



AGRICULTURAL ROBOT TRACTOR

Ms.Divya.R.Dhagate¹, Ms.R.S.Kadam²

^{1,2} *Electrical Department, Guru Gobind Singh Polytechnic Nashik .Maharashtra, (India)*

ABSTRACT

This paper describes the four major works in agriculture field which is done by a robot without giving any human support. For designing automatic system we provide proper tracking through the white line following robot concept in which the robot distinguishes white and black or dark surface and follows only white track. With reference to this concept we are going to implement a white line on a farm where actually we need to work & remaining surface is treated as a black surface due to the brownish color of soil. We also provide delay to robot at equal distance to do the agriculture work like pesticide spraying, water supplying, plugging, dropping of seed, accurately and automatically & the solar plant provides the supply source to whole robot which makes it ecofriendly. Through this we are going to integrate the different application of agriculture field which not happened in previous technologies and research

Keywords: *Automation, Robot, Microcontroller*

I. INTRODUCTION

The idea of robotic agriculture (agricultural environments serviced by smart machines) is not a new one. Many engineers have developed driverless tractors in the past but they have not been successful as they did not have the ability to embrace the complexity of the real world. Most of them assumed an industrial style of farming where everything was known before hand and the machines could work entirely in predefined ways – much like a production line. The approach is now to develop smarter machines that are intelligent enough to work in an unmodified or semi natural environment. These machines do not have to be intelligent in the way we see people as intelligent but must exhibit sensible behavior in recognized contexts. In this way they should have enough intelligence embedded within them to behave sensibly for long periods of time, unattended, in a semi-natural environment, whilst carrying out a useful task.

The approach of treating crop and soil selectively according to their needs by small autonomous machines is the natural next step in the development of Precision Farming (PF) as it reduces the field scale right down to the individual plant .One simple definition of PF is doing the right thing in the right place at the right time with the right amount. This definition not only applies to robotic agriculture (RA) but it also implies a level of automation inherent in the machines. Automatic sensing and control (on-the-go) for each task is also important and many research papers



have shown that these systems are feasible but most are too slow, and hence not economically viable, to be operated on a manned tractor.

Most of the current machinery is very weather dependent. Tractors cannot drive on soil when it is wet, sprayers cannot work in high winds etc. Perhaps it will be possible to develop smaller, less intrusive machinery that can allow more tasks to be carried out in marginal conditions. An example might be an autonomous seeder that could function well, while the soil is still wet in the springtime, provided that the soil engagement mechanism is suitably arranged. This would allow the seeds to be planted when optimal for the crop and not be limited by the soil's ability to support Safety is another important factor. Any autonomous vehicle is going to go wrong at some time and the chance of catastrophic failure should be minimized within the design process. A small light vehicle is inherently safer than a large one. Redundant, self-checking systems should be built into the system architecture to allow graceful degradation. The vehicle should be in continual communication with the base station, giving data about current conditions and contexts.

This approach may not be economically justifiable in many broad acre crops but will certainly be more attractive in high value crops where a smart machine can replace expensive repetitive labor. If this approach were taken, it would appear that the crop production cycle could be reduced to three stages: Seeding, Plant care and (selective) harvesting. As the numbers of students are focusing on their higher studies so there is reduction in number of farmers day by day. So using this technology young generation can not only do studies but also simultaneously they can focus on their farming. And also they don't need sell their land or farm to anyone.

II. HARDWARE DESCRIPTION

2.1 PIC Microcontroller

The controller is the heart of entire system, and the whole system should be analyzed in selecting the proper controller. The following items should be considered when selecting a controller:

Type of Input

Type of Output required

Control algorithm needed

Number and Type of Outputs.

We are using PIC 18f4550 Microcontroller. It is a 40 pin Microcontroller with 128 bytes RAM & 4kb flash memory.

Features:

1. Full Speed USB 2.0 (12Mbit/s) interface
2. 1K byte Dual Port RAM + 1K byte GP RAM
3. Full Speed Transceiver
4. 16 Endpoints (IN/OUT)
5. Streaming Port



6. Internal Pull Up resistors (D+/D-)
7. 48 MHz performance (12 MIPS)
8. Pin-to-pin compatible with PIC16C7X5

2.2 DC Motor

Electric motors are used to “actuate” something in your robot: its wheels, legs, tracks, arms, fingers, sensor turrets, or weapon systems. DC-motors are very easy to use, but like most other motors their usefulness for robotics is very dependent on the gearing available. DC-motors are made much more effective if they have an efficient gear ratio for a particular task. DC motors are widely used in robotics because of their small size and high energy output. They are excellent for powering the drive wheels of a mobile robot as well as powering other mechanical assemblies.

Ratings and Specifications:

Several characteristics are important in selecting a DC motor. The first two are its input ratings that specify the electrical characteristics of the motor.

➤ **Operating Voltage:**

If batteries are the source of power for the motor, low operating voltages are desirable because fewer cells are needed to obtain the specified voltage. However, the electronics to drive motors are typically more efficient at higher voltages. Typical DC motors may operate on as few as 1.5 Volts or up to 100 Volts or more. Robotics often use motors that operate on 6, 12, or 24 volts because most robots are battery powered, and batteries are typically available with these values.

➤ **Operating Current:**

The ideal motor would produce a great deal of power while requiring a minimum of current. However, the current rating (in conjunction with the voltage rating) is usually a good indication of the power output capacity of a motor. A low voltage (e.g., 12 Volt or less) DC motor may draw from 100 mA to several amperes at stall, depending on its design.

The next two ratings describe the motor's output characteristics:

➤ **Speed:**

Usually this is specified as the speed in rotations per minute (RPM) of the motor when it is unloaded, or running freely, at its specified operating voltage. Typical DC motors run at speeds from one to twenty thousand RPM. Motor speed can be measured easily by mounting a disk or LEGO pulley wheel with one hole on the motor, and using a slotted optical switch and oscilloscope to measure the time between the switch openings.

➤ **Torque.**

The torque of a motor is the rotary force produced on its output shaft. When a motor is stalled it is producing the maximum amount of torque that it can produce. Hence the torque rating is usually taken when the motor has stalled and is called the stall torque. The motor torque is measured in Newton-meters (metric). The torque of small electric motors is often given in milli-Newton-meters (mN-m) or 1/1000 of a N-m.



➤ Power

The power of a motor is the product of its speed and torque. The power output is greatest at about half way between the unloaded speed (maximum speed, no torque) and the stalled state (maximum torque, no speed). The output power in watts is about $(\text{torque}) \times (\text{rpm}) / 9.57$.

Technical Specification:

- Voltage: 12V DC
- Current: 1.5A

L293D:

The L293D motor driver is available for providing User with ease and user friendly interfacing for embedded application. L293D motor driver is mounted on a good quality, single sided non-PTH PCB. The pins of L293D motor driver IC are connected to connectors for easy access to the driver IC's pin functions. The L293D is a Dual Full Bridge driver that can drive up to 1Amp per bridge with supply voltage up to 24V. It can drive two DC motors, relays, solenoids, etc. The device is TTL compatible. Two H bridges of L293D can be connected in parallel to increase its current capacity to 2Amp.

Features:

- Easily compatible with any of the system
- Easy interfacing through FRC (Flat Ribbon Cable)
- External Power supply pin for Motors supported
- Onboard PWM (Pulse Width Modulation) selection switch
- 2pin Terminal Block (Phoenix Connectors) for easy Motors Connection
- Onboard H-Bridge base Motor Driver IC (L293D)

Technical Specification:

- Power Supply : Over FRC connector 5V DC
- External Power 9V to 24V DC
- Dimensional Size : 44mm x 37mm x 14mm (l x b x h)
- Temperature Range : 0°C to +70 °C

L293D IC:

The driver IC L293D is quad push-pull drivers capable of delivering output currents to 1A per channel respectively. Each channel is controlled by a TTL-compatible logic input and each pair of drivers (a full bridge) is equipped with an inhibit input available at pin 1 and pin 9. The motor will run only when chip inhibit is at high logic i.e. chip inhibit is enabled. The connection diagram is shown below:

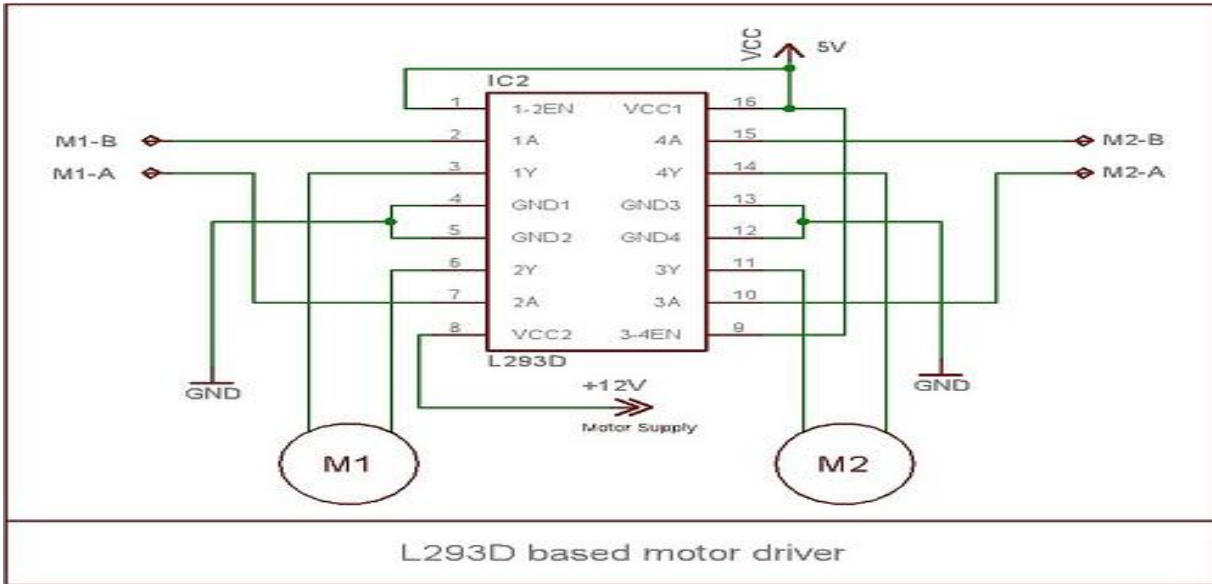


Figure 01.L293D DC Motor Driver Circuit.

Table.1. Control Pin Functions.

Enable	M1-A	M1-B	Function
High	High	Low	Turn right
High	Low	High	Turn Left
High	Low/ High	Low/ High	Stop
Low	Either	either	Stop

III. SOFTWARE DESCRIPTION

The software to control the robot is written in C language. It will be developed in the MPLAB development environment which is provided by Microchip Technology Inc. The C code is compiled with a compiler from HI-TECH PICC C compiler. This compile is integrated into the MPLAB IDE (Integrated Development Environment) platform to ease the design process. Like any other embedded system developing procedure, you need to create a project, then develop the code for the project and compile it. After that you program the microcontroller using the developed program.

The following is the brief tutorial about how to

- Create the project, add files to the project,
- Compile the files,
- Program the robot with the compiled C program.

Create a Project:

First create a new directory called *robot Lab* in your *H* or *C* disk, download a C file called *line follower lab.c* the lab webpage. Right click the link and save it to the directory you just created.

Launch the MPLAB IDE software by clicking Start -> All Programs -> Microchip -> MPLAB IDE V8.10 -> MPLAB. This will start the MPLAB development environment. Create a project in MPLAB. Select the Project Wizard menu item from the Project pull-down menu as illustrated in the following Figure.

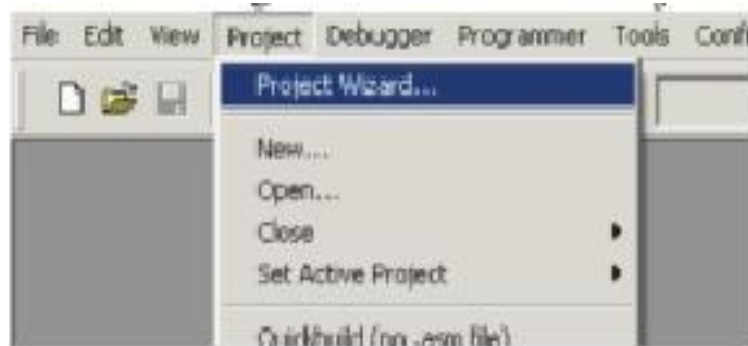


Figure.02. Project Wizard.

In the dialog that opens, click Next to move to the chip selection dialog. Select a PIC18F4550 device then click next. The next dialog is where you get to specify the tool suite associated with the project. Select HI-TECH Universal Tool suite as the Active Tool suite in this dialog, as shown in Figure. Notice that the Tool suite Contents area shows the generic name HI-TECH ANSI C Compiler. Notice also the path shown in the Location area. This path is not important and you do not need to edit it to point to the compiler location. The Universal Tool suite determines the location of the installed compilers via other means.

Click next

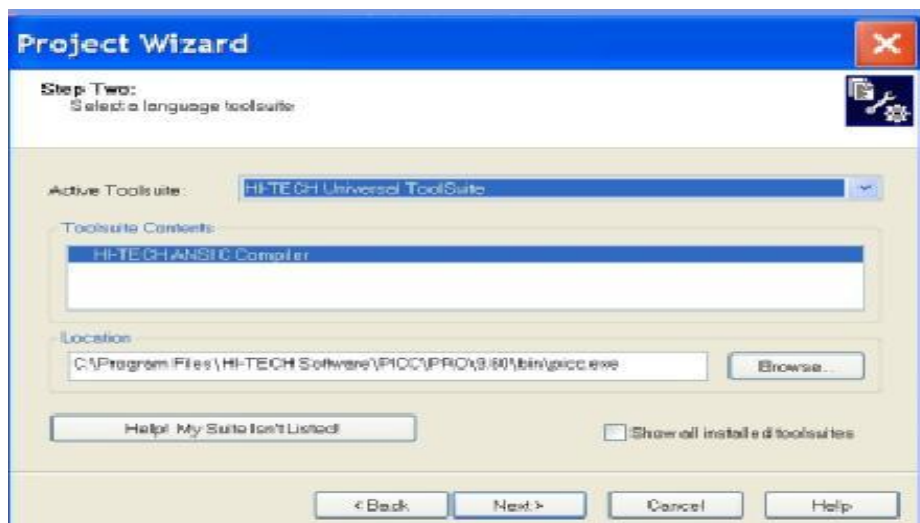


Figure.03. Select HI-TECH Universal Tool Suite in creating project



In the next dialog you can specify the name and location of the project you are creating. Click on *Browse* and change the directory to robot Lab that you have created at the beginning. Type in a meaningful name such as line follower for the project and Click Next.

Lastly the Project Wizard is asking to see if you would like any existing files to be added to the project. We will add the downloaded file in the directory called line follower lab.c to the project. Select the file from the left panel and click on Add to add it to the project. The file should show in the right panel at this time. Then click next and show the project summary. If everything looks okay in the summary click Finish and we are done for this part.

Compile the code:

To ensure that the file was correctly added, click the View menu, and select Project. This will open a window which shows an overview of the project. In this window you will see several folder icons. Under the Source Files icon you should see the name of the file listed next to a file icon. You can double-click the file to open and edit it. If you close the editor window containing our code, you can re-open it by double-clicking the file icon associated with the file in the Project window.

Now let's compile the downloaded code. Select Rebuild from the Project menu, or choose any of MPLAB IDE's short cuts to build the project — you can, for instance, click on the toolbar button that shows the HI-TECH "ball and stick" logo. You will notice that the Project menu has two items: Build and Rebuild. The Build menu item action only processes those source files that have changed since the last build, then performs the code generation and link step. Rebuilding a project will always process every source file in the project, regardless of whether they have changed. If in doubt, use Rebuild. The HI-TECH Build buttons are linked to the Build menu item. The program should now be compiled. You will see a window open showing content similar to that shown in Figure below.

```
Build | Version Control | Find in Files | PICKit2
Executing: "C:\Program Files\HI-TECH Software\PICC\PRO\9.60\bin\picc.exe" -pass1 Z:\EE292
Executing: "C:\Program Files\HI-TECH Software\PICC\PRO\9.60\bin\picc.exe" -olinefollower.cof
HI-TECH C PRO for the PIC10/12/16 MCU family (Lite) V9.60PL3
Copyright (C) 1984-2008 HI-TECH SOFTWARE
(1273) Omniscent Code Generation not available in Lite mode [warning]

Memory Summary:
Program space   used  2B3h ( 691) of  2000h words (  8.4%)
Data space      used   8h (  8) of  170h bytes (  2.2%)
EEPROM space    used   0h (  0) of  100h bytes (  0.0%)
Configuration bits used  1h (  1) of   2h words ( 50.0%)
ID Location space used   0h (  0) of   4h bytes (  0.0%)

Running this compiler in PRO mode, with Omniscent Code Generation enabled,
produces code which is typically 52% smaller than in Lite mode.
The HI-TECH C PRO compiler output for this code could be 359 words smaller.
See http://microchip.htsoft.com/portal/pic\_pro for more information.

Loaded Z:\EE292\motortest\linefollower.cof.

***** Build successful! *****
```

Figure.04 Information after the code is successfully built

These shows the steps both MPLAB IDE and the compiler took to build the project. The lines starting with Executing: show the command lines passed to the compiler command-line driver, PICC that were used to actually build. Note that there are two commands: one for the only source file the project contains; the other for the code generation and link step. The compiler has produced a memory summary and there is no message indicating that the build failed, so we have successfully compiled the project.

If there are errors they will be printed in Build tab of this window. You can double-click each error message and MPLAB IDE will show you the offending line of code, where possible. If you do get errors, check that the program is what is contained in this document.

Program the Robot:

The programmer you are using is called PICKit 2. Plug this programmer into the robot programming connector (note pin 1) on the board and also connect it to your PC with the USB connector. Turn on the power switch of the robot.

At this time, click on Programmer -> Select Programmer -> PICKit 2. Then you are ready to program the robot with the compiled code. Click on Programmer -> Program, the robot should be successfully programmed as shown in the following Figure. This means that the program is downloaded to the memory of the robot and ready to run.

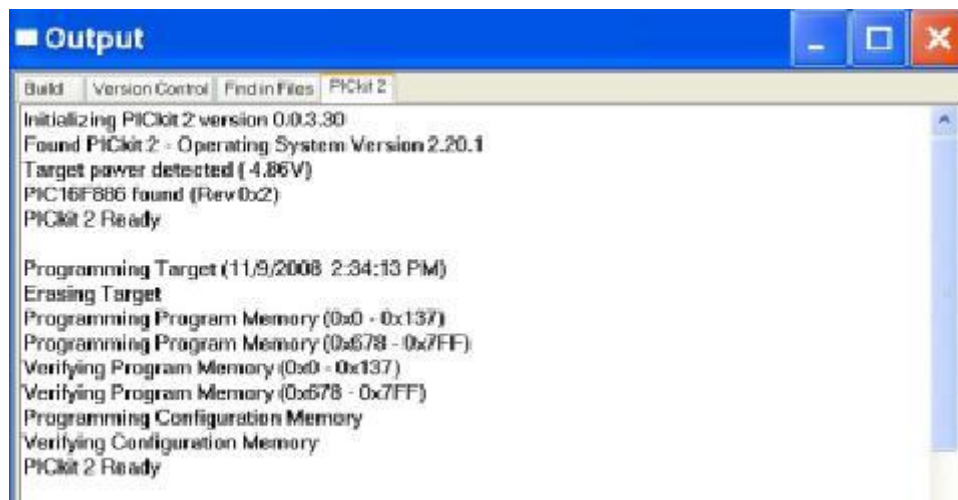


Figure.05. Programming the robot

If you want to program the robot every time you have successfully compiled the code, you can select Program -> Settings -> Program after every successful build.

IV. INTERFACING AND MODELLING

Introduction to the Hardware:

The hardware includes a mobile robot and a sensing board which are shown in the Figure. The long board on the top of the robot is the control board which controls the robot. There are four wheels and three motors to drive the robot and one motor to control the tool. One 12V battery is used to supply the motor and the control board. The green LED on the control board is for power supply indication. A switch is connected to switch on or off the supply to the control board and the motor.



Figure 06. A Picture of the mobile robot.

Robot:

The mobile robot includes the control board, batteries, four motors and wheels. The schematic of the robot is shown in Figure below. The “heart” of the control board is a PIC18F4550 microcontroller. Data sheets and information on this chip are discussed in previous chapter. This chip contains a wide variety of useful peripherals on board such as: analog-to-digital converters (convert analog signals to digital signals), pulse-width-modulation (generate modulated voltage pulses whose width and period can be controlled), comparators, UART (for serial communications), etc.

PORTA (RA0-RA7) and PORTB (RB0-RB7) of the microcontroller are 8-bit wide, bidirectional ports. The corresponding data direction registers are TRISA and TRISB. Setting a TRISA bit (=1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output. It is the same for TRISB.

For the robot, PORTD of the microcontroller are all configured as output and digital pins. RD0, RD1, RD2, RD3, RD4, RD5, RD6, RD7 AND RD8 are connected to the L293D to control the DC motor. The DC motor operation is controlled by the command given by the user or driver through the PC. The command to the microcontroller is sent with the help of Zig Bee to the rx/tx pin of the microcontroller. Back two wheels motors are connected in parallel and connected to the L293d and for front wheel one motor is used for the two wheel as gear arrangement is there. The gear is connected to the shaft of the motor. In this robot, we have not controlled the speed of the motor, we are only switching the motor on and off but we can control the speed of the motor using pulse width modulation. We can also interface the sensor to the robot depending on the application. We have prepared prototype in which we have only taken only one application of tool up and down for digging the farm. A LCD is used to display the motion of the robot.

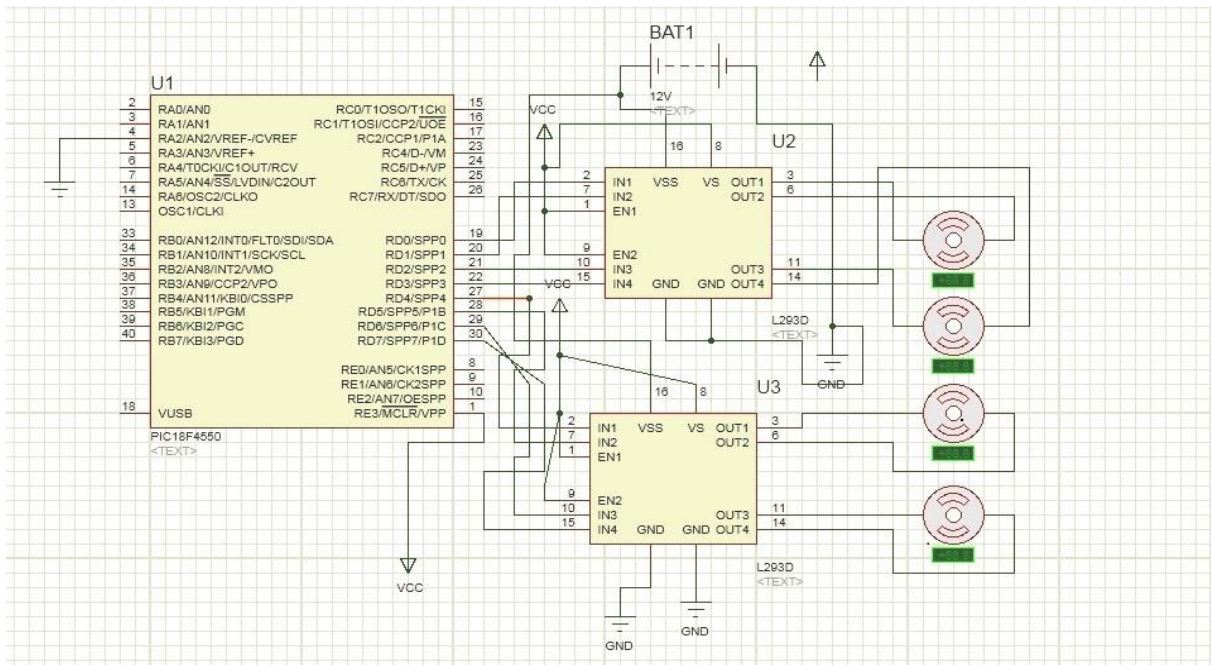


Figure.07. Schematic of the control board.

V. CONCLUSION

An initial outcome from this study indicates that most of these autonomous systems are more flexible than conventional systems and may reduce labour costs and restrictions on the number of daily working hours significantly. Moreover, it is possible to substitute the most trivial working routines with autonomous systems although some routines are nearly impossible to automate due to the required accuracy of the specific tasks.

Our prototype Agricultural Robot Communicates through Zig Bee technology. These vehicles include an electronic robot vehicle that can provide cropping formation using sensors and robot tractor capable of performing various implement function. Application of inexpensive navigation sensors to the robot farming system makes the system economically adaptable with the environment. With the development of robot farming system, food production can be increased considerably and economically.

REFERENCES

- [1] PIC Microcontroller and Embedded System by M.A.Mazidi
- [2] Programming and Customizing The PIC Microcontroller by Myke Predko
- [3] Babu, R., *Mitigating the correlations in INS-aided GPS tracking loop measurements: A Kalman filter based approach*, in *17th Int. Tech. Meeting of the Satellite Division of the U.S. Institute of Navigation*. 2004: Long Beach, California.
- [4] Babu, R. and J. Wang, *Improving the quality of IMU-derived Doppler estimates for ultra-tight GPS/INS integration*, in *GNSS2004*. 2004: Rotterdam.



- [5] Bongiovanni, R. and J. Lowenberg-Deboer, *Precision Agriculture and Sustainability*. Precision Agriculture, 2004. 5: p. 29.
- [6] R. Howitt, J. Medellin-Azuara, and D. MacEwan, "Measuring the employment impact of water reductions," Univ. California, Davis, CA, 2009.
- [7] J. S. Famiglietti, M. Lo, S. L. Ho, J. Bethune, K. J. Anderson, T. H. Syed, S. C. Swenson, C. R. de Linage, and M. Rodell, "Satellites measure recent rates of groundwater depletion in California's central valley," *Geophys. Res. Lett.*, vol. 38, p. L03403, 2011.
- [8].R. Seager, M. Ting, I. Held, Y. Kushnir, J. Lu, G. Vecchi, H. P. Huang,