**CS-308-2014 Final Report**



**Cyborg** TU-**7**

Kanishak Kataria 100050026

Rohit Yaduvanshi 100050027

Sonal Chouhan 100050036

Pratik Kumar 100100018

**Table of Contents**

[1. Introduction](#h.bq0c22nz6fgb)

[2. Problem Statement](#h.pjudw5qfxl7a)

[3. Requirements](#h.dniwgn6l11gi)

[3.1 Functional Requirements](#h.z3vcgqms2ree)

[3.2 Non-Functional Requirements](#h.myl1pnl39a2l)

[3.3 Harwdare Requirementss](#h.nv28a9k4ors8)

[3.4 Software Requirements](#h.q95wqbndkk8z)

[4. System Design](#h.8oxftuju71ct)

[5. Working of the System and Test results](#h.xtfc77m7zb81)

[6. Discussion of System](#h.pz5lwmv2gq41)

[7. Future Work](#h.vb4fqfet69jx)

[8. Conclusions](#h.kpvfwc80v36d)

[9. References](#h.tkz21pap18al)

## **1. Introduction**

AgroBot, abbreviated as AGB, is employed for an automated watering of plants in green houses, small and terrace farms, which need minimal surveillance. AgroBot also helps farmers to timely and accurately water farms and also monitor moisture hold of soil to better manage and schedule irrigation in farms.

**Motivation**

* Irrigation is very important task in farming, which is the main occupation in India, and requires a lot of labor and time. AgroBot simplifies the task of irrigation to just refilling overhead tank.
* Spending water efficiently really matters when a lot of areas are facing water scarcity. AgroBot is expected to outperform conventional irrigation when it comes to efficient water spending. Since plants differ in their water requirements, one does not have to water every kind of plant at same frequency, AgroBot performs selective irrigation to points as per requests made by plants.

## **2. Problem Statement**

The overall problem statement can be divided into broadly these 3 parts -

● **Sensing and tweeting status of the water content in troughs** -

* Need to sense water level in troughs and tweet this to the server periodically, thereby monitoring the water content of plants in greenhouse remotely.
* Need to design as many water level sensors as troughs and a communication protocol in place to deliver messages.

● **Transfer of instructions from server to AgroBot** -

* Sending information from the central server about the trough that need water and its level of water.
* Reach the trough using line follower mechanism and water it according to the current water level

● **Autonomous watering of plant** -

* Water the trough based on the water level it already has, using a water pump, running time of which is to be controlled by AgroBot, thereby controlling the total flow of water.

## **3. Requirements**

### **3.1 Functional Requirements**

* **Exchanging information about the water level in the trough with the server -** The troughs tweet the status of water level to the center server in continuous intervals.
* **Transfer of message from server to AgroBot about the troughs that need water** - Using the data obtained by the center server from the troughs, it asks AgroBot to go and water those troughs which display the troughs it needs to water on its LCD display.
* **Motion of AgroBot to the correct trough -** After getting the troughs which need watering, AgroBot uses its two-dimensional mapping of the GH and current location to move to the trough and water it.
* **Proper amount of watering based on the water level in the trough -** AgroBot has a self-controlled battery-operated motor pump which helps in watering the trough at a constant rate. The watering is done for a specific time duration depending on the waterflow rate and amount of water needed in the trough.
* **Resetting the position on low water content level** - AgroBot moves to a default place in the GH whenever the water content in the tank goes low. After refill AgroBot resumes the task of refilling.

### **3.2 Non-Functional Requirements**

* **Price of water level sensors** - We try to make circuits as cheap as possible, since they are to be installed at multiple points in a farm, making this model economically feasible and worthy of installation.
* **Safety of AgroBot** - Since AgroBot contains an overhead water tank, it is required that bot circuitry is insulated and safe from water intrusion.
* **Battery backup to AgroBot and trough sensor** - The watering bot and sensors should have continuous power supply as the continuously listen and transmit requests to the server

### **3.3 Harwdare Requirements**s

* + 1. **Water Level Circuit**
* General Purpose PCB
* NPN Transistors
* Resistors
* LEDs
* Wires and Connectors
* Solder Wire
* Berg Pins

**Circuit to Tweet the Water Level at Trough End**

* Water Level Circuits made above
* FireBird V
* 1 ZigBee Module

**Central Server**

* 1 Laptop
* 2 ZigBee Module

**AgroBot**

* FireBird V
* 1 ZigBee Module
* Water pump on robot
* L293d Motor Controller Circuit
* Plastic Water Tank to hold water on AgroBot
* Pipe
* Plastic for insulation

**Greenhouse**

* White line path with black markings
* Small troughs/plastic bottle to water
* Thermocol
* Black tape

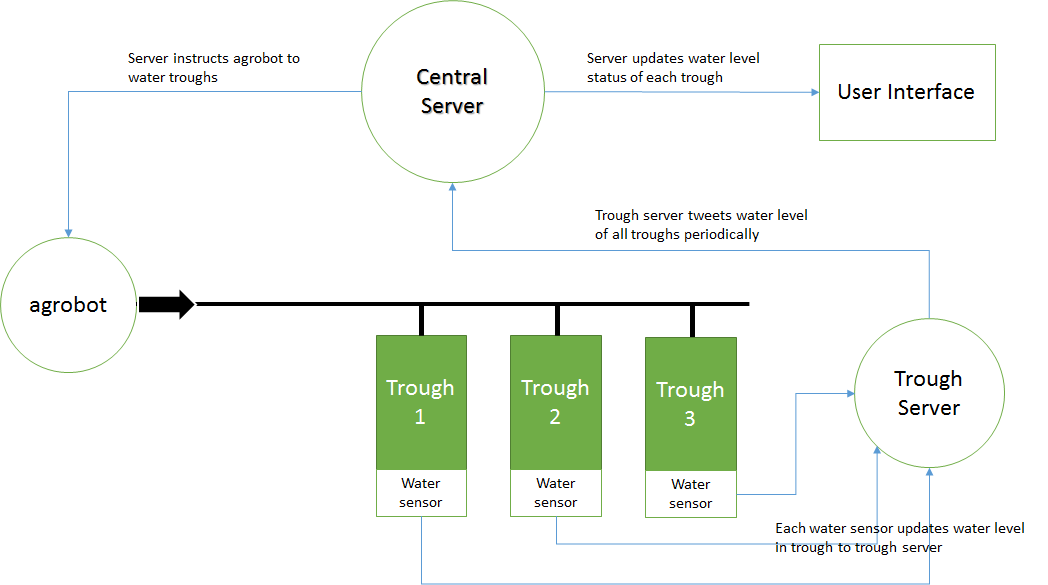
### **3.4 Software Requirements**

* Keil
* X-CTU
* Netbeans
* jdk 1.7.0 and jre7 or higher

## **4. System Design**

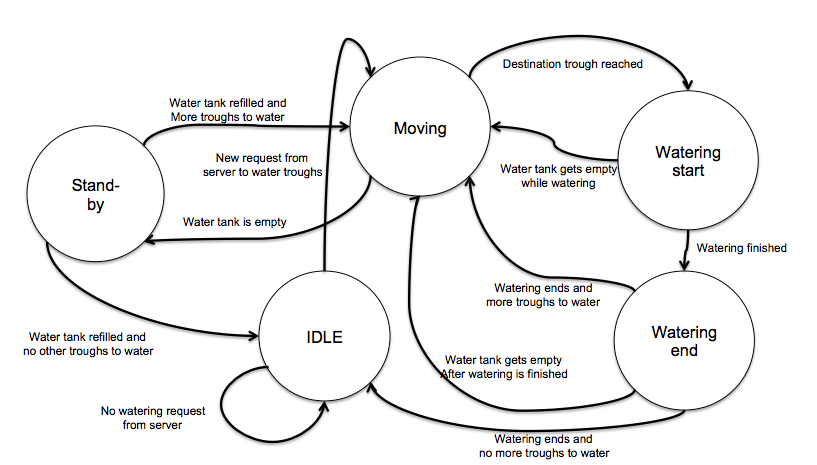
**Architecture of the System**

* The following figure explains the architecture of the entire system.
* It explains the relationship among various components of the system



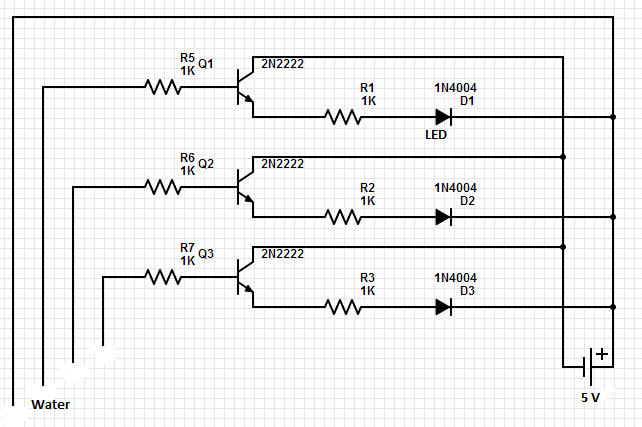
**Finite State Machine**

* This figure represents the Finite State Machine(FSM) of the overall system
* It explains the flow of control among various components and state of the system



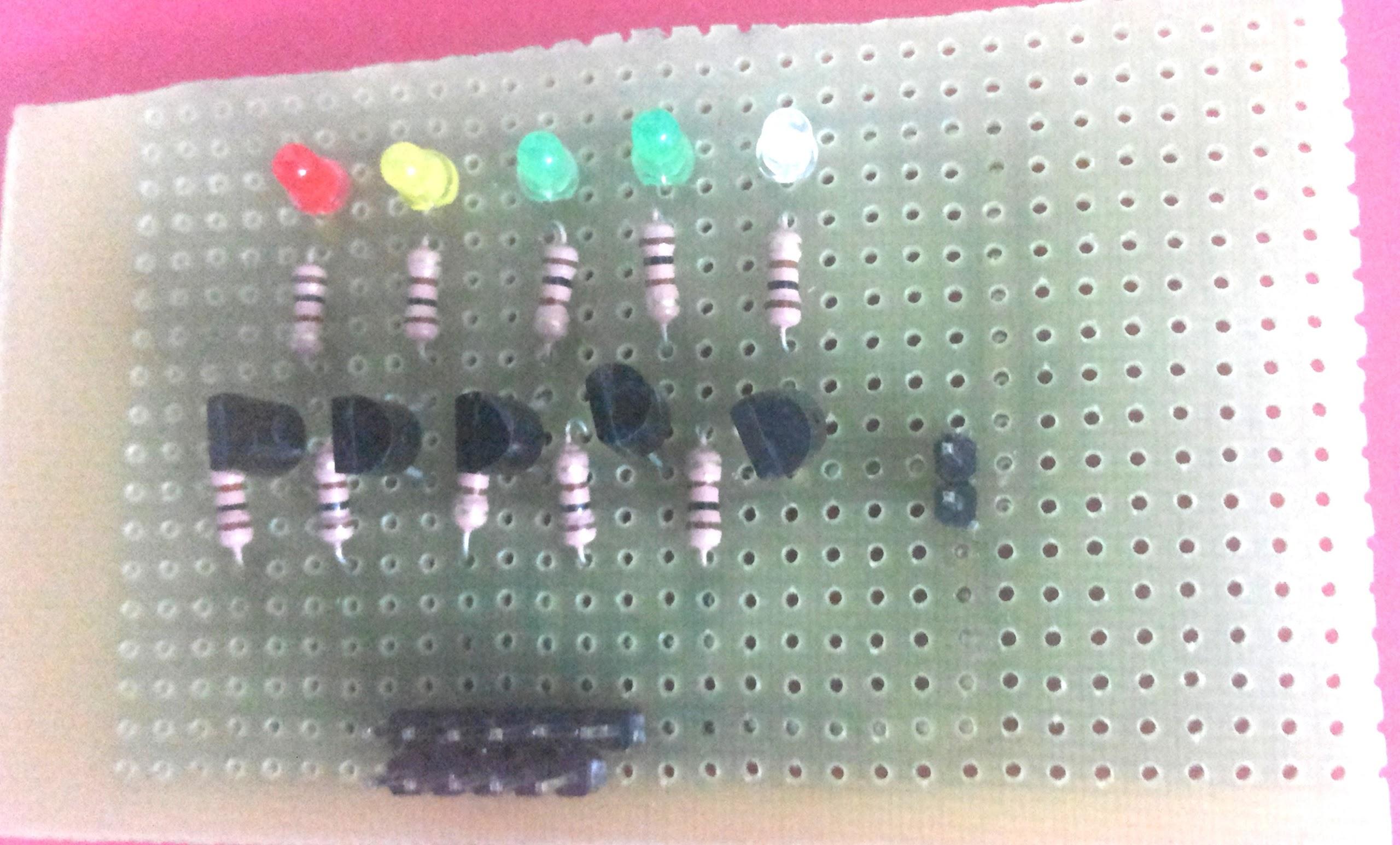
**Circuit diagram for water level sensor**

* The circuit shows water level for 3 levels.
* We have made this circuit for 5 levels.
* This can be further extended to any number of levels



**Snapshot of circuit of water level sensor**

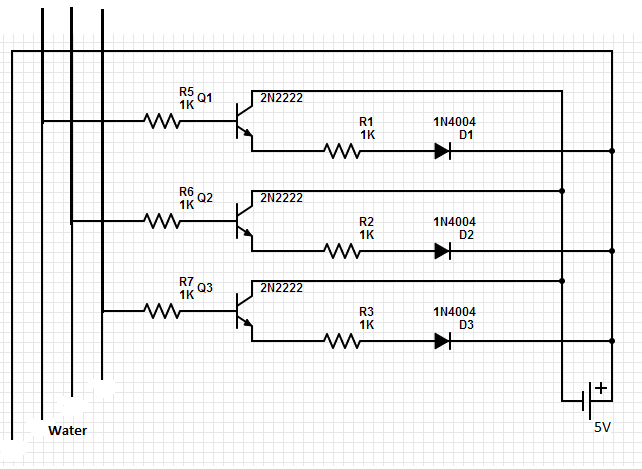
* This picture shows the water level sensor circuit



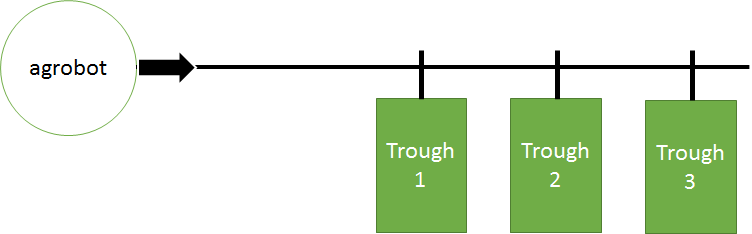
## **5. Working of the System and Test results**

**Working of the System**

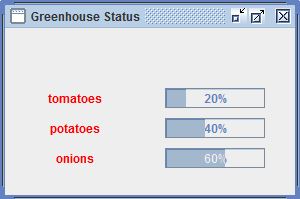
* **Sensing the Water Level** in the Troughs.
  + **Capacitive Continuous Sensor**
    - Initially, a capacitive sensor to measure the water level in the trough was tried.
    - A cylindrical capacitive sensor was made by wrapping aluminium foil around a straw and passing a metal wire in the straw.
    - These acted as the electrodes for the capacitor.
    - Water and Air were the dielectrics.
    - When the water level changes, the net capacitance changes.
    - The change in water level brought about a change in capacitance which in turn connected to a 555 timer circuit will bring a change in frequency.
    - This sensor required a lot of signal conditioning and reading of frequency values instead of conventional voltage values. Hence it wasn’t successful.
  + **Discrete Sensor**
    - The system currently uses a discrete water level sensor with 5 levels after the failure of above capacitive sensor
    - The circuit that was built consisted of transistor, LED, resistors, etc
    - The transistor acted as a switch for LED to glow when a particular water level comes up
    - The following figure shows the circuit for the discrete sensor



* + - The 3 probes in the top left are the point from which we get a signal when water level changes.
    - ADC is used to read the values from the above probes.
    - FireBird and XBee at the trough to transmit these water level values.
* **Live Tweeting** of water level from the troughs to the Central Server.
  + The bot reads the 2 sensor values from the troughs
  + Sensor Value 1 = a, b, c
  + Sensor Value 2 = d, e, f
  + a, b, c, d, e, f are 0 or 1
  + Byte transmitted = a + 2b + 4c + 8d + 16e + 32f + 33
  + 33 is added to binary equivalent of fedcba to make it printable
* **Transmission** of Trough IDs that need water from server to the Watering Bot
  + Currently the system works for 2 troughs each having a 3 level water-level sensor
  + The received byte is parsed and water level in each of the troughs is calculated
  + If the trough is **already full** or it is being **currently serviced** by AgroBot, the request is ignored
  + For each trough a new byte containing the data is created which is sent to the AgroBot
  + The first 4 bits of the byte are for the trough id and last 4 bits are for the water level
  + This byte sent to the AgroBot is then read and the trough which needs watering is identified.
* **Line Following** and Locating the trough on the line
  + Black tapes on the white lines denote the location of the trough
  + Each trough is located beside a central line corresponding to the above black tape on which AgroBot moves.



* **Watering Mechanism** using Pump and **Time-based Water Flow Control** 
  + Water pump has been used to draw water from the water tank on AgroBot
  + L293d Motor Controller Circuit is used for controlling the water pump
  + A 3.3 V signal from AgroBot to L293D Circuit turns the motor on and 0 V signal turns the motor off.
  + Calibration the water pump with time versus water flow was done.
  + Using this information and based on current water level in trough, duration of running the water pump was decided
* **Real time** displaying of water levels using user interface
  + Displays water level of troughs and refreshes each time updates occurs at central server
  + Following is the screenshot of the user interface at an instant.



**Functional requirements and Testing strategy-**

|  |  |
| --- | --- |
| **Functionality** | **Testing Strategy Employed** |
| * **Exchanging information about the water content in the trough with the server** - The troughs tweet the status of water level to the center server in continuous intervals | * 2 troughs are kept with different water levels.   + Completely empty   + Partially filled * The current status of water level is displayed on the firebird kept at the trough end. This verifies that we get the correct water level at the FireBird * The water level is changed after sometime and new status is again tweeted. * Water level readings obtained on the server verifies the working of the tweeting mechanism. |
| * **Transfer of message from server to AgroBot about the troughs that need water** - Using the data obtained by the center server from the troughs, it asks AgroBot to go and water those troughs | * Server has the knowledge of water levels in different troughs based on the above functionality * Once the water level of the trough reaches below a certain level, server sends a message to AgroBot about the trough that needs water. * This has been tested by reducing the water level in either of the troughs. The bot moves to the troughs with low water level and waters it. |
| * **Motion of AgroBot to the correct trough** - After getting the troughs which need watering, AgroBot uses its two-dimensional mapping of the GH and current location to move to the trough and water it. | * AgroBot has the knowledge of different troughs which require water. * It moves to the troughs following the line-follower path. * This is tested independently by sending the AgroBot to a particular trough |
| * **Proper amount of watering based on the water level in the trough** - AgroBot has a self-controlled battery-operated motor pump which helps in watering the trough at a constant rate. The watering is done for a specific time duration depending on the waterflow rate and amount of water needed in the trough. | * 2 troughs are kept with different water levels.   + Completely empty   + Partially filled * Depending on the current water level, the amount of watering to be done is calculated. * AgroBot then waters the trough for the required time duration. * This functionality is verified by completely filling 2 troughs with different water levels as mentioned in first point |
| * **Resetting the position on low water content level** - AgroBot moves to a default place in the GH whenever the water content in the tank goes low. After refill AgroBot resumes the task of refilling. | * We did not achieve the refilling part * This could have been achieved by using the same water level circuit that we have prepared. * The bot starts from a base position and after watering a trough, it reaches the position again. At the base, there can be a checking mechanism which waters the AgroBot if water level falls down. |

## **6. Discussion of System**

**What all components of your project worked as per plan?**

* **Sensing the Water Level** in the Troughs.
* **Live Tweeting** of water level from the troughs to the Central Server.
* Transmission of Trough IDs that need water from server to the Watering Bot
* Line Following and Locating the trough on the line
* Watering Mechanism using Pump and Time-based Water Flow Control
* Real time displaying of water levels using user interface

b) What we added more than discussed in SRS?

* We did not add anything more than discussed in SRS
* We only added the details on how the water level sensor was built and how the watering mechanism works which wasn’t mentioned in the SRS.

c) Changes made in plan from SRS:

* All the functional requirements have been achieved except the automatic refilling of the watering bot
* Besides these, there were minor changes in the greenhouse structure. Instead of 2-dimensional greenhouse, currently system works for a single line following path with troughs kept on regular interval denoted by black markings

## **7. Future Work**

**Enhancement of Re-Usable components**

* Water Level Sensor Circuit - 3

The number of levels in the water level sensor circuit could be increased from 5 to 8/16/n using the following methods

* **Shift Register**We can use a shift register to read the values of each of the n probes sequentially  
  Clock Signal of n ups and downs would be required in this case  
  *Overall, we need 1 pin instead of n pins in this case*
* **Multiplexor**

We can use a multiplexor(2^k to 1) with k control pin and 1 output pin  
 We need to set the control pins for all possibility and read each of the output

*Overall, we need 1 + log(n) pins instead of n pins in this case.*

* 3 Way Communication Protocol
  + If the number of troughs is increased, the 3 Way communication protocol can be extended using the mechanism of check before sending. Any communication should be preceded by a request of sending data and only if an ack is received, data transmission is started.
  + Currently the number of troughs is only 2 and acking is restricted to completion of a request. On increasing the number of troughs, better acking system with acking of each transmission can be implemented
* Watering System
  + Currently, the bot moves to each trough, waters it and goes back to the base position. That is the bot handles the requests of the troughs one at a time.
  + This mechanism can be improved by handling multiple requests at a time. Bot receives the information of all the troughs that need water and then using the best possible path which leads to minimum energy consumption, it waters all the troughs.
  + Also when the water level goes down, the bot reaches the base position where it refills itself automatically and then resumes the task of watering.
  + Also the current greenhouse system assumes the troughs to be present in a single line. A 2-dimensional network can also be served using the bot by modifying the line following module of the bot.

**Other Future Improvements**

* **Auto Refilling** of water in the AgroBot when the waterlevel goes low in AgroBot
* This model can be extended for **large farms** with **multiple troughs** and **multiple bots.**
* Use of **RFID Tag** on each of the trough to identify the **type of plant** and hence the **amount of water** required can be varied depending on the type of plant
* Supply of nutrients can also be done.
* Improvement of the user interface by including features like percentage of water consumed by each plant, etc

## **8. Conclusions**

With designing this model of a farm we try to ease the complexity and tediousness of regularly watering plants which nowadays has become very difficult given the busyness in this fast moving world. Since it is equally important that plants are taken proper care of, we proposed model of a green house which is maintained with a minimal surveillance and hence is highly likely to be adaptable in house lawns, small farms, and terrace farms particularly in cities where explicit supervising would be nearly impossible.

Also keeping in mind, the financial status of many people who want to avail this kind of a facility, we designed cheap water level sensors which made installing the model economically feasible. The model also efficiently spends water depending upon the necessity of plants.

This model is currently focusing on a small greenhouse. With the increase in the size of the greenhouse, the model can be extended to monitor each plant individually and study the water consumption by each of them.

There are also **re-usable components** of the system such as the water level sensor and the 3-way communication protocol which can be used in other projects as well.

## **9. References**

* Plant Watering Autonomous Mobile Robot, Hema N, Reema Aswani, Monisha Malik  
  <http://www.iaesjournal.com/online/index.php/IJRA/article/download/1305/pdf>
* <https://www.youtube.com/watch?v=uQj8GKQgQxg>
* http://gardenbot.org/howTo/soilMoisture/