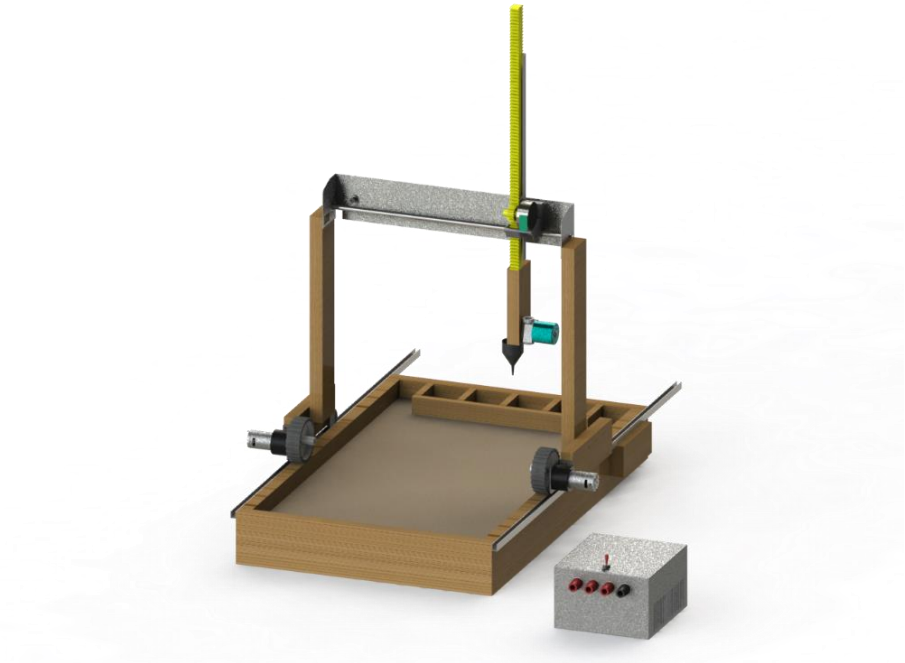
PROJECT REPORT
FARMBOT



Group Members

Mentor:

Prof Neelu Jain

Aarohi Kumar	14105015
Avi Dubey	14105048
Karan Thakur	14105058
Subham Roy	14105071
Haider Zulfiqar	14105075

Index

ACKNOWLEDGEMENT

It is indeed a matter of great pleasure and proud privilege to be able to present this project on "Agricultural Farmbot". We would like to take this opportunity to express our respect and deep gratitude to our guide **Professor Neelu Jain** for giving all necessary guidance required for this project, apart from being a constant source of inspiration and motivation. It was our privilege to have worked under them.

We are also thankful to the minor project coordinator **Prof Jyoti Kedia** for providing us relevant information and necessary clarification. We are also thankful to **Prof Neena Gupta** (Head, Electronics and Communication Department) for the regular guidance, cooperation, encouragement and kind help.

Aarohi Kumar
Avi Dubey
Karan Thakur
Subham Roy
Haider Zulfiqar

Abstract

Agriculture is one of the oldest activities known to man. Growing crops for food, medicine, fuel is the most important economic activity for our survival. With the growth in population, the required agricultural efficiency is estimated to be increased up to 25% more than the present figure.

The project “Farmbot” is a prototype for increasing the efficiency in agriculture, specially designed for farming practices of India. The project uses Raspberry Pi and Arduino Nano as the controller. The robot is based on the concept of a Cartesian manipulator. The robot is capable of moving inside a defined 3-dimensional space. It is capable of both manual and automatic control. The robot performs basic agricultural functions like sowing seeds, watering them, and tracking the growth of individual plants.

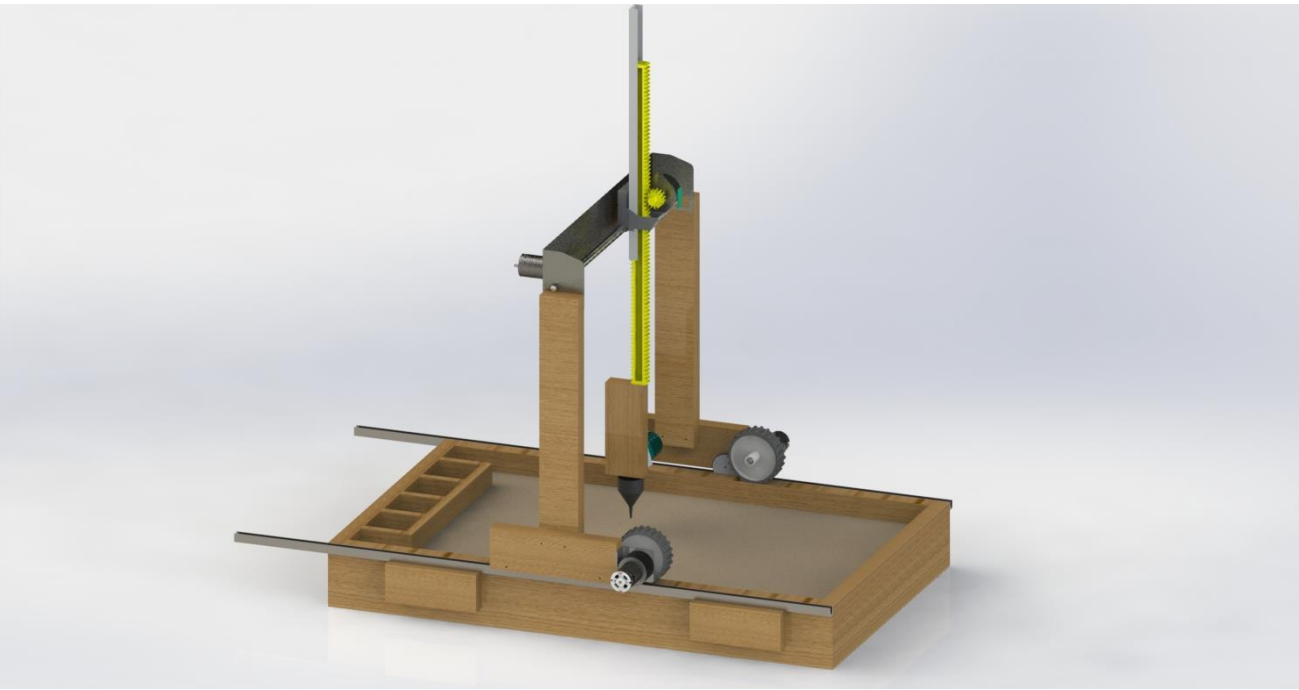


FIGURE 1 : A 3D RENDERED VIEW OF THE FARMBOT

Chapter 1
Introduction

The Farmbot project is inspired by the Genesis v1.0 robot [1] which achieves a similar automation in farming. But, the cost of this robot is not what an average Indian person could afford. The latest version is priced at $3900 or roughly Rs. 2,61,000. Moreover, the Genesis bot is aimed at gardening and small-scale agriculture automation, which is, in our opinion, inadequate for the food crop oriented agriculture practices of India.

Our Farmbot is designed keeping the above point in mind. The differentiating factor is the cost, at around Rs. 10,000, it is a mere 5% of what the Genesis costs.

1.1 Project Overview

The aim of this project is to make an autonomous agricultural robot to automate agricultural tasks with minimal human intervention. The main objective is to minimize human effort in growing food crops.

- The robot is a 3-dimensional Cartesian manipulator, which means it is able to move in a defined 3-dimensional space along x, y and z coordinates.
- Raspberry pi board is used as the main controller for the robot. Thus, the robot is capable of autonomously decide the course of action required.
- Arduino Nano is used for precise numeric control of the robots moving head.
- The project uses both open-loop and closed-loop feedback systems for precision movement.
- The user is also able to connect with the robot using Wi-Fi technology and send manual commands, if the need be.

1.2 Motivation for this project

Robotics and automation can play a significant role in society meeting 2050 agricultural production needs. For six decades, robots have played a fundamental role in increasing the efficiency and reducing the cost of industrial production and products. In the past twenty years, a similar trend has started to take place in agriculture, with GPS and vision-based self-guided tractors and harvesters already being available commercially. More recently, farmers have started to experiment with autonomous systems that automate or augment operations such as pruning, thinning, and harvesting, as well as mowing, spraying, and weed removal [2].

The driving factor behind the project was the need of autonomous farming robot in a developing nation like India which is on the path of agricultural and technological development. In a nation like India, agriculture is perceived as a backward or sub-normal occupation. Thus, people don't prefer to take it up as an occupation. The problem is two-fold-

1. Due to several factors leading to hardships in this occupation, the number of farmers is on a decline.
2. Due to a skewed penetration of technology in agriculture, development in agriculture is not vast as it is in other fields like service and manufacturing.

So, the Farmbot is our humble contribution to minimize this skew in the agricultural field.

Chapter 2
Working Mechanism and Mechanical Systems

2.1 Structural Representation

The structural representation of the robot is as follows

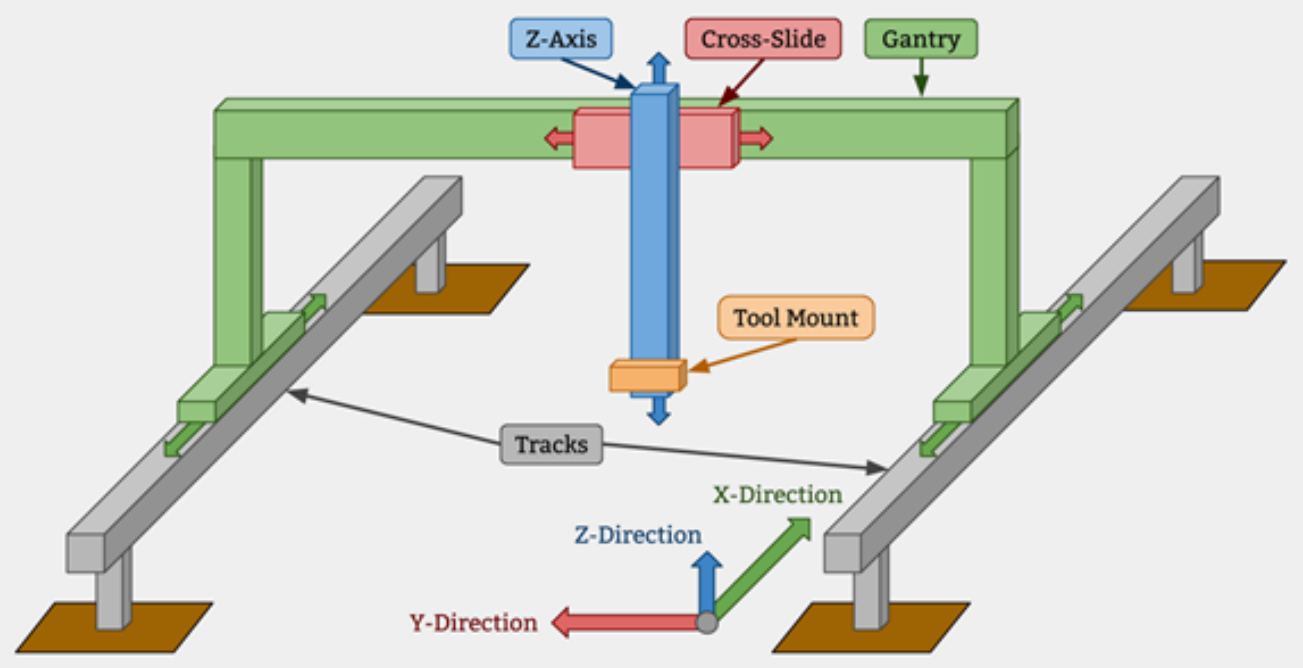


FIGURE 2 : STRUCTURAL REPRESENTATION OF FARMBOT

The structure is based on the concept of Cartesian coordinate manipulator. In this the principal axis of movement of control are linear or along x, y and z Cartesian coordinates. Thus, there are 3 degrees of freedom. This mechanical arrangement simplifies the motion and controlling equipment. Cartesian coordinate robots with the horizontal member supported at both ends are called Gantry Robots. They resemble gantry cranes in general. These robots are widely used in pick and place mechanisms and plotters. [3]

Farmbot is based on the gantry robot concept. The x, y and z axis of movement is shown in figure 2. The mechanical concepts are descripted in the further sections.

2.2 Movement in x-direction

The Farmbot has a gantry system as shown in Figure 2, which will move on the horizontal tracks, in x direction with the help of two Johnson motors. The gantry is fixed with respect to the body or chassis of the bot. The gantry consists of a printer bed which has a carriage moving in y direction only.

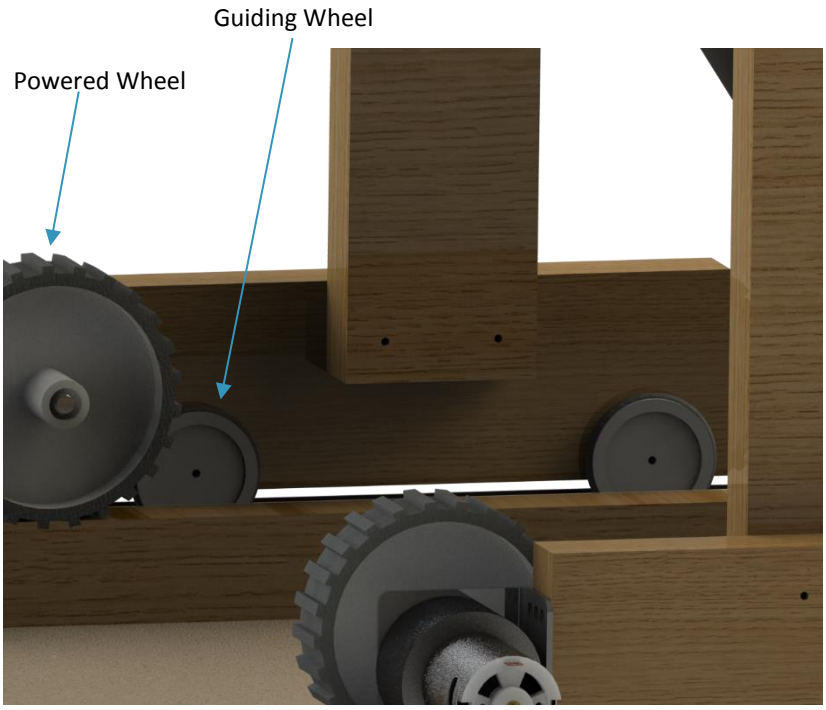


FIGURE 3 : CLOSE UP VIEW OF GUIDING WHEELS

A rail and guiding wheels are used to keep the gantry aligned with the body. The guiding wheels and rail are shown in figures 3 and 4. The small guiding wheels are unpowered, i.e. without driving motors. For pushing the gantry, two bigger wheels are connected to two Johnson Motors. The Johnson motors are driven through a high-current motor driver, L298. The working of the driver is explained in upcoming section.

To keep the movement of gantry steady and balanced, the powered wheels are accompanied with rotary encoders which form a closed loop feedback system along with the controller (Arduino Nano). The optical encoders are explained in upcoming section.

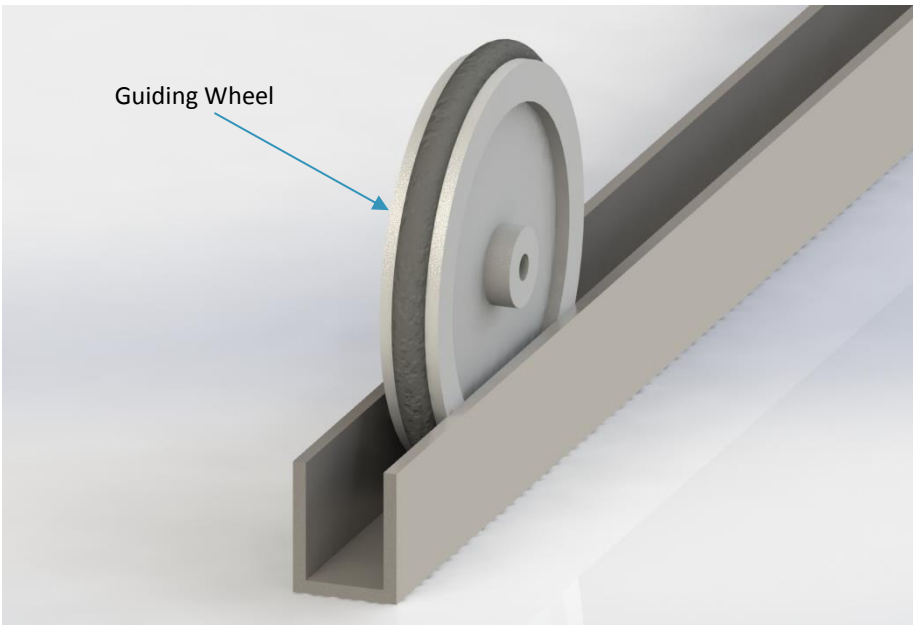


FIGURE 4 : LATERAL VIEW OF GUIDING WHEEL AND RAIL

2.3 Movement in y-direction

Along the y-direction, movement is done along a steel lead rod. The carriage and the lead rod system is shown in figure 5.

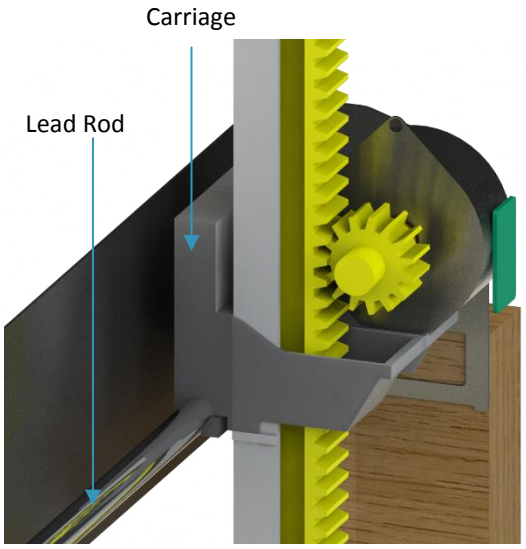


FIGURE 5 : LATERAL VIEW OF CARRIAGE AND LEAD ROD

The carriage is moved with the help of a DC brushed motor and the reference of position is kept using an optical encoder in the form of marked plastic tape. An IR (infrared) LED pair is used where when light from transmitter reflects across the tape and the receiver produces pulse according to the speed of movement. The Arduino Nano controller is used to identify the changes in pulse output of the photodiode, whenever the colour underneath it changes. This system is called a linear encoder. This provides the controller with the following parameters,

a) The speed of movement of carriage
b) The absolute (or relative) position of carriage on the leading rod.

2.4 Movement in z-direction

Rack and pinion mechanism is used for movement along z-direction. It is a type of linear actuator that converts rotational motion into linear motion. Here in figure 6, circular gear is called pinion which engages teeth on linear bar called rack. Rotational motion applied to pinion causes rack to move relative to pinion thereby translating rotational motion of pinion into linear motion, directed in the z-direction.

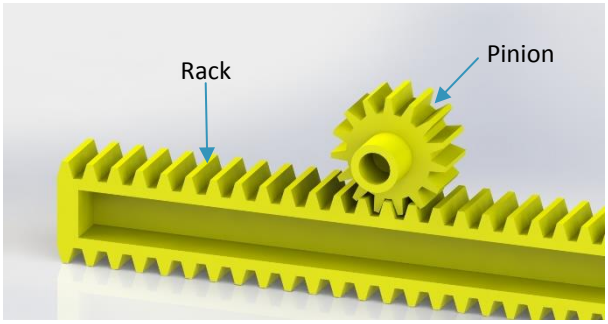


FIGURE 6 : RACK AND PINION MECHANISM

The pinion is guided along the z-direction using a PVC casing. The casing does not allow the movement of rack along any other axis, thus the rack has only one degree of freedom. The rack is supported by the casing from one side and the pinion from the other side.

2.5 Mechanism for sowing seeds

A container with seeds is attached to the boundary of the device. Different containers may be used, that can contain different types of seeds. A vacuum pump controlled by the raspberry pi is used to create suction through a nozzle. This nozzle is mounted on the rack. This nozzle acts as both a picker and dropper for the seeds. The vacuum pump creates suction that will hold a few seeds to the nozzle and drop them at the required spot.

The suction pump used is a rotary vane pump. This type of pump is very cheap and simple but the suction power may be limited.

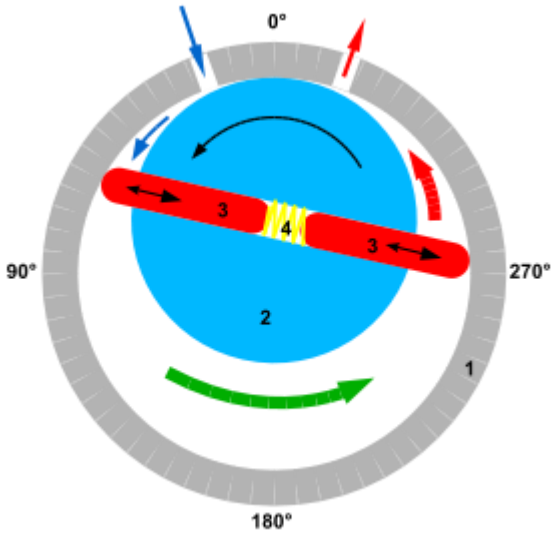


FIGURE 7 : ROTARY VANE PUMP

The Rotary Vane Pump was invented by Charles C. Barnes and works on the principle of creating eccentricity. Vanes are allowed to slide into and out of the rotor and seal on all edges, creating vane chambers that do the pumping work. On the intake side of the pump, the vane chambers are increasing in volume. These increasing-volume vane chambers are filled with fluid forced in by the inlet pressure. Inlet pressure is actually the pressure from the system being pumped, often just the atmosphere. On the discharge side of the pump, the vane chambers are decreasing in volume, forcing fluid out of the pump. The action of the vane drives out the same volume of fluid with each rotation. [4]

The pump is fitted with a nozzle and a flexible pipe so that it can move freely when attached to the rack end. The nozzle is provided to decrease the surface area and hence increase the pressure created due to suction.

2.6 Mechanism for Irrigation

An elevated water tank carries the water to be used for irrigation. The flow of water is controlled using a solenoid valve, which is controlled using the raspberry pi. The solenoid valve is a magnetic device having an inductive coil used as a solenoid to push and pull a plunger which acts as a valve. This device is commonly used to control the flow of fluids in hydraulic and pneumatic systems.

Figure 8 shows the construction of the solenoid valve.

The Solenoid valve has two electrical inputs pertaining to the two terminals of the coil. The valve is normally closed, i.e. no flow of water is permitted. When an electric current is passed through the coil the plunger is pulled inside the solenoid and releases the valve and water flows. The coil is rated to work at 24V and normally takes a constant current of about 130mA to switch.

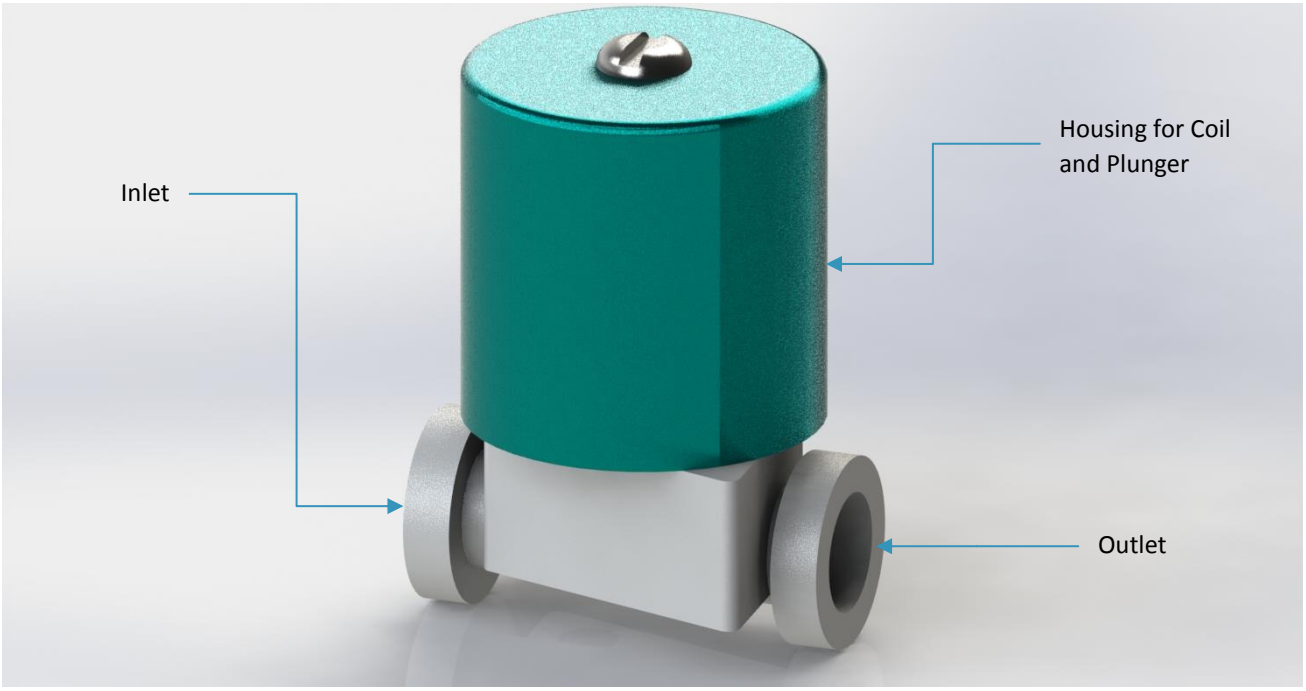


FIGURE 8 : SOLENOID VALVE

2.6.1 Drip Irrigation System

A drip irrigation system using the above-mentioned solenoid valve is implemented. In this, the Raspberry Pi retrieves the ambient temperature information using a web service or WebAPI. The received temperature information is stored for the particular instance. The time period for keeping the solenoid valve on is calculated using the temperature. Thus, the solenoid valve allows only a calculated amount of water to drip.

The idea behind this is that, with increase in ambient temperature, the rate of evaporation of water from soil also increases, thus more amount of water is needed for the drip, while in cooler or humid weather needs a lot less amount of water for the drip. This results in precise and calculated use of available water.

2.7 Electro-Mechanical Devices

The main electromechanical devices used are as follows:

1. Johnson Motor (Reduction gear included)
2. Brushed DC Motor (without reduction gear)
3. Bipolar Stepper Motor

2.7.1 Johnson Motor

The Johnson Motor is a high torque brushed DC Motor. It is manufactured by Johnson Electric. The Johnson motor used in our project is rated at 12V and produces a torque of 30kgcm. The Johnson motor comprises of two parts, the DC brushed motor assembly and the reduction gear assembly. The following figure shows the components of Johnson motor.

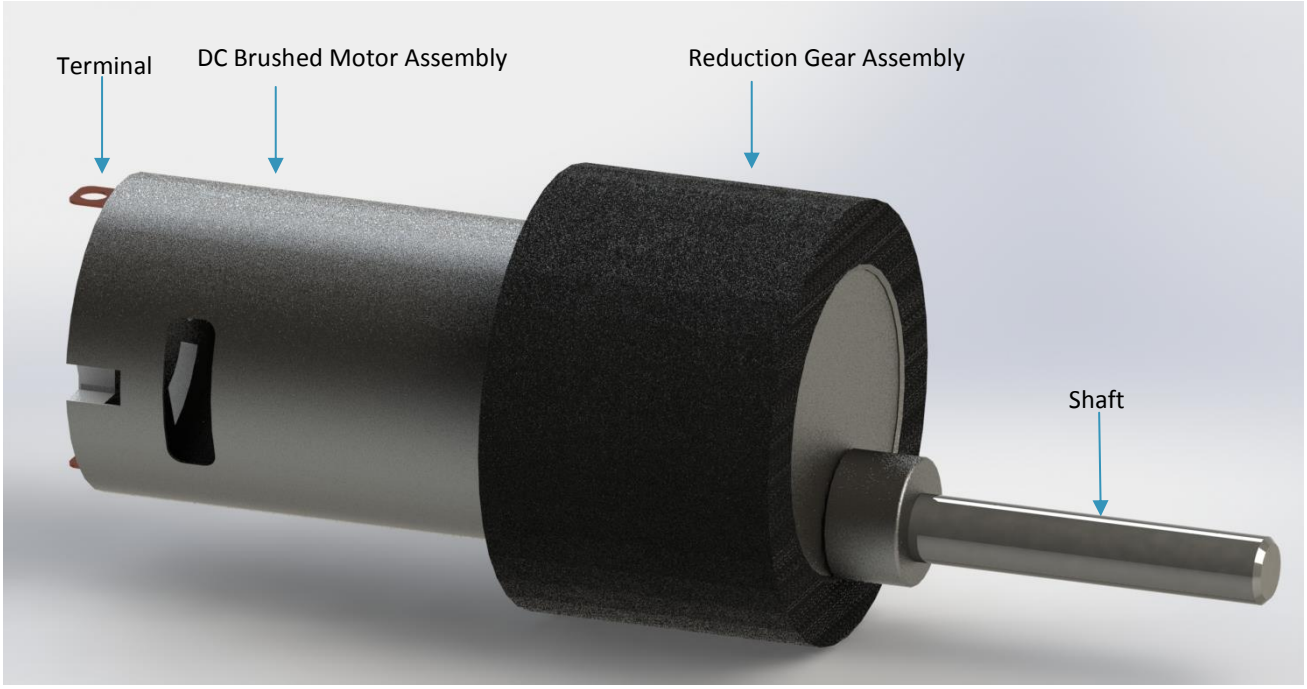


FIGURE 9 : JOHNSON MOTOR

The reduction gear assembly is provided to reduce the angular velocity of rotation and increase the torque. This is based on the principle of conservation of angular momentum.

A typical Johnson motor operating at 12V DC supply draws a current of around 300mA at unloaded state and about 1.2A in an appropriately loaded state.

The DC brushed motor assembly constitutes of a commutator, which acts as a mechanical rectifier. The function of the commutator is to maintain the direction of generated magnetic field, so that the torque is in one direction only. Without a commutator, the motor will not be able to rotate as completing each rotation will result in reversal of magnetic field. This will again force the motor to rotate in the opposite direction. It not only provides power to the coils but also reduces the friction. The commutator is generally made of carbon, but metallic commutators with gold plating are also available. The particular model used in our project uses the latter.

2.7.2 Brushed DC Motor

The brushed DC motor is a high RPM DC motor. The motor is rated at 12V and has an RPM of 1200 rotations per minute. It is important to note that there is no reduction gear assembly in a DC brushed motor, which is why the rpm is high. The figure below describes the DC brushed motor.

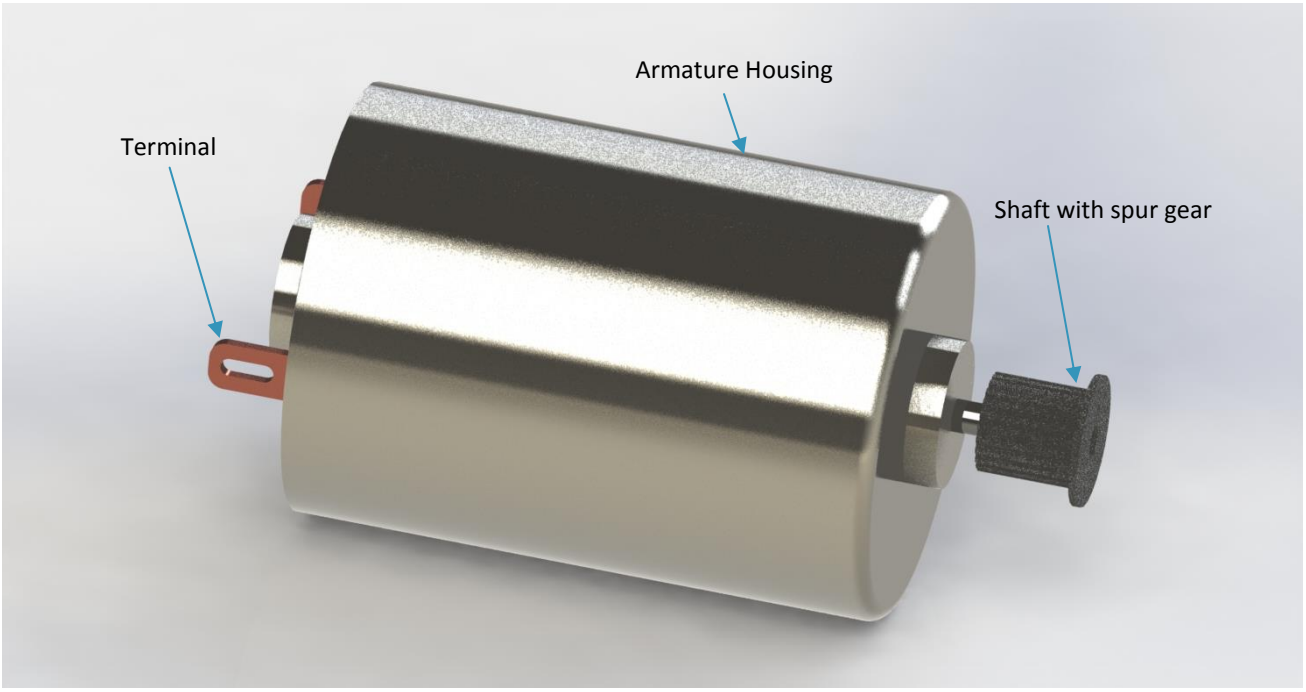


FIGURE 10 : DC BRUSHED MOTOR

The DC brushed motors working is same as the working of the DC brushed assembly of Johnson motor. It is also comprised of a commutator, but the motor is designed for light loads only.

2.7.2.1 Working of a brushed DC motor

In a typical DC motor, there are permanent magnets on the outside and a spinning armature on the inside. The permanent magnets are stationary, so they are called the stator. The armature rotates, so it is called the rotor.

The rotor contains an electromagnet. When an electric current is run in this electromagnet, it creates a magnetic field in the armature that attracts and repels the magnets in the stator. So, the armature spins through 180 degrees. To keep it spinning, the polarity has to be changed. The brushes handle this change in polarity. They make contact with two spinning

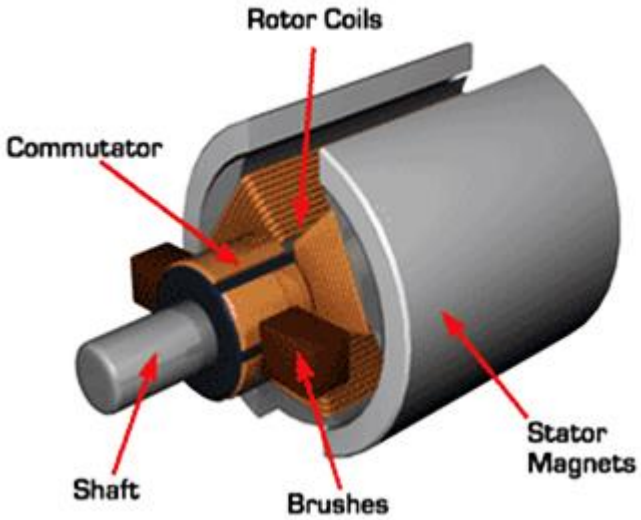


FIGURE 11 : INSIDE A DC BRUSHED MOTOR [7]

electrodes attached to the armature and flip the magnetic polarity of the electromagnet as it spins. [6]

2.7.3 Stepper Motor

Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called "phases". By energizing each phase in sequence, the motor will rotate, one step at a time. With a computer controlled stepping you can achieve very precise positioning and/or speed control. [5]

The stepper motor provides an open loop type control system which is good because of simplicity.

The advantages of stepper motor are as follows,

- Positioning – Since steppers move in precise repeatable steps, they excel in applications requiring precise positioning such as 3D printers, CNC, Camera platforms and X,Y Plotters. Some disk drives also use stepper motors to position the read/write head.
- Speed Control – Precise increments of movement also allow for excellent control of rotational speed for process automation and robotics.
- Low Speed Torque - Normal DC motors don't have very much torque at low speeds. A Stepper motor has maximum torque at low speeds, so they are a good choice for applications requiring low speed with high precision.

But stepper motors do have limitations,

- Low Efficiency – Unlike DC motors, stepper motor current consumption is independent of load. They draw the most current when they are doing no work at all. Because of this, they tend to run hot.
- Limited High Speed Torque - In general, stepper motors have less torque at high speeds than at low speeds. Some steppers are optimized for better high-speed performance, but they need to be paired with an appropriate driver to achieve that performance.
- No Feedback – Unlike servo motors, most steppers do not have integral feedback for position. Although great precision can be achieved running ‘open loop’. Limit switches or ‘home’ detectors are typically required for safety and/or to establish a reference position.

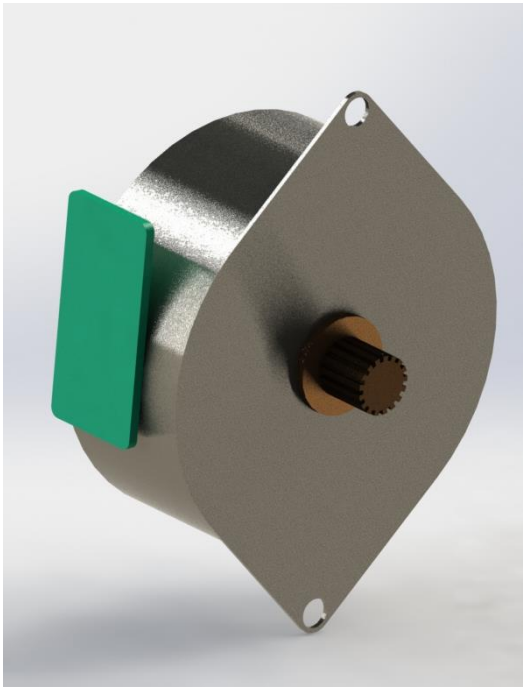


FIGURE 12 : BIPOLAR STEPPER MOTOR

There are various kinds of stepper motors available. The significant ones are,

- Bipolar Stepper Motors
- Unipolar Stepper Motors

Unipolar drivers, always energize the phases in the same way. One lead, the "common" lead, will always be negative. The other lead will always be positive. Unipolar drivers can be implemented with

simple transistor circuitry. The disadvantage is that there is less available torque because only half of the coils can be energized at a time.

Bipolar drivers use H-bridge circuitry to actually reverse the current flow through the phases. By energizing the phases with alternating the polarity, all the coils can be put to work turning the motor.

A two-phase bipolar motor has 2 groups of coils. A 4-phase unipolar motor has 4. A 2-phase bipolar motor will have 4 wires - 2 for each phase. Some motors come with flexible wiring that allows you to run the motor as either bipolar or unipolar.

In Farmbot, a bipolar stepper motor is used in moving the rack along the z-direction. A pinion is attached to the shaft of the stepper motor. The rack is bound to move along z-axis only. The stepper motor receives its control signals from a L298 motor driver.

The working of a bipolar stepper motor is explained below.

2.7.3.1 Working of Bipolar Stepper Motor

Bipolar permanent magnet [8] and hybrid motors are constructed with two windings around the armature. The four terminals of the windings are provided as is. To drive the stepper motor, we need to energize each coil in such a way that the armature rotates in a particular direction.

The motor itself is simpler but the drive circuitry needed to reverse the polarity of each pair of motor poles is more complex. The schematic in Figure 13 shows how such a motor is wired, while the motor cross section shown here is exactly the same as the cross section shown in Figure 13. The drive circuitry for such a motor requires an H-bridge control circuit for each winding. Briefly, an H-bridge allows the polarity of the power applied to each end of each winding to be controlled independently. The control sequences for single stepping such a motor is shown below, using + and - symbols to indicate the polarity of the power applied to each motor terminal:

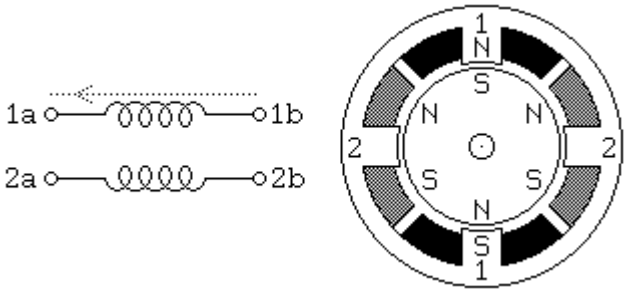


FIGURE 13 : WIRING IN A BIPOLAR STEPPER MOTOR [8]

Terminal 1a	+---+---+---+---
Terminal 1b	--+---+---+---+-
Terminal 2a	-+---+---+---+--
Terminal 2b	---+---+---+---+
	time ⟶

The control sequences are generated using the Arduino Nano board, using the Stepper.h library. Explanation for this is given in subsequent sections.

Chapter 3
Electronic Components

3.1 Power Management System

All components (except the vacuum pump) of the bot are powered by a single 500W ATX12 PSU (**Power Supply Unit**). It is capable of converting 240V, 50Hz mains supply into DC, which can be further utilized by various electronic components. The particular PSU we are using is capable of producing maximum output power of 500W and DC voltage in range 3.3V-12V. The current rating for the PSU is as follows:

Voltage	Maximum Current Output
3.3V	16 A
5V	18 A
12V	15 A

Apart from the above power rails, negative power rails of -12V and -5V are also available. These are called auxiliary rails and are generally provided for backward compatibility with older RS-232 systems, which required both negative and positive voltages.

Such an ATX12 PSU is generally used as a power source for computer motherboards and LED lighting systems. In the case of LED lighting systems constant current output is required, so a PSU is a good and optimized system for providing constant power.

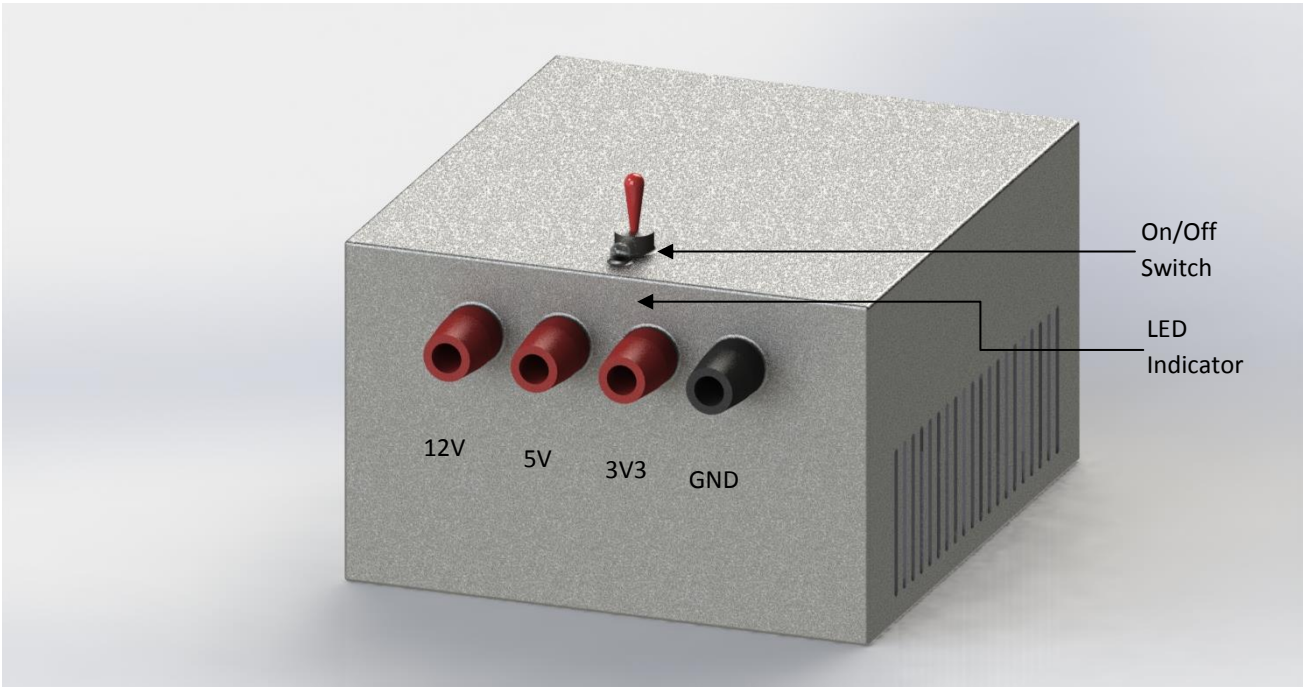


FIGURE 14 : POWER SUPPLY UNIT OF FARMBOT

The standard PSU comes with a 24-pin Molex connector, whose pinouts are shown as follows,

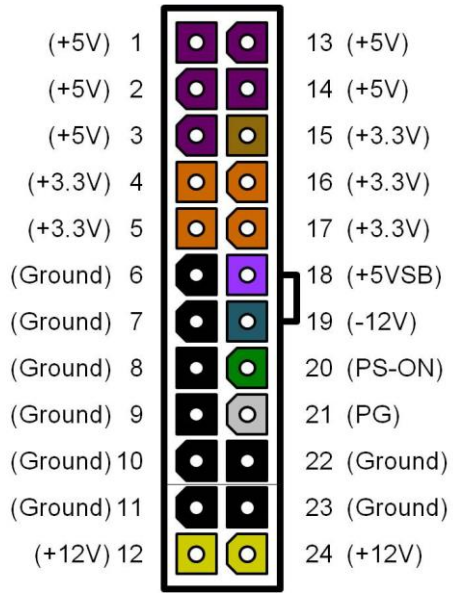


FIGURE 15 : PINOUT OF 24-PIN ATX12 PSU

Pin	Name	Description
1 **2** **3** **13** **14**	+5 V	Provides +5V supply (max 18A)
4 **5** **15** **16** **17**	+3.3V	Provides +3.3V supply (max 16A)
24	+12 V	Provides +12V supply (max 15A)
6 **7** **8** **9** **10** **11** **22** **23**	GND	Ground Terminals
18	+5Vsb	+5V Standby voltage This voltage is provided irrespective of the PSU state.
19	-12V	-12V backward compatible (300mA max)
20	PS-ON	Active Low pin for turning on the power supply. The PSU turns on when the PS-ON is pulled down to ground.
21	PG	Power Good or Power OK pin. This pin is turned on when the PSU output voltage is stable. The PSU takes 500ms on an average to reach a constant DC voltage.

The advantages of using an ATX12 PSU in this project were as follows,

a. Better maintainability – No need of providing different regulator ICs (like 7805, 7809, etc.) on the circuit board. The PSU itself provides all the required voltages.
b. Better Efficiency – The PSU used has an efficiency of around 75% while a linear regulator like 7805 has an efficiency of around 56%. This is because of switched mode operation, in which the output voltage is generated by rapidly charging an inductor or capacitor using a switching device like a MOSFET.
c. Very regulated voltage output – The PSU has inbuilt voltage controller that eliminates any spikes or irregularities in output voltage. Thus, after the PSU powers on and reaches the 'Power Good' state, a constant DC supply is provided.

Although, the PSU also has some disadvantages,

a. Expensive – The standard ATX12 PSU costs around 5 times more than a regular linear voltage regulator.
b. Bulky – The area and size of the PSU is larger as compared to the linear regulator.
c. Complex – The circuit is very complex as compared to the linear regulator.

3.1.1 Working of a simple Switched Mode PSU (Buck Converter)

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage (while stepping up current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor,

although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter). [10]

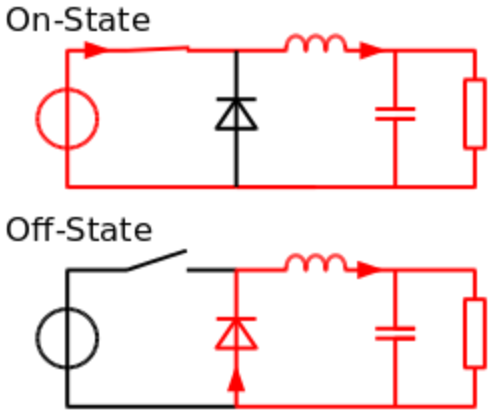


FIGURE 16 : THE TWO CIRCUIT CONFIGURATIONS OF A BUCK CONVERTER: ON-STATE, WHEN THE SWITCH IS CLOSED, AND OFF-STATE, WHEN THE SWITCH IS OPEN (ARROWS INDICATE CURRENT ACCORDING TO THE DIRECTION CONVENTIONAL CURRENT MODEL).

The basic operation of the buck converter has the current in an inductor controlled by two switches (usually a transistor and a diode). In the idealized converter, all the components are considered to be perfect. Specifically, the switch and the diode have zero voltage drop when on and zero current flow when off and the inductor has zero series resistance. Further, it is assumed that the input and output voltages do not change over the course of a cycle (this would imply the output capacitance as being infinite).

A buck converter operates in continuous mode if the current through the inductor (I_L) never falls to zero during the commutation cycle.

When the switch pictured above is closed (top of figure 16), the voltage across the inductor is ($V_L = V_i - V_o$). The current through the inductor rises linearly. As the diode is reverse-biased by the voltage source V, no current flows through it;

When the switch is opened (bottom of figure 16), the diode is forward biased. The voltage across the inductor is $V_L = -V_o$ (neglecting diode drop). Current I_L decreases.

A graph of output current voltage is shown in figure 17.

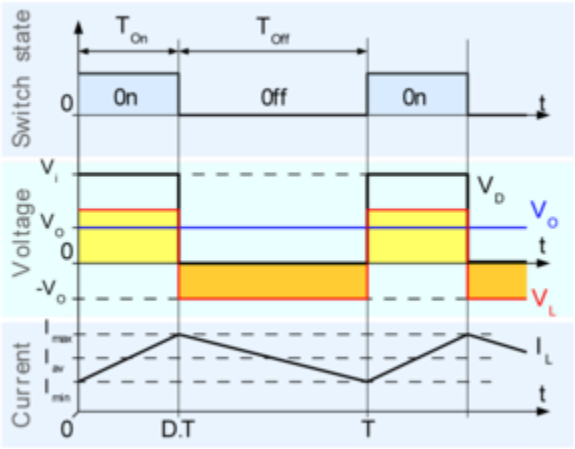


FIGURE 17 : OUTPUT CURRENT AND VOLTAGE OF BUCK CONVERTER

3.2 ULN2003A Darlington Array IC

The ULN2003A is an array of seven NPN Darlington Transistors capable of 500mA, 50V output. It features common-cathode fly-back diodes for switching inductive loads. In the same family are ULN2002A, ULN2004A, as well as ULQ2003A and ULQ2004A, designed for different logic input levels.

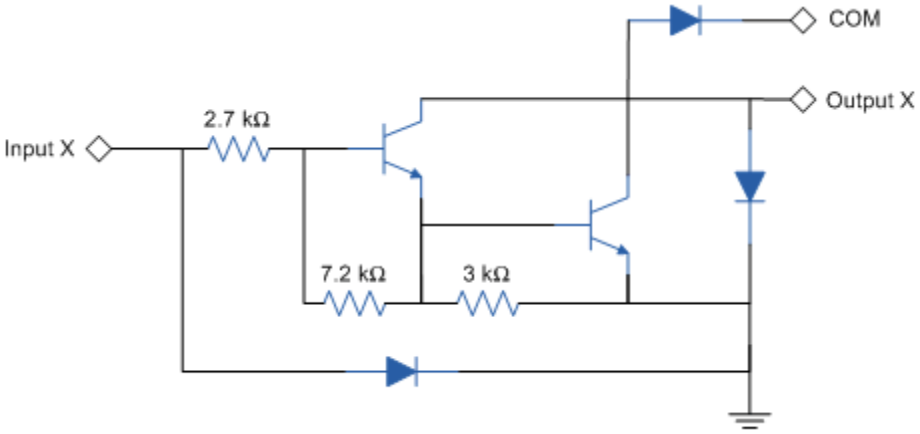


FIGURE 18 : INTERNAL CIRCUIT OF A SINGLE CHANNEL IN ULN2003A

Resultant gain of the Darlington: $\mathrm{Gain}_{(\mathrm{total})} = \mathrm{gain}_1 \times \mathrm{gain}_2$

The ULN2003 is known for its high-current, high-voltage capacity. The drivers can be paralleled for even higher current output. Even further, stacking one chip on top of another, both electrically and physically, has been done. Generally, it can also be used for interfacing with stepper motor, where the motor requires high ratings, which cannot be provided by other interfacing devices.

Main specifications:

- 500 mA rated collector current (single output)
- 50 V output (there is a version that supports 100 V output)
- Includes output fly-back diodes

Inputs are compatible with both TTL and 5-V CMOS logic.

The pinout of a standard ULN2003A IC is provided in figure 19. [11]

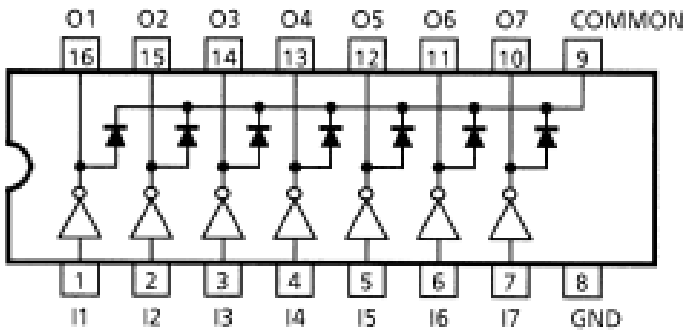


FIGURE 19 : PINOUT OF ULN2003A

3.3 L293D Motor Driver IC

L293D is a typical Motor Driver IC, which allows DC motor to drive on either direction. L293D is a 16-pin IC, which can control a set of two DC motors simultaneously in any direction.

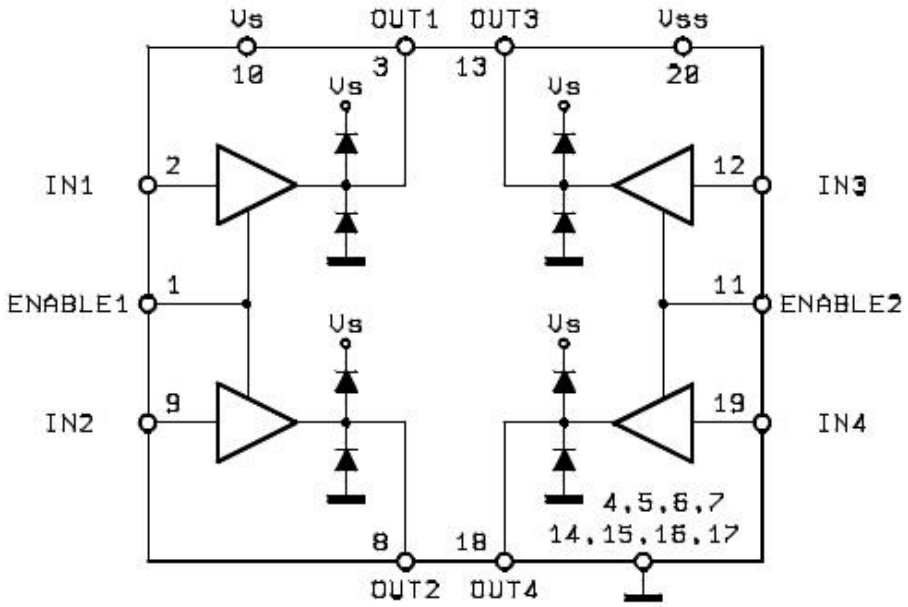


FIGURE 20 : INTERNAL CIRCUIT OF L293D

Various features of L293D are

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Output Current 600 mA Per Channel
- Peak Output Current 1.2 A Per Channel

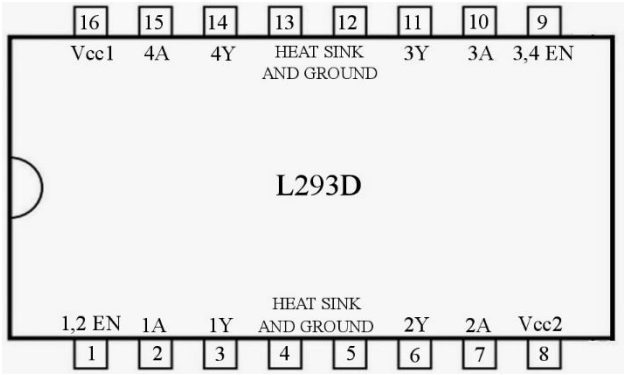


FIGURE 21 : PINOUT OF L293D

The pinout of standard L293D is shown in figure 21. Pin Descriptions are as follows:

1. **Vcc1**: Regulated 5V supply for powering the IC.
2. **Vcc2**: High voltage supply for driving the motor.
3. **EN:** Enable Pin. The IC has two enable pins for the two channels. These pins are active high, so providing a logic high will turn on the particular IC.
4. **A**: Logic input received from the microcontroller IC. TTL logic is supported by the IC.
5. **Y**: Output voltage level. The DC motor is directly connected to the Y pins.
6. **Ground**: The ground pins in L293D acts as both ground and heatsinks. The heatsinking is not implicit, i.e. not provided internally in the IC. The user has to make a solder glob linking the pins 4, 5, 12 and 13 for the heatsinking. The extra solder keeps the IC cool by dissipating extra heat away from it.

3.4 L298 Driver IC

The L298 is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver de- signed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the in- put signals.

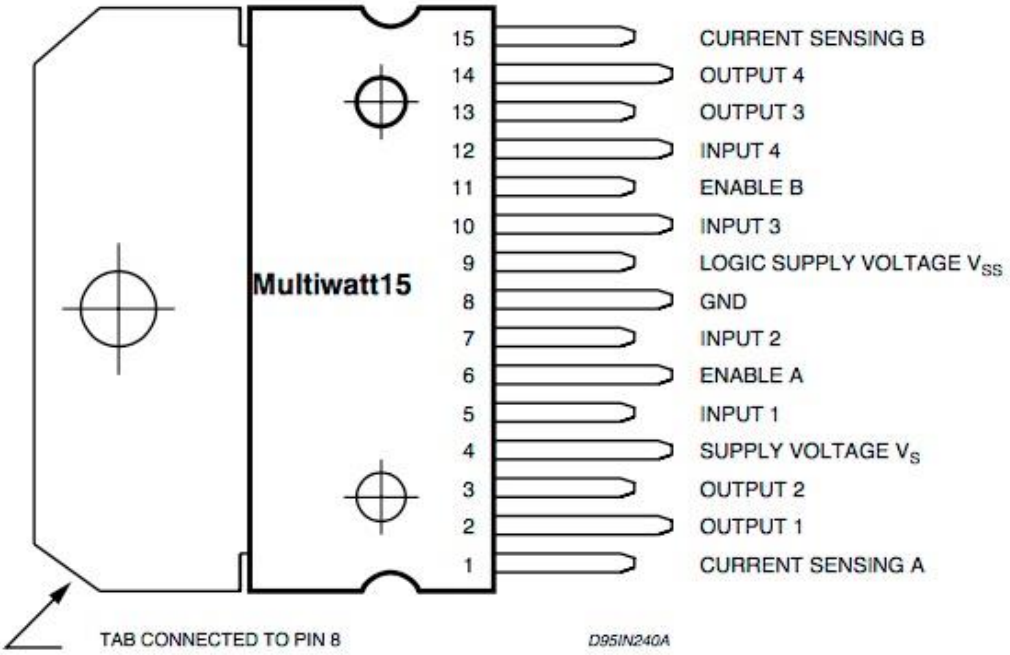


FIGURE 22 : L298 PIN DIAGRAM

- Driver: L298N
- Driver power supply: +5V~+46V
- Driver I_o: 2A
- Logic power output V_{ss}: +5~+7V (internal supply +5V)
- Logic current: 0~36mA
- Controlling level: Low -0.3V~1.5V, high: 2.3V~Vss
- Enable signal level: Low -0.3V~1.5V, high: 2.3V~Vss
- Max power: 25W (Temperature 75 degree Celsius)
- Working temperature: -25C~+130C
- Dimension: 60mm x 54mm
- Driver weight: ~48g
- Other extensions: current probe, controlling direction indicator, pull-up resister switch, and logic part power supply.

3.4.1 Advantages of using L298 over L293D

- L298 comes in a PowerSO22 package. This type of package is suitable for high current ICs. This IC runs a lot cooler than L293D
- Higher current output of the IC as compared to L293D.

The drawback of this IC is its unconventional pin profile (distance between pins), due to which it is difficult to place it in a standard PCB. Also, there are no built-in fly-back diodes.

3.5 7805 linear voltage regulator

7805 IC is the most common member of 78xx family and provides regulated 5V supply. Salient features of the 7805 IC are

- 3 terminal regulators
- Output current: max 1.5A
- Internal short-circuit current limiting

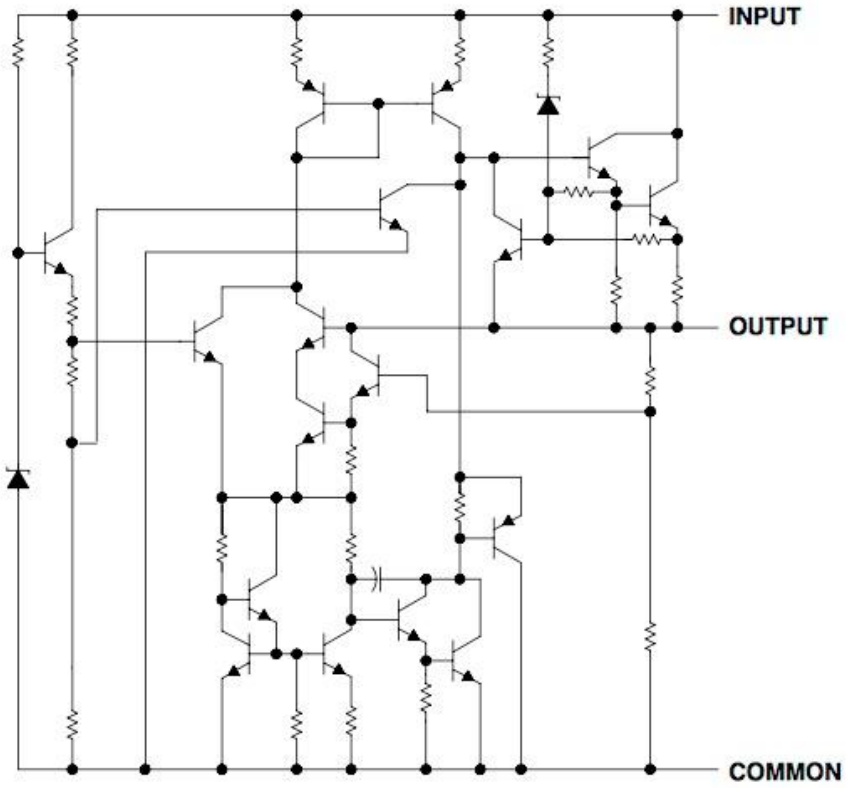


FIGURE 23 : INTERNAL CIRCUIT OF 7805

3.5.1 Advantages of using 7805 IC

a. These ICs do not require additional components to provide a constant, regulated source of power, making them easy to use, as well as economical and efficient uses of space. Other voltage regulators may require additional components to set the output voltage level, or to assist in the regulation process. Some other designs (such as a switched-mode power supply) may need substantial engineering expertise to implement.

b. These ICs have built-in protection against a circuit drawing too much current. They have protection against overheating and short-circuits, making them quite robust in most applications. In some cases, the current-limiting features of the 7805 devices can provide protection not only for the 7805 itself, but also for other parts of the circuit.

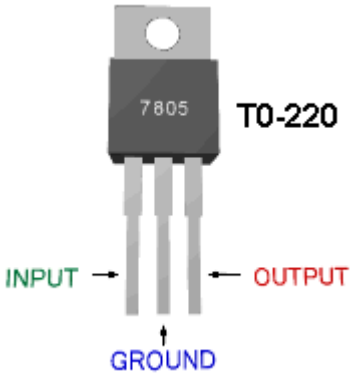


FIGURE 24 : 7805 IC PIN CONFIGURATION

3.5.2 Disadvantages

a. The input voltage must always be higher than the output voltage by some minimum amount (typically 2.5 volts). This can make these devices unsuitable for powering some devices from certain types of power sources (for example, powering a circuit that requires 5 volts using 6-volt batteries will not work using a 7805).
b. As they are based on a linear regulator design, the input current required is always the same as the output current. As the input voltage must always be higher than the output voltage, this means that the total power (voltage multiplied by current) going into the 7805 will be more than the output power provided. The difference is dissipated as heat. This means both that for some applications an adequate heatsink must be provided, and also that a (often substantial) portion of the input power is wasted during the process, rendering them less efficient than some other types of power supplies. When the input voltage is significantly higher than the regulated output voltage (for example, powering a 7805 using a 24-volt power source), this inefficiency can be a significant issue. Buck converters may be preferred over 7805 regulators because they are more efficient and do not require heat sinks, but they are more expensive.

3.6 SPDT Relay (Single Pole Double Throw)

A relay is an electrically operated switch. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The particular relay we are using, use an electromagnet for its switching action. Figure 25 shows a schematic of the SPDT relay.

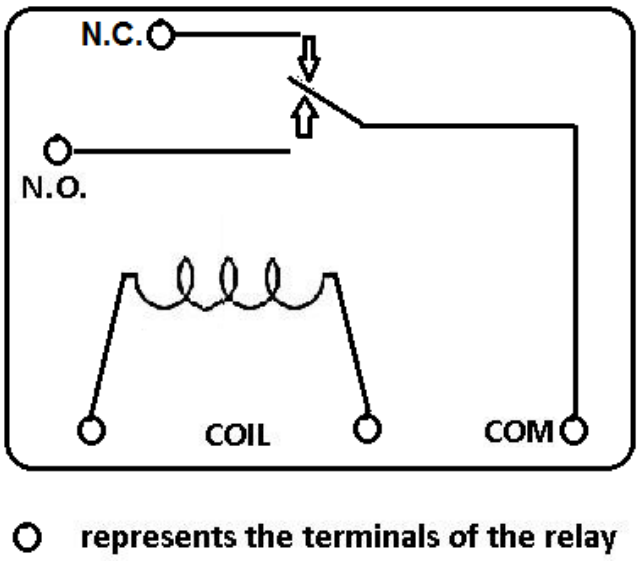


FIGURE 25 : SCHEMATIC OF SPDT RELAY

- Voltage: max 250V
- Current: max 10A
- Minimum driving voltage: 5V

3.7 IR Pair

The IR pair consists of IR LED designed to emit light in the IR region (950nm wavelength). The LED has a 1.5V of forward voltage drop and a max current capability of 50mA.
The photodiode or photo-transistor is used to detect this IR radiation emitted by the IR LED. This device is made to work in reverse bias. If no light is detected by the photodiode, then a very small reverse current known as 'Dark Current' flows through it. When the IR light falls on the photodiode, the thickness of the depletion region decreases and a current is established in the reverse bias.

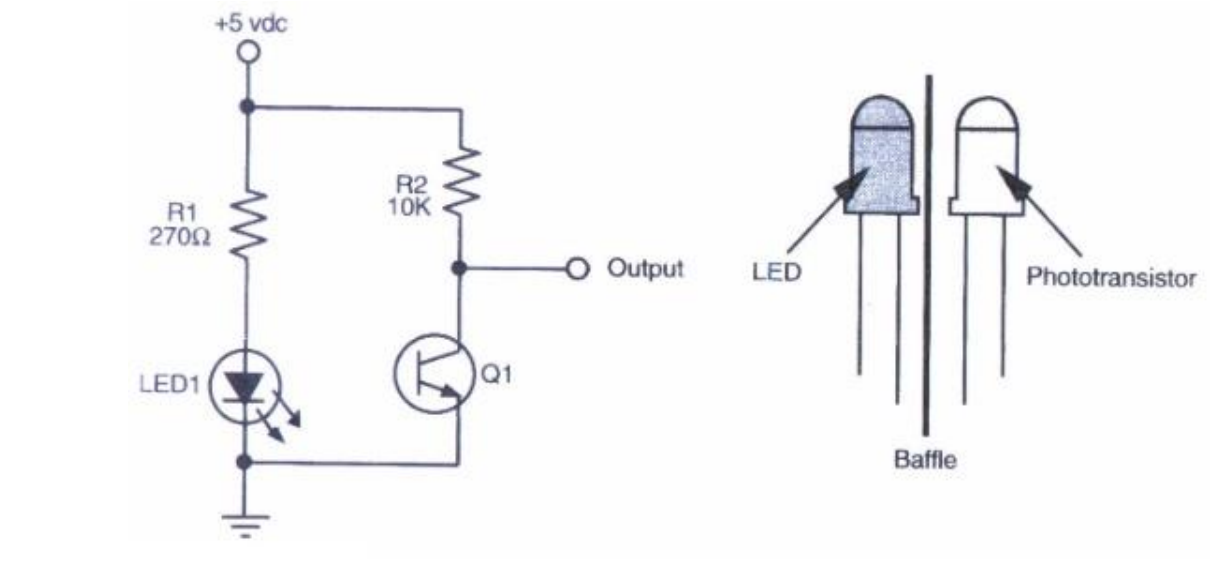


FIGURE 26 : IR LED PAIR CIRCUIT

Chapter 4
Circuit Diagrams

4.1 Interfacing Circuit for Motor and Drivers

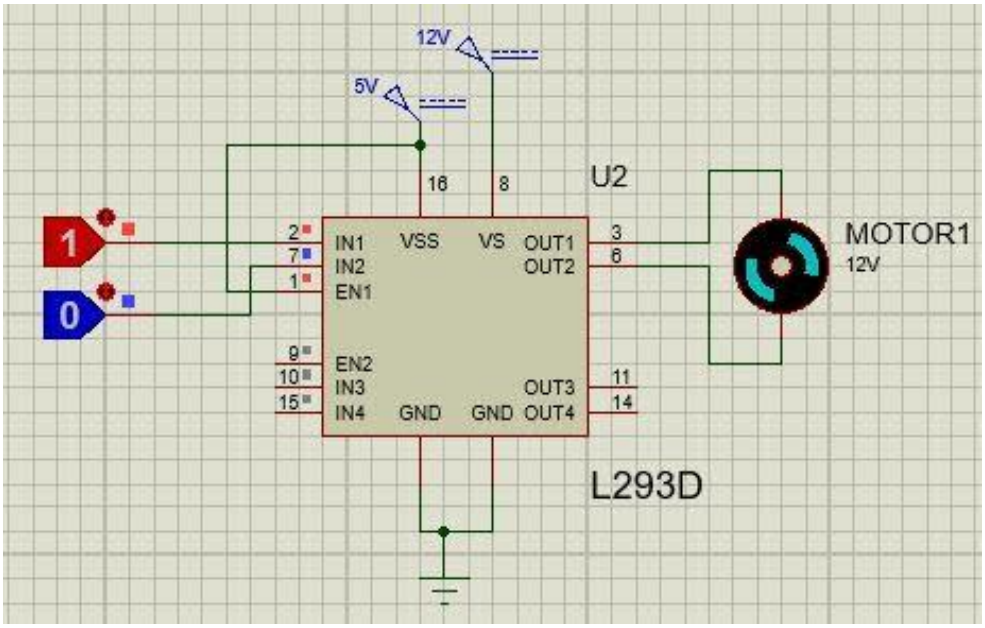


FIGURE 27 : INTERFACING L293D AND CARRIAGE MOTOR

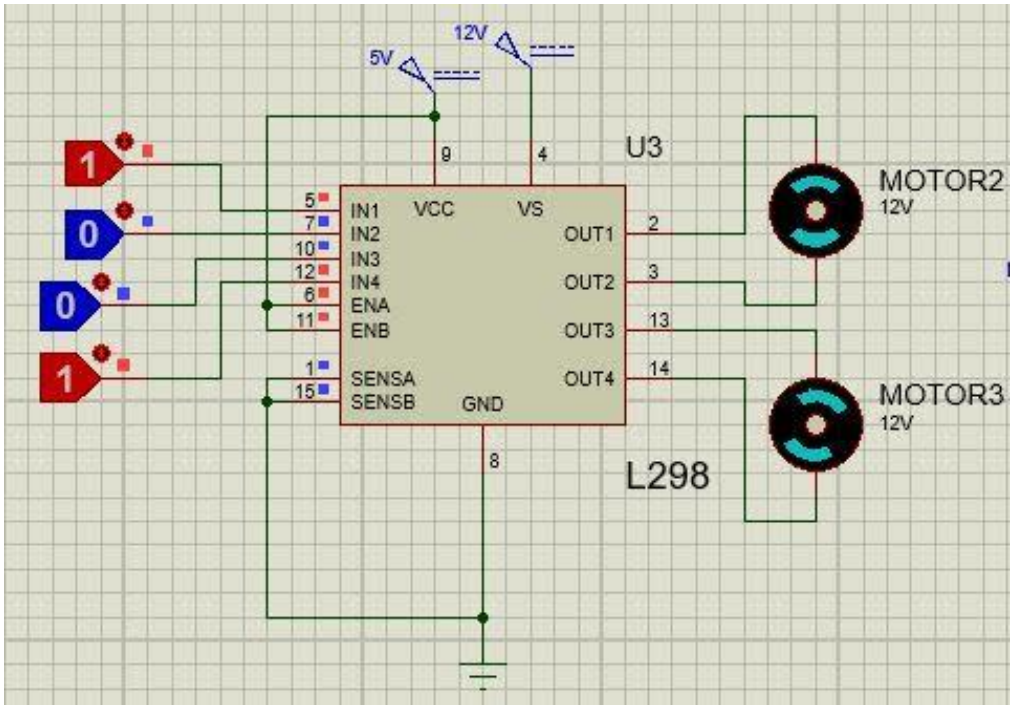


FIGURE 28 : INTERFACING L298 AND JOHNSON MOTOR

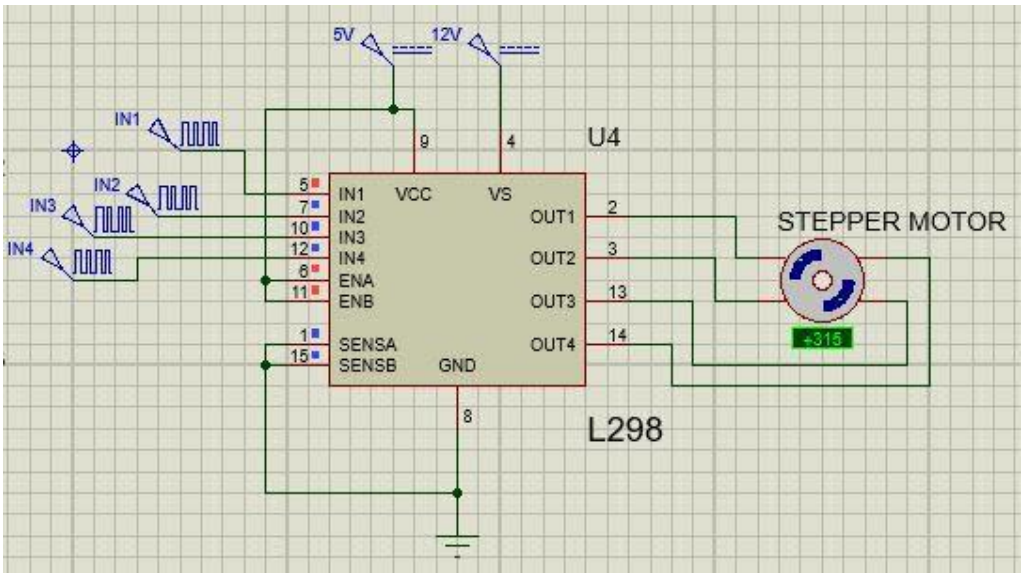


FIGURE 29 : INTERFACING L298 AND STEPPER MOTOR

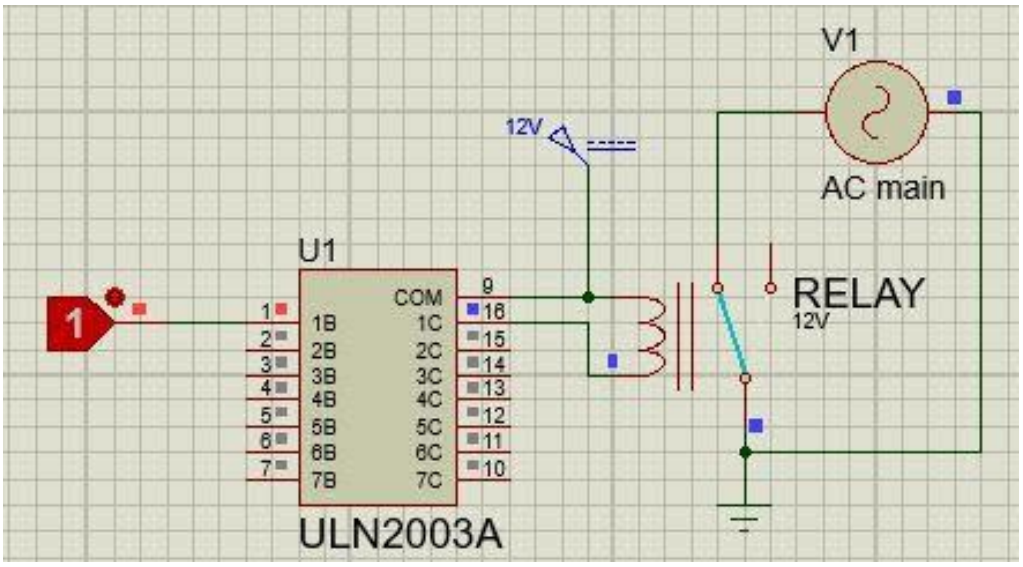


FIGURE 30 : INTERFACING ULN2003A WITH SPDT RELAY

4.2 Main Circuit Diagram

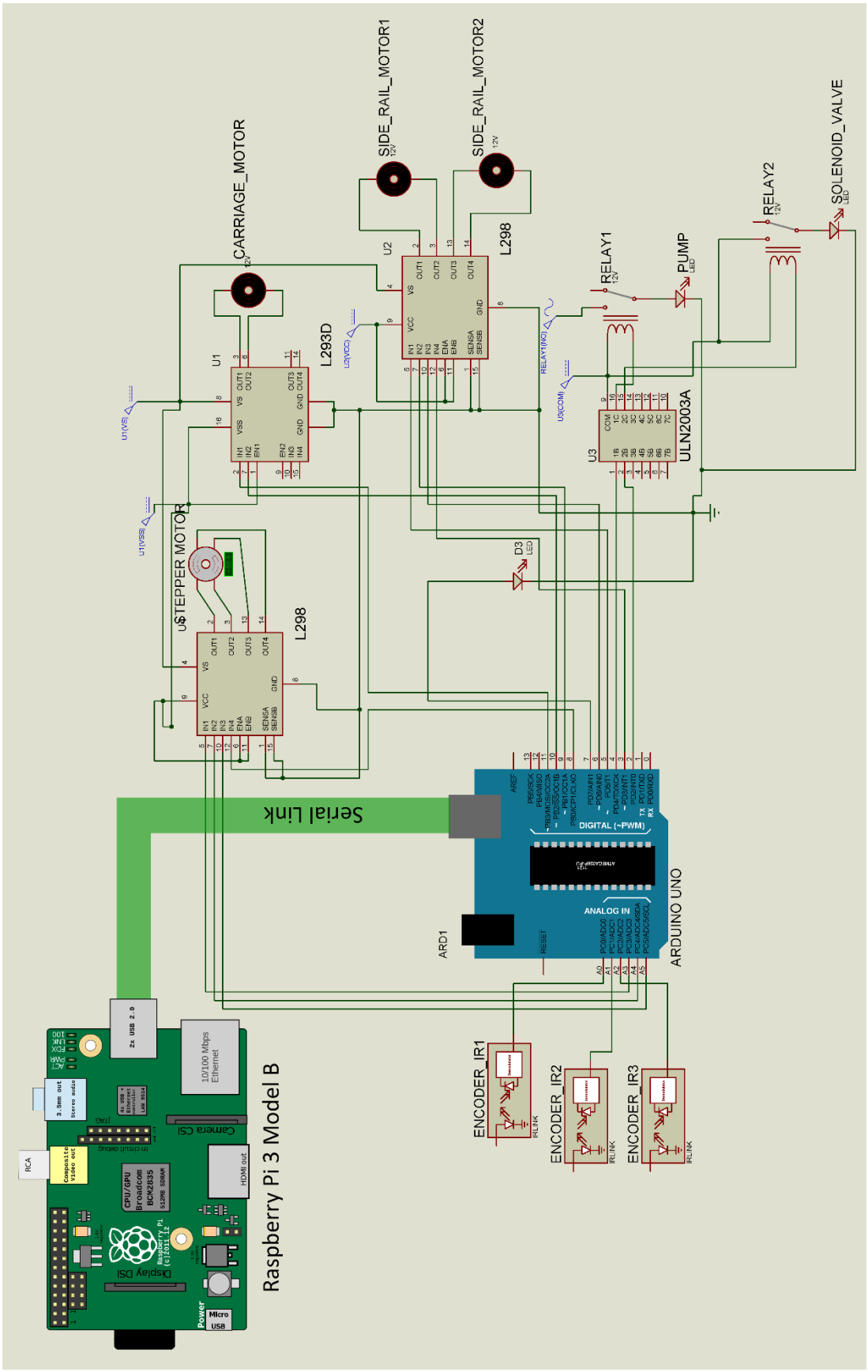


FIGURE 31 : MAIN CIRCUIT DIAGRAM FOR FARMBOT

Chapter 5
Control Systems

We have Raspberry Pi 3 Model B serving as core module for all our functionalities providing Arduino Nano instructions to perform tasks assigned by user and tasks needed accordingly. Raspberry Pi acts as brain to whole circuitry and serving as interface between user and bot. Arduino Nano is a microcontroller board which is further connected to other circuitry for movement of wheels along length and breadth of chassis. For movements along length and breadth i.e. x and y directions encoder is used to have the count of the movement. IR led pair is used for the counting and photo detector sends signal to Arduino Nano as it comes across the white spaces along the length of encoder.

Python 2.7 is used for coding in Raspberry Pi as Python is default language used for all hardware implementations in Raspberry Pi. Python is a high-level interpreting language.

A rectangular box is made where all seeds are kept. Through suction mechanism of pump seeds are sucked and sowed in soil. Valve is used for irrigation purposes. Solenoid valve is used as a switch which controls the flow of water through the pipe.

5.1 Raspberry Pi

The Raspberry Pi is a credit card-sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word processing, browsing the internet, and playing games. It also plays high-definition video.

Raspberry Pi3 Model B is used in the bot.

5.1.1 Technical Specifications:

- Broadcom BCM2837 Arm Cortex A53 Quad Core Processor powered Single Board Computer running at 900MHz
- 1GB RAM
- 40pin extended GPIO
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Micro USB power source.

FIGURE 32 : RASPBERRY PI 3 MODEL B BOARD

The Raspberry Pi performs the following functions:

1. The various operations performed by the Farmbot are all controlled by the Raspberry Pi. The control signals are then sent to the Arduino Uno serially which are then mapped with functions in C programming to control the hardware devices used in the Farmbot.
2. A camera is mounted on the gantry for observing the plant growth and to detect the presence of weeds. This is done with the help of computer vision and image processing. The camera is connected to the Raspberry Pi, which allows us to use OpenCV in python to perform the various computer vision algorithms.
3. With an internet connection, the system can automatically syndicate weather forecast information using several online APIs, and accordingly decide the amount of water to use for irrigation. For example, if rains are expected in near future, the system can use lesser amount of water.

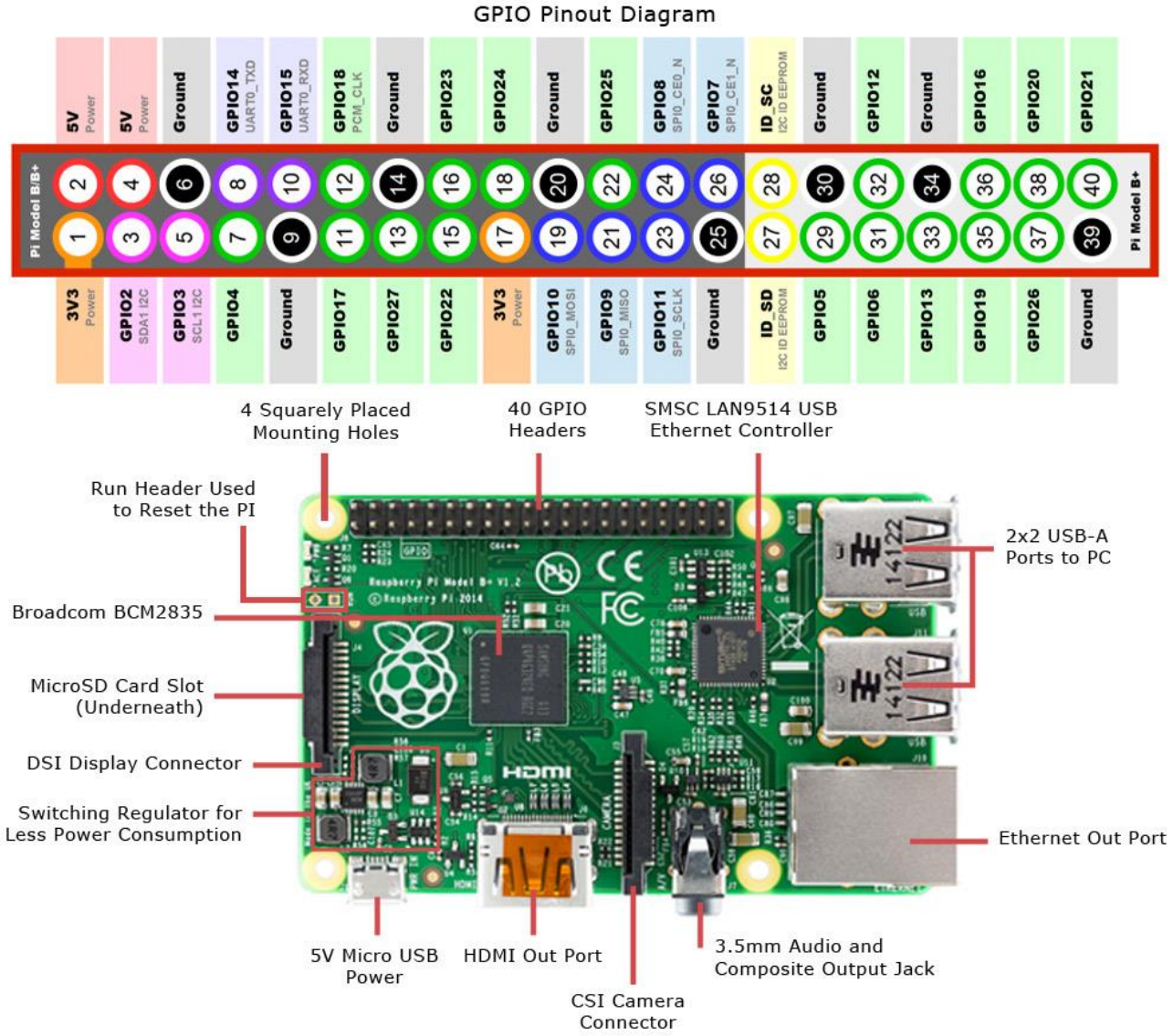


FIGURE 33 : SCHEMATIC DIAGRAM OF RASPBERRY PI

5.2 Arduino Nano

The Arduino Nano [12] is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.

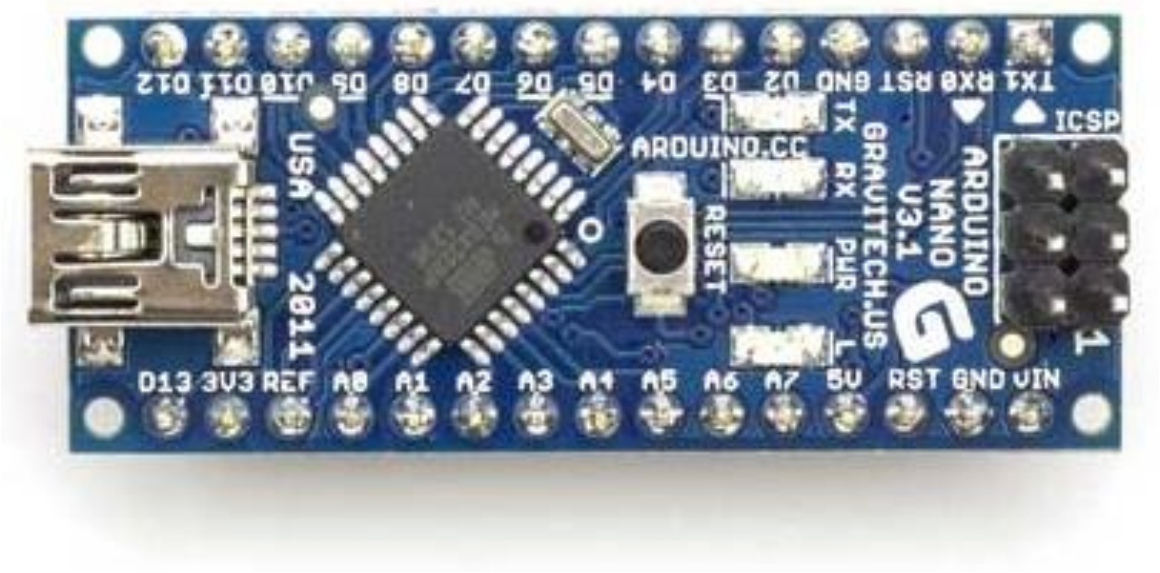


FIGURE 34 : ARDUINO NANO BOARD

5.2.1 Technical Specifications

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"
Length	45 mm
Width	18 mm
Weigth	5 g

5.2.1.1 Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

5.2.1.2 Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

5.2.1.3 Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega168 ports.

5.2.1.4 Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Nano's digital pins.

The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet.

5.2.1.5 Programming

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

5.2.1.6 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nano-farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

5.2.2 Role in Farmbot

The Arduino here, is acting as a priority queue for performing the operations, as the system can do only one work at a time. The Raspberry pi acts as a feed for control commands, which the Arduino interprets and stores in a queue arranged in the order of their priority. The commands are purged once they are executed.

The Arduino is also responsible for controlling the motors and actuators. This is done using the principles of Numeric Control, widely used in applications like printers and CNC machines. A feedback mechanism composed of optical encoders is used to keep track of the movement of the gantry and the cross slide/carriage. The optical encoders employ an IR LED pair which are made to pass over a marked tape, with alternate black and white (clear) markings. The movement generates a pulse output across the photodetector, from which the relative displacement can be calculated by counting the number of negative edges starting from a certain time instant.

5.3 Feedback Systems

5.3.1 Closed Loop Feedback Process:

In this type of process the output of previous stage is also fed to input to give the output of next stage. In this a feedback mechanism is there which keeps track of the output.

5.3.2 Open Loop Mechanism:

In this there is no feedback mechanism and input of next stage is independent of output of previous stage.

5.4 Optical Encoders

We have used two types of optical encoder mechanism to keep track of movement of gantry and movement of carriage.

5.4.1 Linear Encoder

A linear encoder is a sensor, transducer or read head paired with a scale that encodes position. The sensor reads the scale in order to convert the encoded position into an analog or digital signal, which can then be decoded into position by a digital readout (DRO) or motion controller.

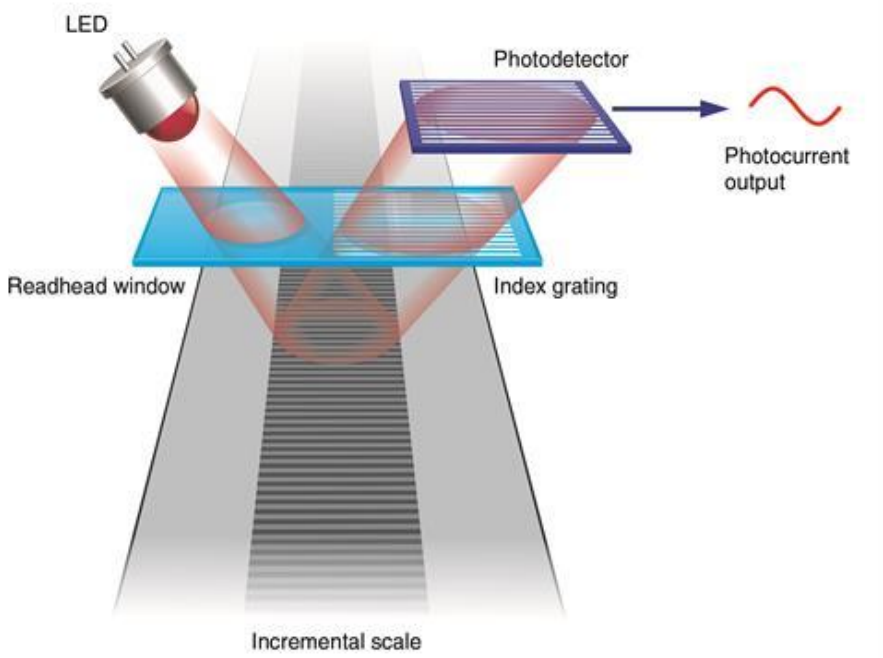


FIGURE 35 : LINEAR ENCODER SYSTEM USED IN PRINTERS

5.4.2 Rotary Encoder:

A rotary encoder, also called a shaft encoder, is an electro-mechanical device that converts the angular position or motion of a shaft or axle to an analog or digital code. There are two main types: absolute and incremental (relative).

FIGURE 36 : A TYPICAL LINEAR ENCODER

For tracking the motion along x-axis (Gantry) two rotary encoder are built on both sides. Incremental type of encoder is used. The output of incremental encoders provides information about the motion of the shaft, which is typically further processed elsewhere into information such as speed, distance

and position. Photodiodes are used as optical encoders which signals Arduino about the motion of gantry. Johnson motors are used for movement of gantry in x-direction.

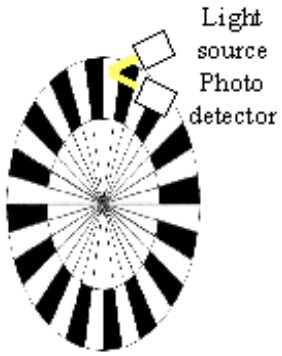


FIGURE 37 : ROTARY ENCODER SYSTEM

For tracking the motion along Y-axis (Carriage) an optical linear encoder is used. A rectangular area composed of alternating white and black marks is used as scale to keep track of motion of carriage in Y-axis. A DC motor is used for movement of carriage inside the gantry structure.

For movement, along x and y directions closed-loop feedback mechanism is used.

Motion along z-direction is controlled by rack and pinion method.

Chapter 6
Code for Raspberry Pi

```
1  import time,serial
2  import forecastio,datetime
3
4  api_key = "ccac2134fe3fe1d21eddc53d1a5fe85b"
5  lat=30.7333
6  lng=76.7794
7  current_time=datetime.datetime(2016,11,19,2,23,37)
8
9  forecast = forecastio.load_forecast(api_key, lat,lng,time=current_time)
10
11 # x=length of x direction
12 # y=length of y direction
13 # z=length of z direction
14 # rack_pitch= distance move by rack in one rotation
15 # x_div = no of divisions on x-axis
16 # y_div = no of divisions on y-axis
17 # c_x=no. of rows to be moved
18 # c_y=no. of columns to be moved
19 # c_z=no. of revolutions in z-direction
20 # 2000ms=4mm
21 #function=pump, import=currently
22 ##function==>position,pump,valve
23
24 fahr=(((forecast.currently().temperature)*9)/5)+32
25 fr=fahr
26
27 x=int(raw_input("Length in x-direction is:"))
28 y=int(raw_input("Length in y-direction is:"))
29 z=int(raw_input("Length in z-direction is:"))
30 rack_pitch=4
31
32 x_div=int(raw_input("No. of divisons in x-axis is:"))
33 y_div=int(raw_input("No. of divisions in y-axis is:"))
34 c_z=(int(z/rack_pitch))*10+25
```

The above lines are first 34 lines of code.

Time, Serial, Forecastio, Datetime are the packages which are imported using import function from python2.7 libraries for the proper functioning and efficiency of code.

Forecastio module provides information about weather and climate importing live information from weather forecasting website darksky.net for using this module we provide latitude, longitude information along with current_time which is an optional parameter. Then using API key we can access information.

Comments are used to describe the different variables and functions used in the code.

Variables x, y, z are global variables which define lengths along x, y and z directions respectively. Rack pitch is a fixed parameter and already defined by programmer. The spacing between seeds that is the number of divisions in x and y directions are also defined by user.

```
ser = serial.Serial(
    port="COM4",
    baudrate=9600,
    parity=serial.PARITY_ODD,
    stopbits=serial.STOPBITS_TWO,
    bytesize=serial.EIGHTBITS
)

ser.write("Z"+c_z)
```

This part defines the serial communication between arduino and raspberry pi. In this we define various parameters such as port number, baud rate, parity etc. needed to establish serial communication.

```
def position(seed):
        c_y=int(seed%x_div)
        c_x=int(seed/x_div)+1
        ser.write("X"+c_x+"\n")
        ser.write("Y"+c_y+"\n")
```

This function provides Arduino the co-ordinates of the seed when user has provided the where the operation is needed to be performed for instance, where the seed is to be placed.

```
def pump(pp):
    if(pp=="ON"):
        ser.write("A"+1+"\n")

    if(pp=="OFF"):
        return ser.write("A"+0+"\n")
```

This function signals Arduino to turn on and off the pump for suction.

```
def valve(fhr):
    if(fhr>20 and fhr<79):
        return ser.write("V"+4000+"\n")
    elif(fhr>79 and fhr<89):
        return ser.write("V"+6000+"\n")
    elif(fhr>90):
        return ser.write("V"+8000+"\n")
```

This function transmits Arduino to turn on/off valve for certain time period defined in milliseconds according to temperature and humidity of place.

```
def b2b():
    ser.write("X"+0+"\n")
    ser.write("Y"+0+"\n")
    ser.write("Z"+0+"\n")

def gantryLights(gL):
    if(gL=="ON"):
        ser.write("L"+1+"\n")
    if(gL=="OFF"):
        ser.write("L"+0+"\n")
```

The function b2b is used to transmit the co-ordinates of initial location to Arduino.

Function gantryLights is used to signal Arduino to turn on/off the lights on the gantry.

```
if __name__=="__main__":
    while(1):
        sed=int(raw_input("Enter the place where the seed is to be placed:"))
        pp=raw_input("you want pump to be ON/OFF:")
        gL=raw_input("you want lights to be ON/OFF:")
        pump()
        postion(sed)
        valve(fr)
        sleep(5)
        b2b()
```

If the python interpreter is running that module (the source file) as the main program, it sets the special `__name__` variable to have a value `__main__`. If this file is being imported from another module, `__name__` will be set to the module's name. This is the main part of the code. In this we are taking input from user.

Chapter 7
Future Scope and Improvements

7.1 Scope

1) **Research**: Farmbot can be used in labs for precise growing experiments in fields of Botany or Plant Biology. The above-mentioned schematic can be easily installed in labs and simplifies the process of analyzing the growth.
2) **For Food Sovereignty**: Farmbot can be used in the Backyard, on the rooftop or in greenhouses for organic farming and lets the user decide as to what is to be grown. It gives people the option for a healthy lifestyle.
3) **Commercially**: Farmbot can be used for small commercial businesses to save labor cost and improve efficiency. The project can be expanded to a larger scale with minimal additional cost.

7.2 Improvements

1) Real time feed of parameters of soil such as humidity, temperature of soil, its perception levels etc. to be provided to the bot for efficient decision making based on comparison with previous statistics as received from previous stored data on cloud.
2) Making the bot mobile (connecting wheels to it) so to cover wide area, wide field. The bot can then cover whole field on its own.
3) Weeds are major obstacles in the substantial growth of plants so to overcome this we can make a mechanism to remove weeds from the field.
4) Quality of seeds play an important role in growth as well as harvest of plant. Through image processing we can test the quality of seeds based on comparison between high yield seeds i.e. ideal seeds with the one which we are using.
5) Predictability of growth of seeds according to season and which seed will have a higher yield and thus farmer can make a profit. This can be done from regular monitoring of field data and weather report.
6) To make bot truly autonomous by monitoring and observing the previous inputs given by user i.e. using machine learning.

Chapter 8
Shortcomings and Result

The following parts of the robot's motion and functioning were tested:

- Proper motion along all the axis.
- Proper working of the suction pump system, in picking and dropping seeds.
- Proper working of the Solenoid Valve.
- Proper flow of control between Arduino Nano and Raspberry Pi.

The methodology used for testing is as follows,

1. The setup was manually moved to one end.
2. The robot was powered on and allowed to boot completely.
3. A manual command was given to move the control to a particular location.
4. After this, the robot was made to function in the automatic mode also.
5. The robot was made to retrieve the ambient temperature from a web service and the drip amount for irrigation was checked.

The following were the observations that were made during testing,

- The movement was proper along all the axis of motion.
- Raspberry Pi was able to properly communicate with the Arduino Nano controller.
- The closed loop feedback system parameters were adequate in moving the robot control to the specified spot.
- The amount of water drip was consistent with the temperature data received from the web service.

In spite of obtaining the above observations, there were some shortcomings in the design that we intend to work on, in the near future.

- The gantry system is inadequate for large sizes.
- The depth of the soil bed is also inadequate for growing many crops with extended root systems.
- The design is only able to plant those crops which rely on seeds. Crops like rice cannot be planted in this design.

We, conclude that such a robot or automation system might be able to increase the efficiency in agriculture to meet the needs of the future, if the suggested shortcomings were overcome.

References

1. Farmbot Genesis 1.0 Whitepaper at https://farmbot.io/whitepaper/
2. Agricultural Robotics and Automation http://www.ieee-ras.org/agricultural-robotics
3. Cartesian Coordinate Manipulator https://en.wikipedia.org/wiki/Cartesian_coordinate_robot
4. Rotary Vane Pump https://en.wikipedia.org/wiki/Rotary_vane_pump
5. Stepper Motors https://learn.adafruit.com/all-about-stepper-motors/what-is-a-stepper-motor
6. How does a brushless electric motor work? http://electronics.howstuffworks.com/brushless-motor.htm
7. http://www.zgcmotor.com/upload/Image/20150521/20150521150754_33455.jpg
8. Introduction to Stepper Motors http://mechatronics.mech.northwestern.edu/design_ref/actuators/stepper_intro.html
9. Inside ATX12 Power Supply Unit http://www.electronics-tutorials.ws/blog/convert-atx-psu-to-bench-supply.html
10. Buck Converter https://en.wikipedia.org/wiki/Buck_converter
11. ULN 2003A circuit diagramhttp://codeandlife.com/wp-content/uploads/2012/02/uln.png
12. Arduino Nano Technical Specifications https://www.arduino.cc/en/Main/ArduinoBoardNano