CS101 Project Documentation and Report

FIREBIRD –V based Moisture sensing cum irrigation bot(IRRIBOT)

TEAM : 331

Mahen kinnur
140040024
Raj Krishnan 140010007
Shivank Kumar 140010024
Jagesh Golwala 14D070005

Table Of Contents

1. **Introduction**:

Today, with a decrease in the land available for farming, it is necessary to ensure that we get maximum efficiency and output from the available land. And irrigation is an important component of farming.

We see many crops getting destroyed every year due to inadequate water supply or due to excess rainfall. Although it is quite difficult to prevent the latter one, we can definitely do something to avoid the former reason.

And this has been our motivation for this project - A drip irrigation bot - where we have also taken into account another environmental factor - avoiding wastage of water. Thus. we have tried make a robot on a small scale which can solve two important environmental issues.

Our robot will stop on detecting a pot nearby. Using a soil moisture sensor, we will measure the moisture level of the soil. It gives us readings in the range of 0 - 255. Then based on the readings of the soil moisture sensor, we will irrigate the soil accordingly upto an optimum value.

Thus. we have added intelligence to our robot - wherein it waters the soil up to a required level and avoids extra watering.

2. **Problem Statement:**

We first aim that our robot follows a white line accurately. It can take sharp turns as and when necessary.

Then, we aim to stop a robot on seeing an obstacle on the right - we will keep pots on the right. Since it moves a little distance ahead after detecting a pot, we bring our robot a little back.

Following which, one of the robotic arms attached to a servo motor and having a soil moisture sensor on the other hand, is rotated upto an appropriate level so that the soil moisture sensor is properly inserted in the soil. We then wait for a few seconds allowing the soil moisture sensor to measure the soil moisture.

We now move the bot a little ahead so that our second robotic arm, to which we have connected a pipe, rotates to an appropriate level and waters the soil. We have moved the robot a little forward to adjust to our mechanical limitations so as to cover a distance between the two robotic arms.

Depending on the readings, we will rotate the second robotic arm for a time of either of 0 seconds, 1.5 seconds or 3 seconds.

3. Requirements:

A) Hardware Requirements

1. FireBird V
2. Moisture Sensor : To detect soil moisture content
3. Water Container and Pipe : Used to water the plant
4. Servo motors : To rotate the robotic arms
5. Sharp Sensor : To detect the presence of the obstacle in the right i.e. the pot

B) Software Requirements

AVR Studio and bootloader: To program instruction onto a given bot

4. Implementation:

A) Functionality:

a) **Determining and following the white line:** The white line underneath is determined using the white line IR sensors and using its ADC readings and comparing on white and black surfaces to determine the extent of the white line, and then using the readings to follow the white line using the algorithm which keeps track of the position of the bot relative to the white line and sends necessary commands to align the bot along the white line.

b) **Detecting obstacles :** the bot is interfaced with sharp IR sensors which determines pot as obstacles then the moisture arm and water arm actuates to water the soil c)**moisture level determination :**the moisture level is determined using the moisture level sensor which indicates the moisture in terms of ADC value(0-255).the higher the value the lower is the moisture content and in such a way the moisture level is categorized into high, low or medium moisture.

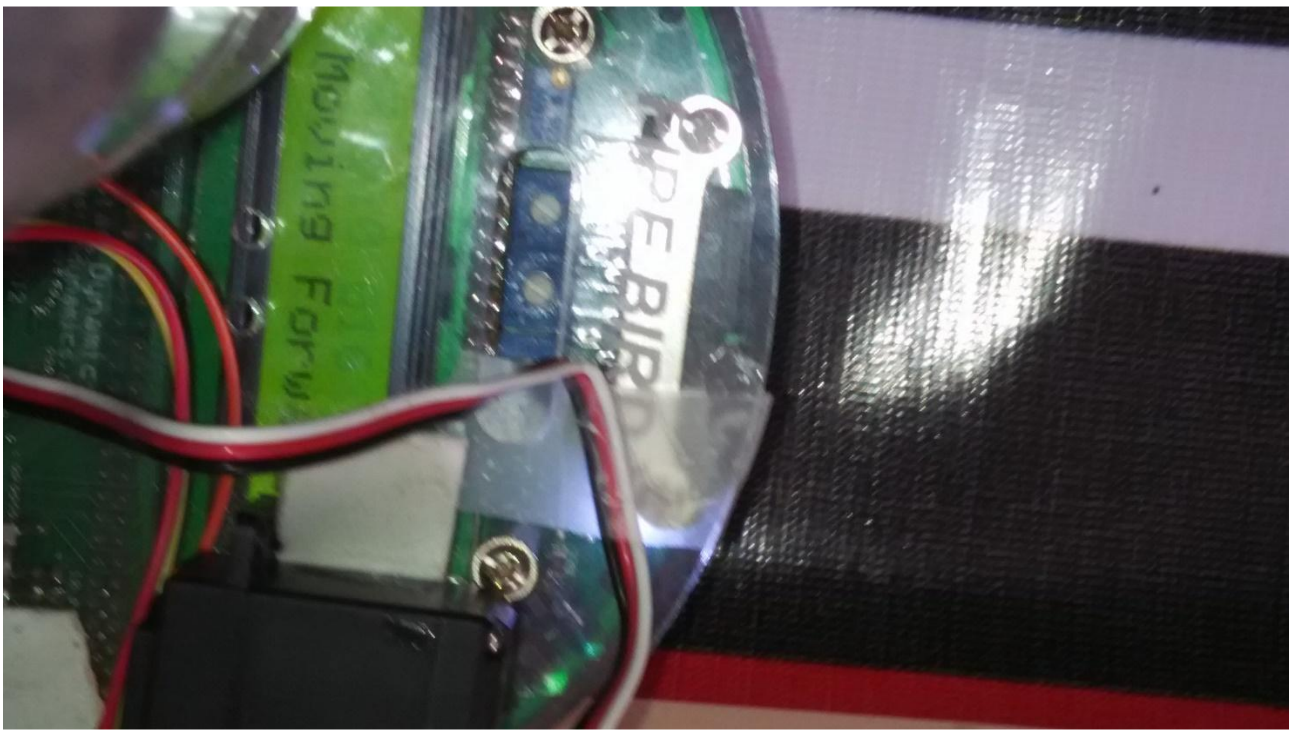
d)**Servo operation:** The servos connected with the water arm and the moisture sensor arm are operated once the right sharp IR sensor detects the pot .this helps in event–driven operation of measuring the moisture and watering the soil.

5. Testing Strategy and Data:

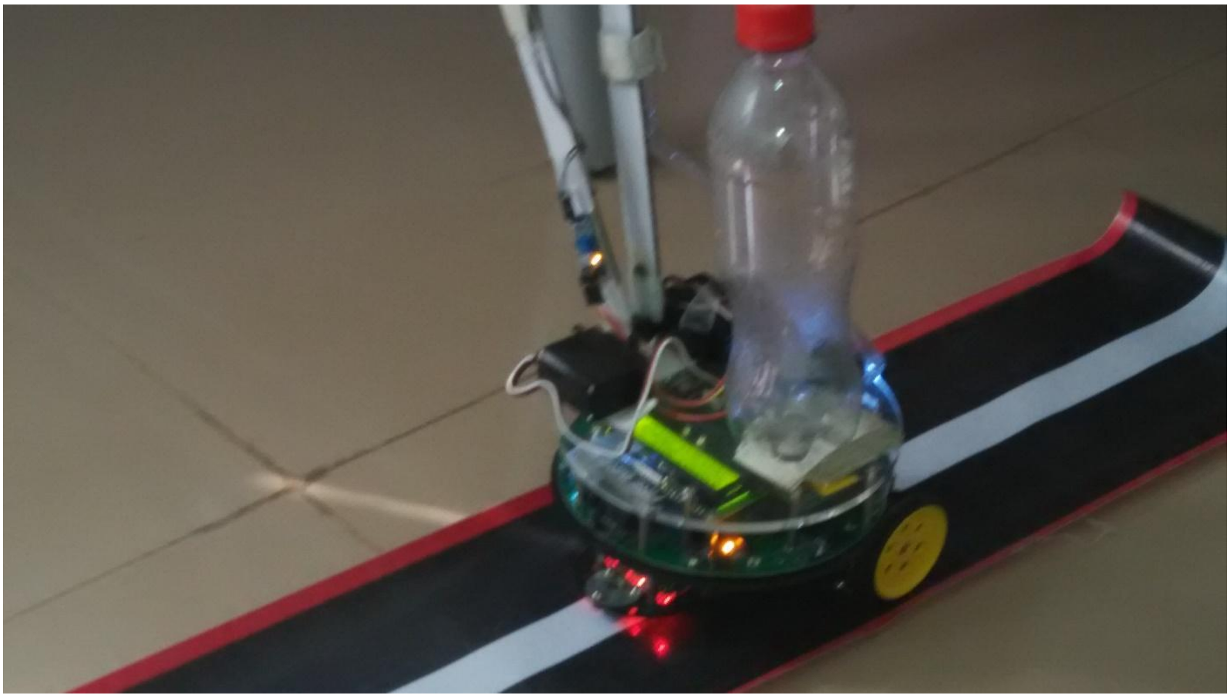
a) The white line sensors of the fire-bird constantly monitors its path whether it's following the white line or not. So the first testing strategy was to make the bot follow white line in all possible orientations for eg- when the bot is tilted from its path to right and to the left(not large deviation)

The bot is expected to turn and properly align itself along the white line drawn on the flex.Accordingly every command execution is displayed in the LCD display(for eg-taking a left turn , moving forward)

Bot orientation with LCD display:



Isometric view of the bot :

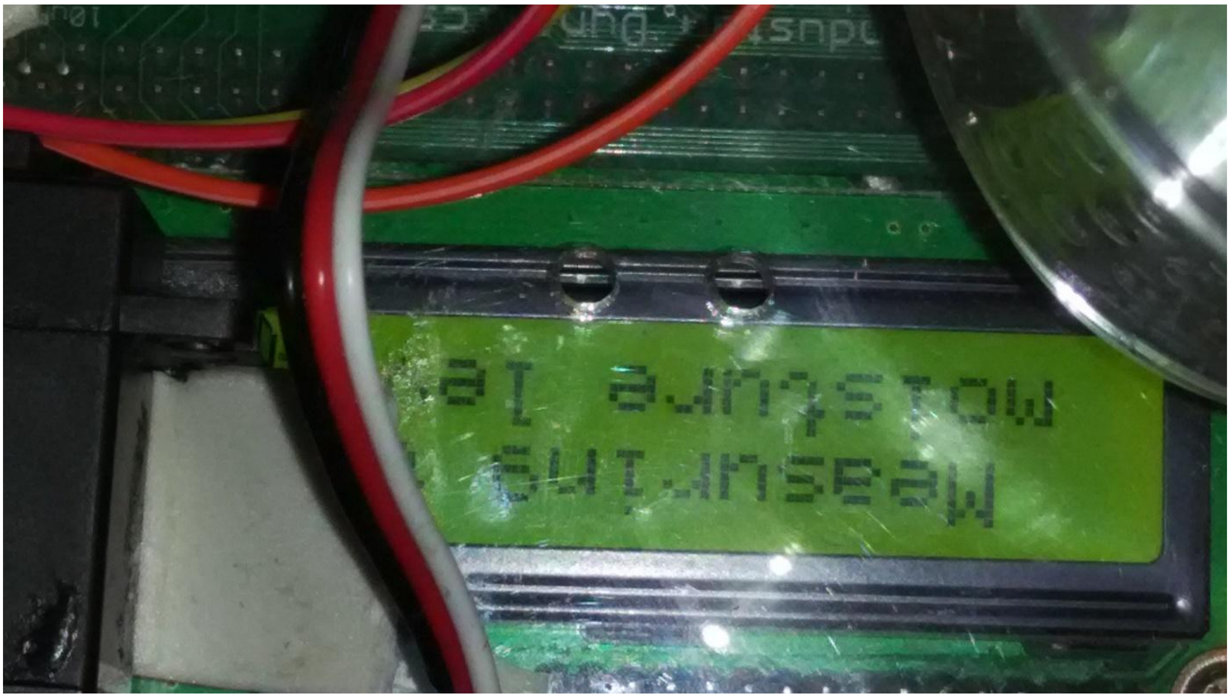


Situation:

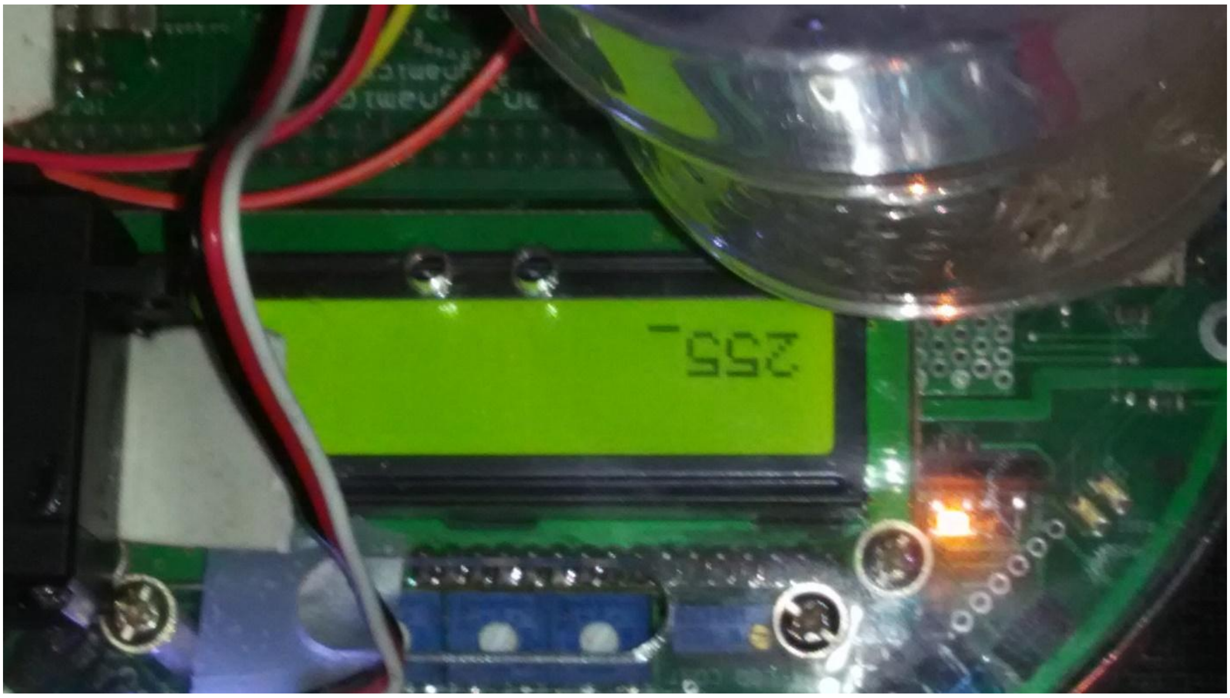
b) The second part was to asses the working of the 'right' sharp IR sensor that it indicates lower ADC values after an obstacle comes in the right and accordingly the line following fire-bird V has to stop.

the bot stops after sensing the pot:

Situation:
C) The moisture sensor interface is checked after proper connections.
We dipped the moisture sensor in wet soil and correspondingly it should return ADC values in the range 0-255
Measuring the moisture level:



Displaying the ADC value:



Situation :

c) We intended to stop the bot after detecting the front obstacle ,the front IR proximity sensor is calibrated in that way so as to make it stop after sensing an obstacle in the front .

LCD display when bot stops:

Situation :

d) We had to augment all the systems and now as the bot senses the right obstacle (pot) it should stop and consecutively take 3 readings of the moisture level in the soil and then lower the water arm to water the soil as per the following algorithm:

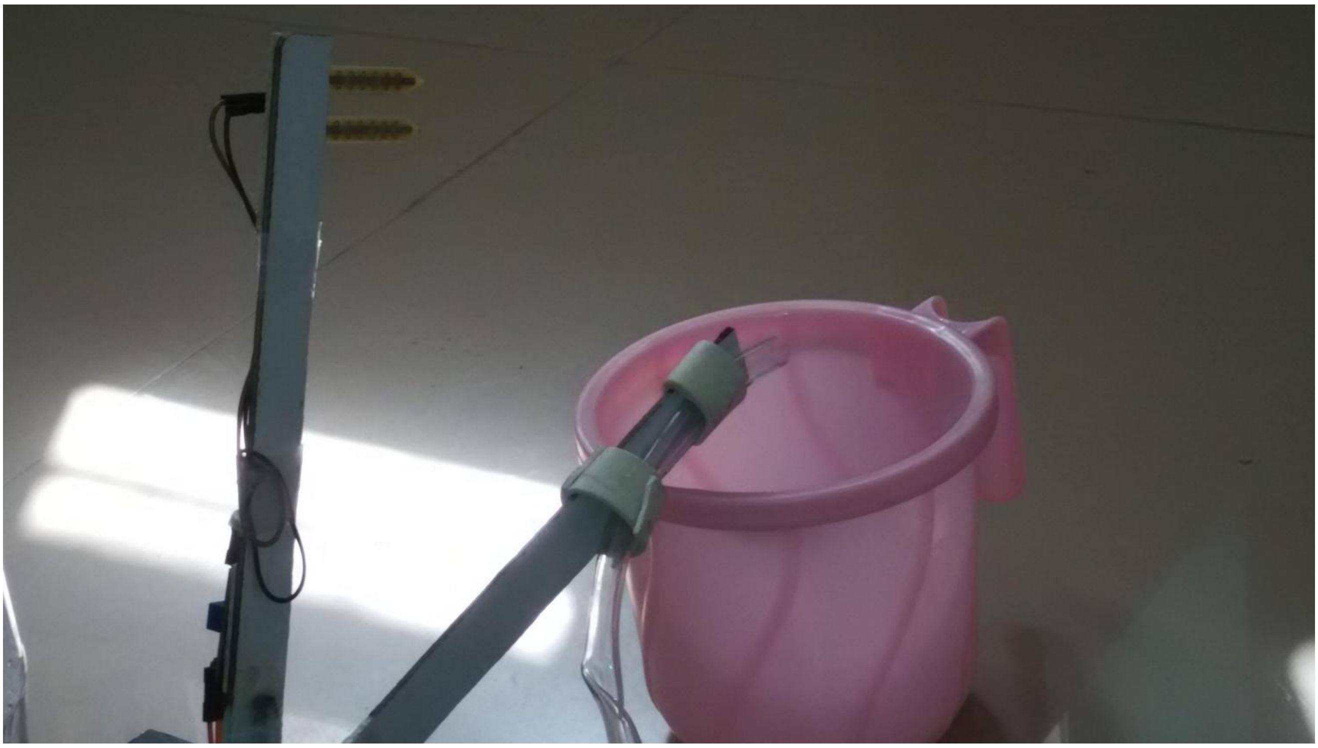
I) Low moisture (ADC{180-255}): need to water the soil for longer duration.

II) Medium moisture(120-180): need to water for a shorter duration.

III) Optimum moisture(<120): no need to lower the water arm as watering is not required.

Sensing the moisture level :

Lowering the water arm to water through the pipe:



6. Discussion of System:

A) What all worked as per plan?

1. The White Line follower :

Getting the ADC values from pin 1,2,3 is very crucial as it is used to sense the white line underneath . We selected values less than 16 in decimal and less than 10 in “hex” to follow the white line and it did well. Lighting conditions were optimal so the white line following part executed perfectly only excluding those extreme conditions of lighting-too bright and too light as well as too hot or too cold environments as the white line sensors are infact IR sensors.

2. Operating the right Sharp IR sensor:

Initially we had a difficulty in operating the sharp IR sensor as its calibration was difficult. But we were finally successful in doing that. We were able to detect the obstacle at a max range of 30cm and make the bot stop by setting its velocity to be zero.

3. Moisture Sensor interface:

Successfully able to interface the bot with the moisture sensor in analog mode and getting the moisture level values after ADC conversion in real time.

4. **Operating the servos**(Water arm and the moisture arm):

Simultaneous and independent control of both the servos was done successfully to operate the water arm and the moisture sensor arm with precision

B) What we added more than discussed in SRS?

Precise position of lowering of the moisture arm and the water arm:
In order to successfully water small plants in pots , we had to take the moisture level reading at the same position in the pot and lower the water arm at the same position also .this was done by first measuring the moisture level and then moving backwards for 300 ms (calibrated after a no.of attempts) and then lowering the water arm.

(C) Changes made in plan:

1)Water arm for watering :

We didn't get 5V dc motor for watering and thus we had to resort to alternative methods for watering which was just using the gravity

2) Single servo for moisture arm :

We had to use single servo for moisture arm as it we didn't get 5V dc motor to operate the watering mechanism.

And we couldn't operate more servos than 3(but it was much safer to use 2 considering power consumption)

3)couldn't decide about the water dispersal constant:

we were suggested to use ultrasonic sensors to measure the water level(which decides the water flow rate) but we were not able to use Ultrasonic sensors due to the multiple reflections from the sides which leads to closing this option.

Remedies: we made a rough estimate of the average of the water dispersal constant by experimenting a no. of times

4)stopping the bot at regular intervals:

Initially we thought of using the shaft count algorithm to decide the distance moved by the bot so as to stop the bot at such distances.

But then using the Right sharp IR sensor we could easily identify the obstacle and stop.

7. Challenges:

>>One challenge that we faced was to stop the robot on seeing an obstacle.

1] Initially we came up with an alternate solution, of stopping the robot either on regular time intervals and at regular distances.

2] Later on, we were successful in stopping the robot on seeing on obstacle in the front or on the right.

3] When it will see an obstacle in the front, it will make a 90 degree turn right ways.

4] And when it sees an obstacle on the right, it stops, reverses a bit and then servo motor starts rotating.

>>We have had a problem using the formula for calculating the duration for which the water arm

should remain down. This is because the time delay function in the robot can take only compile

time constant values. Hence we are now using two constant delays based on water content

>>The other challenge that we faced was to turn on the buzzer when the water level in the tank is below a certain level. For that, initially we thought of using an ultrasonic sensor, but due to multiple reflections from all the sides of the tank it was to difficult to use the readings to determine the water level. Since using an extra sensor and its corresponding circuit will lead to an increased power consumption, we decided to instead add an extra variable to turn the buzzer on. Assuming water dispersal rate

is constant, we will be adding a constant to the variable for each second there is water output, and once the variable exceeds a pre-decided value based on container capacity, we will turn the buzzer on to indicate “Low Water Level”

>>Also watering the soil was a bit difficult as we were not able to get the DC water pump required.

We managed to avoid using this by using a simple gravity powered mechanism due to which water flows when the arm is lowered below the water level in the tank.

8.Future Work:

- There are many expansions that can be done with respect to this robot, which are mainly oriented towards helping the farmers.
- Since the robot is going to move around the field, a web cam can be added, which is used to detect the presence of rodents or any other harmful bodies to the plants.
- Can also be used as a moving scare crow which also makes noises, a huge improvement over the present case.
- It can also be used to spray pesticides as desired by the farmer.
- There are other sensors available in the market which give the nutrient content of the soil. These can be attached to the robot, so as to get thee corresponding nutrient level.
- Also, a user interface (probably a user – friendly app) can be made, which can be used by the farmers to get the appropriate information.
- Also, they can set the optimum water level of the soil as per their wish and control the pesticide amount to be given to the soil.
- Add a water level sensor and a wireless module to alert the user when the water tank needs to be refilled.
- We can directly check the water level of the tank by using “Water level measurement Sensors”.
- By placing them at the required level in the tank, once the water level falls below that level, the buzzer will indicate that.
- The robot can be made self – sufficient in terms of power by placing a

solar panel on the robot, since the robot will carry out its work mainly during the daytime.

8. Conclusions:

Our work can be generailsed as **“Search + Action”.** An action can be anything depending upon application. There is lot more to it apart from just the irrigation part as we have described to you in the suggested future work over the bot. On our part we have tried our level best to make certain improvements in the so called future social issue “IRRIGATION”.

9. References:

1) Scarecrow Robot By E-Yantra Team : http://www.e-yantra.org/ci/projects/code/486

2) Watering model in the ERTS lab IIT Bombay.

3) Sensor module interfacing https://github.com/eyantra/Sensor-Module-Interfacing

4) Harvesting and sorting bot at e-yantra https://github.com/eyantra/CS684_Harvesting-and-sorting-Bot_2013

5)White line follower robot https://github.com/eyantra/White-Line-Follower

6)Project’s Github link: https://github.com/raj-krishnan/Irribot

Project Demo: https://youtu.be/pjOwyY-QRgs
Software Setup: https://youtu.be/bzmgr9DaA2M