhouse Layout

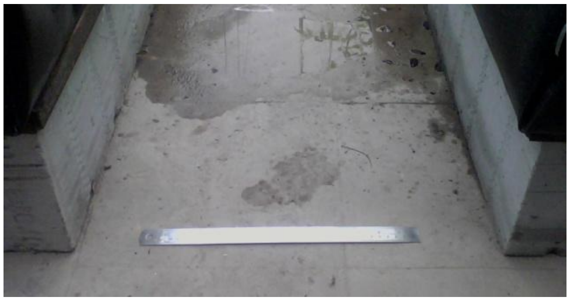


Figure 22: space between troughs

Initially ultrasonic sensor and IR proximity sensor are considered for guiding the robot inside greenhouse. Figure (21) show the layout of the greenhouse where plants are in green colored area (trough).

Ultrasonic:

There are eight ultrasonic sensors available in firebird VI and it has blind spot at less than 6cm. The space available for Firebird VI for moving between troughs is very less (). so the ultrasonic cannot be used for align the robot at the middle of two troughs.

IR proximity:

The next option available is the IR proximity sensors. These sensors give different reading for different obstacles and its not stable. So IR proximity sensors cannot be used. Following table shows the reading of IR sensor #0 on two different walls.

Distance (cm)	IR0 on wall	IR0 on wall 2 (rocky)
0	13	124
1	13	127
2	14	142
3	16	158
4	37-39	170
5	72/73	179
6	10/102	188
7	120/121	195
8	138/138	201
9	151/152	206
10	162/163	210
11	171/173	214

White line:

The white line sensors available on FireBird VI are used for this project, There are eight sensor available out of which only 6 sensors are made use to follow the white line. Availability of white line in greenhouse as showed in the layout about is assumed to be present.

7.4. Programming LPC

LPC1769 ARM cortex-M3 is the main processor of FB VI. The steep learning curve we had to follow to learn this new environment within the time limits of the project duration was one of main challenges we have faced. The firmware running the processor communicates to the sensors and actuators of the bot. It listens to the commands received through UART, decodes the command and executes appropriate operations - like reading and sending out sensor values, activating motors etc.

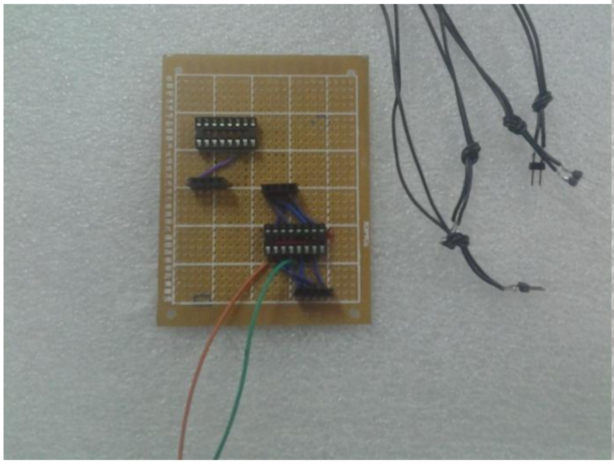
On the hardware point of view LPC processor is a very delicate chip to deal with and GPIO ports are exposed on the motherboard of the robot without any voltage protection. One chip got damaged while working with the bot. Which added additional time and labour to the project to get it replaced. Ultimately we removed the DC motor control logics from the bot's LPC firmware and moved to an ATMega32 chip.

7.5. Controlling the DC motors and pwm

As mentioned above initially all the logic for the control of DC motors resided in the LPC firmware. After the complete implementation the whole logic had to get ported to AVR chip. These two chips are entirely different in their architecture; instruction set and registers leading us to spend extra time on rewriting the code in LPC instructions. The ATMega32 chip is interfaced with the L293D motor driver chip. 2 PWM outputs from the ATMega32 are connected to each of the enable pins of the L293D H-Bridges, which control the speed of motor rotation. 2 GPIO pins each from the microcontroller are used for the direction control of each motor. At first a custom made DC H-Bridge driver circuit made on a common circuit board has been used and later replaced with a commercially available PCB module for its robust design and well made wire connectors.

The mechanical components also played big role in aggravating the difficulty. The free and easy movement of the retractable arm and Archimedes screw mechanism were achieved only after multiple refinements in the design and structure.

Another thing was battery power related issues. The battery had to be in a relatively good charged state for the DC motors to rotate against the friction and inertia. When the battery was in low charge state the motors were not rotated at all. This issue has not been taken care of. Our solution is just to keep battery in good charged condition.



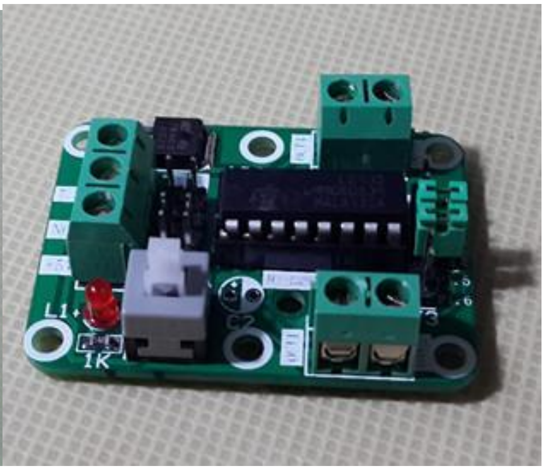


Figure 23: Motor driver circuit

Figure 24: Commercially available motor driver

7.6. Counting the fertilizer grains

As per the initial design, control over the number of grains of fertilizer is supposed to be achieved by the controlled rotation of the Archimedes screw mechanism itself. But we found out that rotation can bring clumps of grains together and make a direct relationship between the amount of grains dispensed and the degrees of rotation impossible. To tackle this, a grain counting mechanism has been added at the nozzle of the dispensing mechanism. The counter included a IR transmitter - receiver pair. The falling grains will cut the IR beam which could be detected by measuring the voltage across the receiver side. With this implementation we are able fine control the dispensing amount.

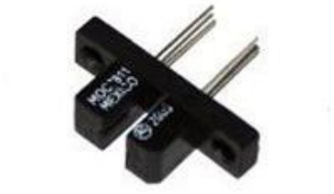


Figure 25: IR transmitter- receiver pair

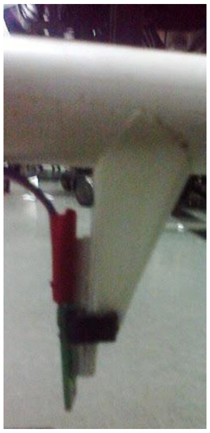


Figure 26: IR transmitter receiver pair mounted on the nozzle of dispenser

7.7. Image processing challenges

First we tried to detect the plant by identification of the stem in the image. this required classifier to identify the object in the image. OpenCV supports Haar Feature-based Cascade Classifier for Object Detection but the only classifier readily available is for face detection. Generation of classifier require hundreds of positive and negative sample images. this classification process take days therefore it is not practical for part of a course project.

The design was changed to identify a QR marker placed at the plant location. This QR code should be distinct for each plant to uniquely identify it. QR identification was a multi step process. The first step is to identify the location of the QR marker. position the camera in correct orientation and taking the picture and computation if the QR code. the biggest challenge was to position the camera in correct orientation as the direction of the robot could vary slightly.

With different lighting and background the color of the red patch was changing from image to image. The identification of the red color marker is done using the image, converting it to hue saturation and intensity format to identify the color range. As the red color is in the 0 as well as 360 range of hue, the spot detection failed frequently.

Markers were placed on top of the trough wall, this caused some incorrect identification of marker because of the background objects, we moved the marker to the center of the trough wall.

Figure 27: QR code with encoded data “tomato plant 1”

7.8. Plant stem detection

While detecting the root of the plant we again faced the problem of orientation lighting and learning so we have to search for an algorithm that should overcome these problems. Also different orientation of the stem may result in false identification. We tested 6 different template matching algorithms available in Opencv. In our tests they all showed better results compared to others in different conditions. We finally decided to use all and used statistical methods to obtain final result.

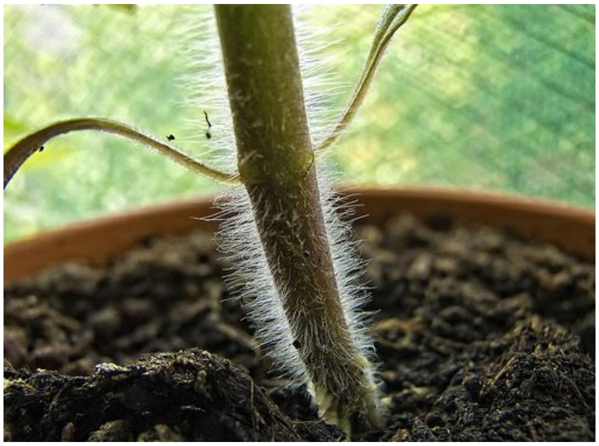


Figure 28: Tomato plant stem

8. Future Scope

8.1. Navigation

Current navigation system makes use of white line following method. It could be be improved with adding methods for obstacle detection and avoidance. For that image processing techniques or FB VI's laser range finder sensor could be used. Another area of improvement is using of localization implementation. With this the bot's controller program can have an idea of current location of the robot in the map. Fault detection and correction in the navigation can be achieved with this. It could also add functionalities like, sending orders to the robot to service a particular plant or plants.

8.2. QR code identification

In the current implementation, plant locations are identified by detecting blue squares. This can be replaced with QR codes. QR codes can store lots information. This storage ability can be used for uniquely identifying the plant. Any other details like plant type, amount of fertilizer needed, time at which the plant has been planted etc can be stored in the code or in another database which has been mapped to a unique ID contained in the QR code. For the QR code detection the camera needs to be exactly aligned with QR code. For this it would require servo controls.

8.3. Better learning algorithms for root detection

In our implementation root detection is done by template matching algorithms, which very error prone. One possible future extension to this module could implement other pattern matching machine learning algorithms. It would need a lot of training data and time to implement.

8.4. Additional functionality like weeding etc

Current design is meant for single use i.e. fertilizer feeding. It would be inefficient to use the robot for a single task. So it is preferable to have multiple functionalities implemented in the robot and have ability to switch between the tasks. A lot of saving could be made in terms of energy and time consumption, if multiple tasks could be performed on a single walk of the robot through greenhouse.

8.5. Liquid dispensing

For making liquid dispensing possible, the Archimedes screw mechanism needs some refinement for avoiding leaks. Also the counter at the nozzle needs to be replaced with a flow measuring mechanism.

8.6. Power management

Power management is the ability of the robot to estimate power usage, plan activities according to that and have failsafe mechanisms like autonomously changing itself, raising an alarm to get help etc. In the current design robot’s built in LED indicators and alarms work independent of our design, which takes care of the low battery conditions by alerting the user. This could be extended by having an implementation which gives out an estimate of how many tasks the robot can do with the current battery level.

9. Final system

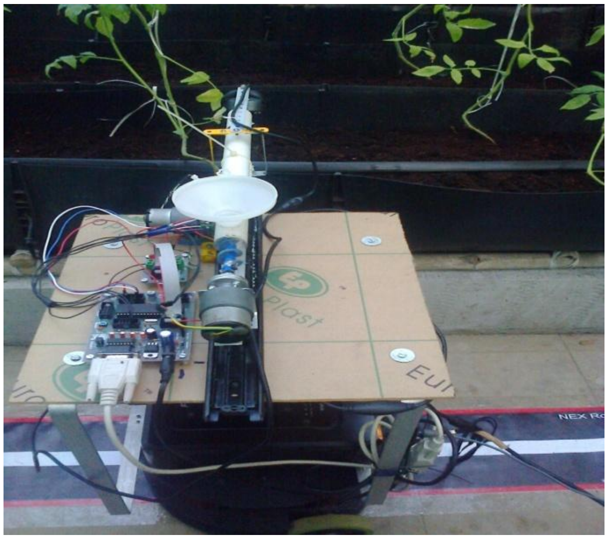


Figure 29: Side view of Feeder robot

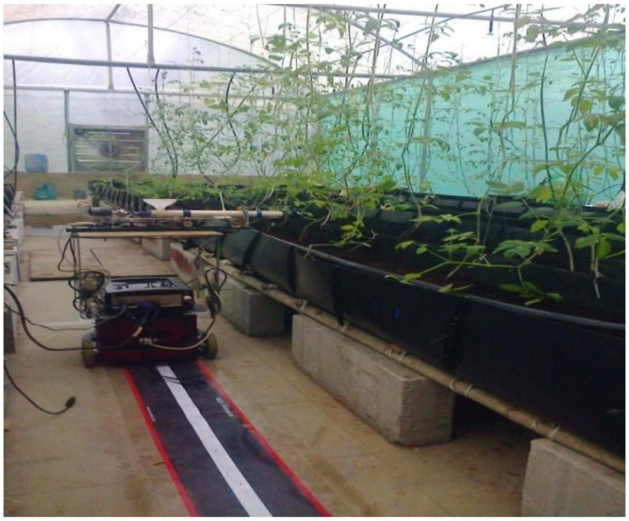


Figure 30: Feeder robot during work

10. Testing Strategy and Data

10.1. Unit testing

Each of the modules are individually tested before integrating.

10.2. Integration testing

Integration test is done after integrating the modules in lab environment and also in greenhouse environment.

11. Constraints

Following are the constraints of the design.

- It can only work till the two batteries have a good amount of charge
- It needs well lit environment
- It require white line navigation tracks
- A blue colored rectangular sticker need to be sticked exactly at the stem position on the trough
- Can only deal with solid fertilizer pellets
- It can fertilize upto the extent of length of its arm.

12. Assumptions

Following are the assumptions made for the design.

- Optimum lighting condition in the greenhouse for the image capture to work
- White line tracks between the troughs for guiding the bot
- Blue colored rectangular sticker on the troughs where the plant is located
- Solid fertilizer pellets of the size of 1-2mm diameter.

13. Conclusions

The robot for feeding fertilizer to the plants in a greenhouse environment has been designed and implemented. A lot of things have been learned about real time systems from this project. Experienced the difficulty involved in implementing real time system with electro mechanical parts. The murphy law ("If anything can go wrong, it will") has been found true in multiple cases. Many things had to be redesigned, rebuilt and fine tuned for the system to work properly. Gained experience in various domains like, mechanical design, electronic circuit design, system design, various microcontroller/microprocessor environments (LPC, AVR, Intel Atom) and various programming languages.

14. References

[1] Fire Bird VI Hardware Manual
[2] Fire Bird VI Software Manual
[3] LPC1769 datasheet
http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf
[4] Atmega32 Board
http://www.onlinetps.com/shop/index.php?main_page=product_info&cPath=75_119&products_id=638
[5] Atmega32 Datasheet
www.atmel.com/Images/doc2503.pdf
[6] Motor Driver

http://www.onlinetps.com/shop/index.php?main_page=product_info&cPath=75_119&products_id=638