1. **INTRODUCTION**

Agribot is a farming management concept using modern technology to increase the quantity and quality of agricultural products. Farmers in the 21st century have access to GPS, soil scanning, data management, and Internet of Things technologies. The goal of smart agricultureresearch is to ground a decision-making support system for farm management. Smart farming deems it necessary to address the issues of population growth, climate change and labor that has gained a lot of technological attention, from planting and watering of crops to health and harvesting.

In IOT based smart agriculture, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, etc.) and automating the irrigation system. IOT (Internet of things) in an agricultural context refers to the use of sensors, cameras and other devices to turn every element and action involved in farming into data. There is a need smart agriculture to expand and develop from what it currently is because this practice will substantially decrease the negative environmental externalities of modern agriculture. Smart cities use Internet of Things (IOT) devices such as connected sensors, lights, and meters to collect andanalyze data. The cities then use this data to improve infrastructure, public utilities and services and more. For Farmers, it is difficult for them to understand technical terms and usage of technology and also itis a cost-effective affair.

According to recent statistics, crop cultivation land in India is shrinking at an unprecedented rate. The primary causes of inconsistent growth and development involve traditional irrigation methods and a lack of water supplies. Moreover, solar-powered agricultural automated process technology developments have the ability to provide major environmental advantages in India. Agribot is an approach to farming management that employs modern technology in order to improve both the quantity and the quality of agricultural products. Farmers are becoming increasingly popular in harvesting and selecting machines, but there are numerous other creative ways in which the farmers utilise automated technology to increase yields.

The demand for food rises above the available agricultural land, and farmers must fill the void. Agriculture robots assist them in doing so. Smart farming research aims to establish a decision-making framework for the operation of farms.In IOT-based smart farming, a machine is built to monitor the crop field and automate the irrigation system using sensors (light, humidity, soil moisture, temperature, and so on). In the agricultural context, IOT (which stands for Internet of Things) describes the use of cameras, sensors, and other devices to convert each component and procedure involved in growing crops into data. We need smart farming to grow and improve beyond where it is now because it will significantly reduce the negative impact on the environment of modern agriculture. Smart towns and cities collect and analyse data using Internet of Things (also known as IOT) devices like sensors that are connected, lights, and metres. Cities subsequently utilise this data to enhance infrastructure, public utilities, and services, among other things. Farmers find it difficult to know technical terms and how to use technology, and it is also a cost-effective endeavor.

## PROBLEM STATEMENT

To provide efficient decision support system using wireless sensor network which handle different activities of farm and gives useful information related to farm. Information related to Soil moisture, Temperature and Humidity content. Due to the weather condition, water level increasing. Farmers get lot of distractions which is not good for Agriculture. Water level is managed by farmers in both Automatic/Manual using that mobile application. It will make more comfortable to farmers. Performing agriculture is very much time consuming.

## OBJECTIVES

* + - To check the moisture content and taking necessary actions.
    - To plough the soil and cut the unwanted grass.
    - To plant the seeds by measuring equi-distance.
    - To send message to the farmers about the actions taken.

## ABOUT PROJECT

Agribot is an agriculture based robotic vehicle which is an automated smart agriculture system which reduces the time and resources that is required while performing it manually.

## LITERATURE SURVEY

* + - **Smart Agriculture: IOT based smart sensors agriculture by Anand Nayyar and Er. Vikram Puri, November 2016.** This paper describes Internet of Things (IOT) technologyhas brought revolution to each and every field of common man’s life bymaking everything smart and intelligent. IOT refers to a network of things which make a self-configuring network. The development of Intelligent Smart Farming IOT based devices is day by day turning the face of agriculture production by not only enhancing it but also making it cost-effective and reducing wastage. The aim / objective of this paperis to propose a Novel Smart IOT based Agriculture assisting farmers in getting Live Data (Temperature, Soil Moisture) for efficient environment monitoring which will enable them to do smart farming and increase their overall yield and quality of products.
    - **Brief Introduction of Paper:** This paper brings insights to construct a framework for robust working on fields and easy for farmers. One of main areas where IOT based research is going on and new products are launching on everyday basis to make the activities smarter and efficient towards better production is “Agriculture”. Agriculture sector is regarded as the more crucial sector globally for ensuring food security. Talking of Indiafarmers, which are right now in huge trouble and are at disadvantageous position in terms of farm size, technology, trade, government policies, climate conditions etc. conference for the name of the paper. In this newly created file, highlight all of the contents and import the prepared text file.
    - **Smart Agriculture Monitoring System Using IOT Dr. Sanjay N. Patil, Madhuri B. Jadhav:** Agriculture is basic source of livelihood People in India. Traditional Farm land irrigation techniques require manual intervention. Project includes sensors such as temperature, humidity, soil moisture and rain detector for collection the field data andprocessed. These sensors are combined with well-established web technology in the form of wireless sensor network to remotely control and monitor data from the sensors.
    - **Project Report on IoT based SMART FARMING SYSTEM CERTIFICATE OFAPPROVAL Counter signed by Yasir Farhir** (IoT) IoT refers to a network of things which make a self-configuring network. The development of Intelligent Smart Farming IoT based devices is day by day turning the face of agriculture production. This report proposes an IoT based Smart Farming System assisting farmers in getting Live Data (Temperature, Soil Moisture) for efficient environment monitoring.

The following are some of the exceptional ideas and successes that these papers helped us define and create with the help of Agribot.

Wireless sensor networks were proposed by Balaji Banu [1] as a way to track and analyze farming conditions and enhance the quality and quantity of crops... Sensors are utilized for monitoring environmental conditions such as water level, temperature, humidity, and so on. This designing system employs ATMEGA8535 and IC- S8817 BS processors, analog to digital conversion, and wireless sensor nodes with wireless transmitting device modules. To retrieve and store information, databases and web applications are used. The sensor node failure and energy efficiencies are handled in this experiment.

The goal of Smart Farming with Agribot Paper [8] is to create "robot farms" where machines will do all of the work. Agribot performs two operations: digging a ploughing hole in the field, planting the seeds at regular intervals and covering the ploughed area with soil. The stepper motor is used to plant seeds and the spike wheel is used to dig a hole. When the robot begins to perform simultaneously, it can detect obstacles in its path using infrared sensors. PSCO controller is utilized to manage all operations. To boost efficiency and make their jobs easier with its multitasking features. By developing this, they are able to overcome the difficulties in their fields regardless of the season.

The paper "Autonomous agribot using Arduino" [6] describes the software as well as the hardware needed for the whole agribot setup. Utilizing Android applications with the wireless internet interface, ultrasonic and a digital compass sensors are used. A working prototype of an agricultural robot for seeding and fertilizing was created by Shiva Prasad B.S., Ravishankar M.N., and B.N. Shoba [8] using a microcontroller. The machine can detect the soil's pH level, temperature, humidity, and the presence of moisture of the soil. The robot is managed by the distant users. The machine is moved to the desired location and the previous actions are carried out using a remote control. The robot can access the internet. The machine is guided to the desired location by DC motors, and its remote controls regulate its speed.

# EMBEDDED SYSTEMS

## BASIC EMBEDDED SYSTEM

An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such asa personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems are controlled by one or more main processing cores that are typically either microcontrollers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors. For example, air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites. (Each radar probably includes one or more embedded systems of its own). Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase their liability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system "is not a strictly definable term, as most system shave some element of extensibility or programmability. For example, handheld computers share some elements with embedded systems such as the operating systems and microprocessors which power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems which don't expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions", and is thus appropriate to call "embedded".

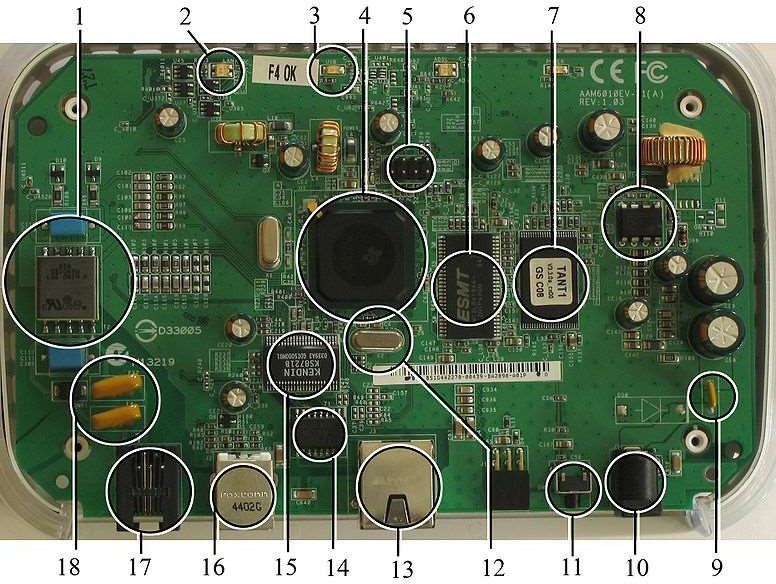
Labeled parts include microprocessor (4), RAM (6), flash memory (7). Embedded systems programming is not like normal PC programming.

Fig2.1.1: A modern Example of Embedded System

In many ways, programming for an embedded system is like programming PC 15 yearsago. The hardware for the system is usually chosen to make the device as cheap as possible. Spending an extra dollar, a unit in order to make things easier to program can cost millions. Hiring a programmer for an extra month is cheap in comparison. This means the programmer must make do with slow processors and low memory, while at the same time battling a need for efficiency not seen in most PC applications. Below is a list of issues specific to the embedded field.

### HISTORY:

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital system installed on the Apollo Command Module and Lunar Module. The first computer to use ICs, it helped astronauts collect real-time flight data.

In 1965, Autonetics, now a part of Boeing, developed the D-17B, the computer used in the Minuteman I missile guidance system. It is widely recognized as the first mass-produced embedded system. When the Minuteman II went into production in 1966, the D-17B was replaced with the NS-17 missile guidance system, known for its high-volume use of integrated circuits. In 1968, the first embedded system for a vehicle was released; the Volkswagen 1600 used a microprocessor to control its electronic fuel injection system.

By the late 1960s and early 1970s, the price of integrated circuits dropped and usage surged. The first microcontroller was developed by Texas Instruments in 1971. The TMS1000 series, which became commercially available in 1974, contained a 4-bit processor, read-only memory (ROM) and random-access memory ([RAM](https://www.techtarget.com/searchstorage/definition/RAM-random-access-memory)), and it cost around $2 a piece in bulk orders.

Also, in 1971, Intel released what is widely recognized as the first commercially available processor, the 4004. The 4-bit microprocessor was designed for use in calculators and small electronics, though it required eternal memory and support chips. The 8-bit Intel 8008, released in 1972, had 16 KB of memory; the Intel 8080 followed in 1974 with 64 KB of memory. The 8080's successor, the x86 series, was released in 1978 and is still largely in use today.

In 1987, the first embedded operating system, the real-time VxWorks, was released by Wind River, followed by Microsoft's Windows Embedded CE in 1996. By the late 1990s, the first embedded Linux products began to appear. Today, Linux is used in almost all embedded devices.

In the earliest years of computers in the 1930–40s, computers were sometimes dedicated to a single task, but were far too large and expensive for most kinds of tasks performed by embedded computers of today. Over time however, the concept of programmable controllers evolved from traditional electromechanical sequencers, via solid state devices, to the use of computer technology,

One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory. At the project's inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. A nearly mass-produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in1961.It was built from transistor logic and had a hard disk for main memory. When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high-volume use of integrated circuits.

### TOOLS:

Embedded development makes up a small fraction of total programming. There's also a large number of embedded architectures, unlike the PC world where 1 instruction set rules, and the UNIX world where there's only 3 or 4 major ones. This means that the tools are more expensive. It also means that they're lowering featured, and less developed. On a major embedded project, at some point user will almost always find a compiler bug of some sort.

Debugging tools are another issue. Since one can't always run general programs on embedded processor, one can't always run a debugger on it. This makes fixing their program difficult. Special hardware such as JTAG ports can overcome this issue in part. However, if user stop on a breakpoint when user system is controlling real world hardware (such as a motor), permanent equipment damage can occur. As a result, people doing embedded programming quickly become masters at using serial IO channels and error message style debugging.

### RESOURCES:

To save costs, embedded systems frequently have the cheapest processors that can do the job. This means user programs need to be written as efficiently as possible. When dealing with large data sets, issues like memory cache misses that never matter in PC programming can hurt anyone. Luckily, this won't happen too often, use reasonably efficient algorithms to start, and optimize only when necessary. Of course, normal profilers won't work well, due to the same reason debuggers don't work well. Memory is also an issue. For the same cost savings reasons, embedded systems usually have the least memory they can get away with. That means their algorithms must be memory efficient (unlike in PC programs, one will frequently sacrifice processor time for memory, rather than the reverse). It also means one can't afford to leak memory. Embedded applications generally use deterministic memory techniques and avoid the default "new" and "malloc" functions, so that leaks can be found and eliminated more easily. Other resources programmers expect may not even exist. For example, most embedded processors do not have hardware FPUs (Floating-Point Processing Unit). These resources either need to be emulated in software, or avoided altogether.

## NEED FOR EMBEDDED SYSTEMS:

The first reason why we need embedded systems is because general-purpose computers, like PCs, would be far too costly for the majority of products that incorporate some form of embedded system technology (Christoffer, 2006). Another reason why we need embedded systems is because general-purpose solution might also fail to meet a number of functional or performance requirements such as constraints in power-consumption, size-limitations, reliability or real-time performance etc.

The digital revolution, started decades ago, has reached a stage that we cannot conduct our normal modern daily lives without this technology. Indeed, it is safe to say that we already own at least one piece of equipment, which contains a processor, whether it is a phone, a television, an automatic washing machine or an MP3 player.

The colossal growth of processing power in small packages has fuelled the digital revolution.

All sectors of the economy have been influenced by the digital revolution and the industry has experienced tremendous developments in all aspects of engineering disciplines (Bruce, 2011).

The uses of embedded systems are virtually limitless, because every day new products are introduced to the market that utilizes embedded computers in novel ways. In recent years, hardware such as microprocessors, microcontrollers, and FPGA chips have become much cheaper. So, when implementing a new form of control. It's wiser to just buy the generic chip and write their own custom software for it. Producing a custom-made chip to handle a particular task or set of tasks costs far more time and money. Many embedded computers even come with extensive libraries, so that "writing own software" becomes a very trivial task indeed. From an implementation viewpoint, there is a major difference between a computer and an embedded system. Embedded systems are often required to provide Real-Time response. The main elements that make embedded systems unique are its reliability and ease in debugging.

### DEBUGGING:

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate, they can be roughly grouped into the following areas:

* + - * Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g., Fort hand Basic).
      * External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi-core systems.
      * An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or Nexus interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
      * An in-circuit emulator replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.
      * A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.
      * Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as assembly code or source-code.

Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software (and microprocessor) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, co-processor). An increasing number of embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronization of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the process or cores, which requires very low-level debugging, at signal/bus level, with a logic analyzer, for instance.

### RELIABILITY:

Embedded systems often reside in machines that are expected to run continuously for years without errors and in some cases recover by them if an error occurs. Therefore, the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

* + - * The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.
      * The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backups selected by an operator. Examples include aircraft navigation, react or control systems, safety-critical chemical factory controls, train signals, engines on single-engine aircraft.
      * The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.
      * A variety of techniques are used, sometimes in combination, to recover from errors both software bugs such as memory leaks, and also soft errors in the hardware:
* Watchdog timer that resets the computer unless the software periodically notifies the watchdog.
* Subsystems with redundant spares that can be switched over to software "limp modes" that provide partial function.
  + - * Designing with a Trusted Computing Base (TCB) architecture [6] ensures a highly secure & reliable system environment.
      * An Embedded Hypervisor is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software. This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
      * Immunity Aware Programming

### EXPLANATION OF EMBEDDED SYSTEMS:

* + 1. **Software architecture:**

There are several different types of software architecture in common use.

### Simple control loop:

In this design, the software simply has a loop. The loop calls subroutines, each of which manages a part of the hardware or software.

### Interrupt Controlled System:

Some embedded systems are predominantly interrupt controlled. This means that tasks performed by the system are triggered by different kinds of events. An interrupt could be generated for example by a timer in a predefined frequency, or by a serial port controller receiving a byte. These kinds of systems are used if event handlers need low latency and the event handlers are short and simple.

An interrupt controller multiplexes a number of possible interrupt sources on the platform for presentation to the processor. The interrupt controller in embedded systems must be configured to prioritize and route interrupts from devices within the SOC and externally attached devices. The Intel architecture–based SOC uses the traditional [Intel architecture](https://www.sciencedirect.com/topics/computer-science/intel-architecture) interrupt processing capabilities. This section outlines some of the low-level [configuration items](https://www.sciencedirect.com/topics/computer-science/configuration-item) associated with the Intel architecture interrupt controllers; however, details of the interrupt controller are specified in Chapter 4.

The requirement to ensure backward-compatible software on Intel architecture platforms extends far beyond the [instruction set architecture](https://www.sciencedirect.com/topics/computer-science/instruction-set-architecture); Intel has gone to extreme lengths to ensure that legacy software can continue to run on new generations of processors. To this end, a number of interrupt controllers are instantiated on Intel architecture platforms. The most basic is known as the 8259. The 8259 Peripheral Interrupt Controller (PIC) was first developed for the 8086 16-bit processor. The controller is very simple to set up and control, but at this point new software should not and in general does not set up the platform to use it. The modern interrupt controller on the Intel architecture platform is known as the local Advanced Peripheral Interrupt Controller (APIC) and I/O APIC. The local APIC is contained within the processor and controls the delivery to the processor. The local APIC is memory mapped to a physical address of 0xFEE00000. In a system with multiple logical processors, there is one local APIC per logical processor. A single processor with symmetric multithreading (SMT) consists of two hardware threads within one core. The local APIC will always appear in the same location for each processor, but each processor accesses its own local APIC. Each local APIC provides a local vector table (LVT). The LVT specifies the manner in which the interrupts are delivered to the core. The IOxAPIC is outside the CPU and integrated into the SOC or chipset; it expands the number of interrupt lines to 24. Each interrupt controller line has a redirection table. The interrupt descriptor tables provide a vector number for the associated interrupt request line. The combination of the local APIC and IOxAPIC on the platform allows interrupts from the devices to be assigned to a specific [interrupt vector](https://www.sciencedirect.com/topics/computer-science/vectors-interrupt) and targeted CPU core. A vector-based interrupt controller improves the efficiency of interrupt processing, as the CPU core can quickly start execution of the appropriate interrupt processing routine. Non-vector-based interrupt controllers usually require the CPU to query the interrupt controller to establish which interrupt to process. The latest ARM processor cores (M3) have introduced a vectored interrupt controller to reduce the overheads traditionally associated with interrupt processing.

When the Intel architecture processor is running in protected mode, the CPU uses the interrupt descriptor table (IDT). The IDT is a table of 256 vectors for exceptions and interrupts.

Usually, these kinds of systems run a simple task in a main loop also, but this task is not very sensitive to unexpected delays. Sometimes the interrupt handler will add longer tasks to a queue structure. Later, after the interrupt handler has finished, these tasks are executed by the main loop. This method brings the system close to a multitasking kernel with discrete processes.

### Cooperative Multitasking:

A non-preemptive multitasking system is very similar to the simple control loop scheme, except that the loop is hidden in an API. The programmer defines a series of tasks, and each task gets its own environment to “run” in. When a task is idle, it calls an idle routine, usually called “pause”, “wait”, “yield”, “nop” (stands for no operation), etc. The advantages and disadvantages are very similar to the control loop, except that adding news of ware is easier, by simply writing a task, or adding to the queue-interpreter.

### Primitive Multitasking:

### In this type of system, a low-level piece of code switches between tasks or threads based on a timer (connected to an interrupt). This is the level at which the system is generally considered to have an "operating system" kernel. Depending on how much functionality is required, it introduces more or less of the complexities of managing multiple tasks running conceptually in parallel. As any code can potentially damage the data of another task (except in larger systems using an MMU) programs must be carefully designed and tested, and access to shared data must be controlled by some synchronization strategy, such as message queues, semaphores or a non-blocking synchronization scheme. Because of these complexities, it is common for organizations to buy a real-time operating system, allowing the application programmers to concentrate on device functionality rather than operating system services, at least for large systems; smaller systems often cannot afford the overhead associated with a generic real time system, due to limitations regarding memory size, performance, and/or battery life.

### 2.3.1.5 Micro kernels and Exokernels:

A Kernel is an intermediary between applications and hardware. This means that applications can run without knowing or caring about the underlying hardware details. It manages low-level tasks such as disk management, task management, and [memory management](https://www.shiksha.com/online-courses/articles/memory-management-techniques-in-operating-system/). Whenever you start a system, Kernel is the first program that is loaded after the bootloader, and it remains in the memory until the [**operating system**](https://www.shiksha.com/online-courses/articles/types-of-operating-systems/) is shut-down. The core component keeps your system running, managing all your software and hardware. Without it, your computer would be a useless pile of metal and plastic. A kernel must be fast and responsive. It needs to handle all the requests and responses quickly and efficiently. To do this, the Kernel uses a lot of kernel-level memory. This memory is isolated from the main memory of your computer. This means that the Kernel can use it to store data and code without affecting other applications.

A microkernel is a logical step up from a real-time OS. The usual arrangement is that the operating system kernel allocates memory and switches the CPU to different threads of execution. User mode processes implement major functions such as file systems, network interfaces, etc. In general, microkernels succeed when the task switching and intertask communication is fast, and fail when they are slow. Exokernels communicate efficiently by normal subroutine calls. The hardware and all the software in the system are available to, and extensible by application programmers.

Microkernels are a newer development and, as such, are not as common as monolithic kernels. They include only the essential services and devices required for the system to function. This results in a smaller kernel that is faster and uses less memory. Here, the user and kernel services are implemented in two different spaces. It has separate User Space and Kernel Space. This reduces the size of the Kernel and results in reducing the size of the operating system.

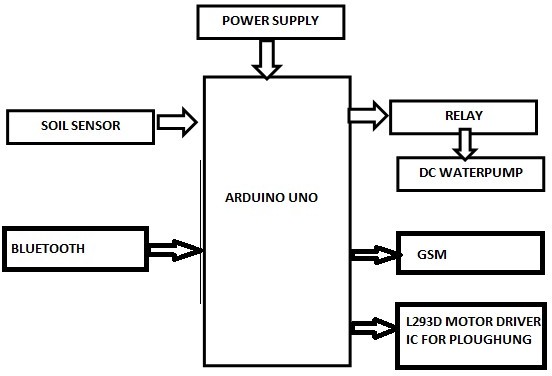
This Kernel has separate resource protection and management. It is suitable for use when performing application-specific customization. Exo kernels are designed for use in mobile devices. They are a variation of microkernels that include additional features specifically for mobile devices, such as power management and support for multiple processors.

# HARDWARE DESCRIPTION

The main components of the project are:

* + - * + ArduinoUNO
        + SoilMoistureSensor
        + Bluetooth
        + L293DMotorDriver
        + GSM Module
        + DC Motors
        + WaterPump

## BLOCKDIAGRAM

Fig 3.1: Block diagram of the whole system

* 1. **ARDUINO UNO**

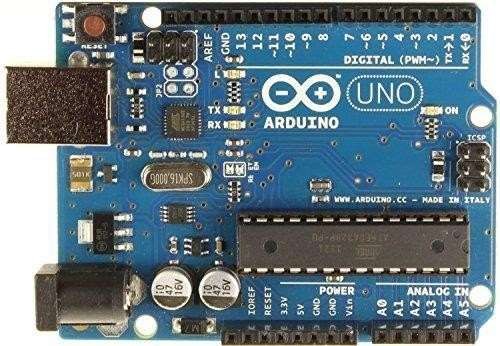
The Atmel AVR core combines a rich instruction set with 32 general-purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction executed in one clock cycle.

Fig 3.2: Arduino UNO Board

This allows very fast start-up combined with low power consumption. The On-chip ISPF lash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip. Boot program running on the AVR core. The ATmega328/P is supported with a full suite of program and system development tools including C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

In simple words, Arduino is an Embedded development platform that consists of both the hardware as well as software parts. Let’s have a look at the hardware side. As we know like any other development board, the Arduino is also made of many basic components. In this article, we will discuss what are those basic **components or parts of an Arduino Board**. This article is completely focused on Arduino UNO only, if you want to know about the other options you can check out our previous article on [Different Types of Arduino Boards](https://circuitdigest.com/article/different-types-of-arduino-boards) where we have compared all the popular Arduino boards and their specifications.

In the UNO board, the main component is the **ATMega328P**. It is the heart of the Arduino UNO. Near the MCU you can see a 16MHz resonator which will give the ATMega328P the clock signal to work. Near that, you can also see a connector named ICSP. It is used to burn the **Arduino bootloader**into the chip. And you can also see the header pins for the I/O.

If you look at the other side of the board, you can spot another microcontroller in a QFN package. It is an **ATMega16U** and is used as a USB -TTL converter. Near that, it will have its crystal and ICSP port to burn the firmware. There will be a reset button near it, which will reset the ATMega328P.

You can see the USB port and DC barrel jack on the left side. You can power the Arduino either through the USB port or the barrel jack. The barrel jack will accept a voltage range of 7-12V. And near the barrel jack, you can find two voltage regulators. One for 5V and one for 3.3V. Let’s check out each component.

### Features:

* + - * 28-pin AVR Microcontroller
      * Flash Program Memory:32Kbytes
      * EEPROM Data Memory: 1Kbytes
      * SRAM Data Memory: 2Kbytes
      * I/O Pins: 23
      * Timers: Two 8-bit/One 16-bit
      * A/D Converter: 10-bit Six Channel
      * PWM: Six Channels
      * RTC: with Separate Oscillator
      * MSSP: SPI and IC Master and Slave Support
      * USART: Yes
      * External Oscillator: upto 20MHz

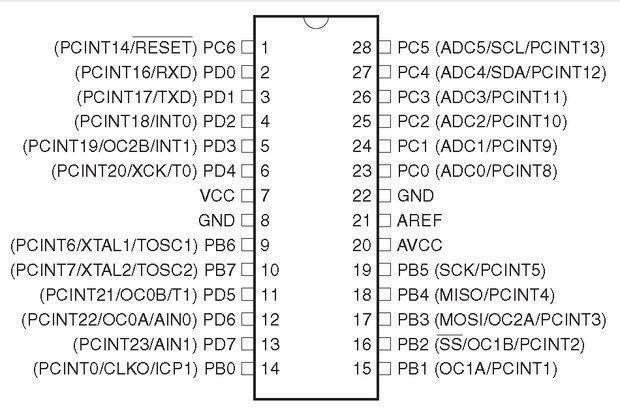


Fig 3.3: Pin Configuration

### Description

Arduino UNO board is the most popular board in the Arduino board family. It is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have the majority of these components in common.

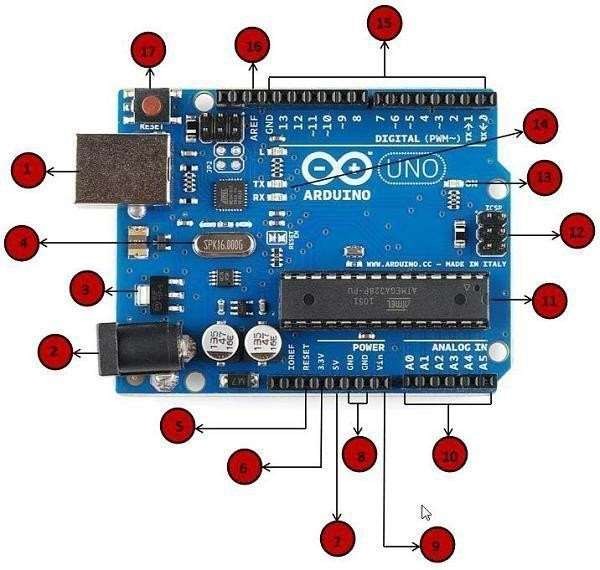


Fig3.4:Arduino UNOboardspecifications

**Power USB:** Arduino board can be powered by using the USB cable from the computer. There is a need to do is connect the USB cable to the USB connection. The USB socket on the UNO has two functions. One is for communication, to connect with the computer through a USB port, and also to load the firmware into the Arduino with the help of the bootloader. The second is to power the Arduino. You can use the USB port to power the Uno directly from any USB port.

**Power (Barrel Jack):** Arduino boards can be powered directly from the AC mains power supply by connect in git to the Barrel Jack.it is used to supply power to the UNO. We can supply 7-12V through it and hence we can use a 12V DC adapter or 9V DC adapter on this Jack to power the Arduino board.

**Voltage Regulator:** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the process or and other elements. The ATMega328 and ATmega16U2 have a maximum input voltage of around 5V and most modules or accessories work on either 5V or 3.3V. The Arduino can accept 7-12V through the Vin pin or the DC barrel jack. So, to step it down, there are two regulators onboard. One is a 5V regulator (marked as 1) for the microcontrollers and the other one is a 3.3V regulator which is used to provide 3.3V through 3.3V pin.

**Crystal Oscillator:** The Crystal oscillator helps Arduino incalculating time. The number printed on top of the Arduino crystal is 16.000H9H. It tells that the frequency is 16,000,000Hertzor 16MHz. For a microcontroller to work it needs a clock source. The clock circuit determines the speed with which the microcontroller operates. How many instructions per second it will execute is dependent on the clock frequency. The ATMega series microcontrollers can use two types of clock sources. One is an

internal RC oscillator that is already built into the microcontroller. But the drawback of using the internal oscillator is that its maximum frequency is limited and it is not that accurate. That is where the second option comes into place, i.e., using an external clock generator. In this case, we will be using a Quartz crystal oscillator or a ceramic resonator for this purpose. In the picture below, you can see two components are marked. The first one is a 16MHz crystal oscillator used for the ATMega16U2 chip and the second one is a 16MHz resonator used for the ATMega328P microcontroller.

**Arduino Reset:** Arduino board can be reset. UNO board can be reset in two ways. First, by using the reset button (17) on the board. Second, we can connect an external reset button to the Arduino pin labeled RESET.As the name indicates this tactile switch is used to reset the ATMega328 microcontroller. It’s connected to the PC6/Reset pin, which is pulled up through a 10K. When the switch is pressed the pin is pulled to the ground and the chip will reset.

### Pins (3.3, 5, GND, Vin)

3.3V (6) −Supply 3.3 output volt

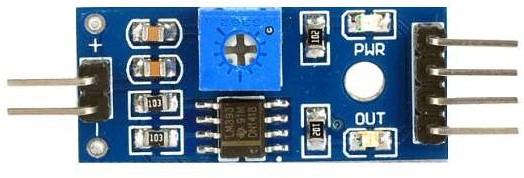
5V (7) − Supply 5 output volt

GND (8) (Ground) – There are several GND pins on the Arduino, any of which can be used to ground the circuit.

Vin (9) – This pin also can be used to power the Arduino board from an external power source, like an AC mains power supply.

**Analog pins:** The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it in to a digital value that can be read by the microprocessor.

**3.3 SOIL MOISTURE SENSOR**

 The Moisture sensor is used to measure the water content (moisture) of soil. When the soil is having water shortage, the module output is at high level; else the output is at low level. This sensor reminds the user to water their plants and also monitors the moisture content of soil. It has been widely used in agriculture, land irrigation and botanical gardening. Capacitance is used by the soil moisture sensor to gauge the surrounding medium's dielectric permittivity. Dielectric permittivity within side the soil relies upon on the quantity of water present. The sensor generates a voltage that is proportional to the soil's water content and dielectric permittivity. The sensor takes an average of the water content along its entire length. Regarding the sensor's flat surface, there is a 2-cm zone of influence, but the edges have little to no sensitivity. The Soil Moisture Sensor is used to monitor soil moisture content to control irrigation in greenhouses, enhance bottle biology experiments, and determine the loss of moisture as time passes due to evaporation and plant uptake.

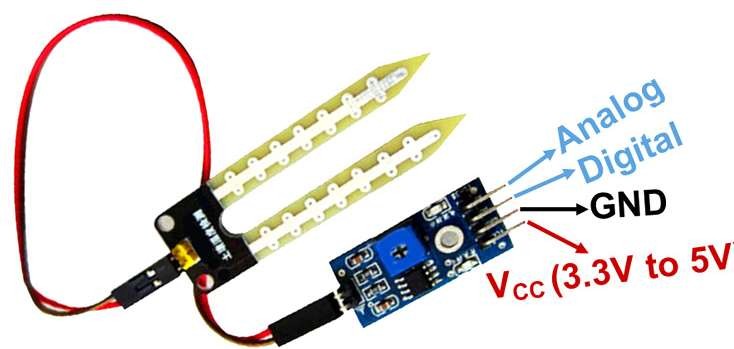
Fig 3.5: Moisture Sensor Module

Fig 3.6: Pin Configuration

### 3.3.1 WORKING PRINCIPLE:

The Soil Moisture Sensor uses capacitance to measure dielectric permittivity of the surrounding medium. In soil, dielectric permittivity is a function of the water content. The sensor creates a voltage proportional to the dielectric permittivity, and therefore the water content of the soil. The sensor averages the water content over the entire length of the sensor. There is a 2 cm zone of influence with respect to the flat surface of the sensor, but it has little or no sensitivity at the extreme edges. The Soil Moisture Sensor is used to measure the loss of moisture over time due to evaporation and plant uptake, evaluate optimum soil moisture contents for various species of plants, monitor soil moisture content to control irrigation in green houses and enhance bottle biology experiments.

### FEATURES AND SPECIFICATIONS:

* + - * Operating Voltage:3.3Vto 5VDC
      * Operating Current:15mA
      * Output Digital - 0V to 5V, Adjustable trigger level from preset
      * Output Analog - 0V to 5V based on infrared radiation from fire flame falling on the sensor
      * LEDs indicating output and power
      * PCB Size: 3.2cm x 1.4cm
      * LM393 based design
      * Easy to use with Microcontrollers or even with normal Digital/Analog IC
      * Small, cheap and easily available

## 3.4 WATER MOTOR

Micro DC 3-6V Micro Submersible Pump Mini water pump For Fountain Garden Mini water circulation System DIY project. This is a low cost, small size Submersible Pump Motor which can be operated from a3~6V power supply. It can take upto 120 liters per hour with very low current consumption of 220mA. Just connect tube pipe to the motor outlet, submerge it in water and power it. Make sure that the water level is always higher than the motor. Micro submersible Pump DC 3-6V DIY project: A mini water pump for a water feature in the garden This small, inexpensive submersible pump motor can be powered by a 3 to 6 volt power source. It has a maximum flow rate of 120 litres per hour and uses very little current (220 mA). You only need to attach a tube pipe to the motor outlet, submerge it in water, and then power it. Make certain the motor is in no way submerged underneath the water. Dry running will make noise and may cause the motor to heat up. Dry run may damage the motor due to heating and it will also produce noise.

Fig 3.7: Water Motor

**3.3.2 SPECIFICATIONS:**

* + - * OperatingVoltage:3-6V
      * OperatingCurrent: 130-220mA
      * FlowRate: 80-120L/H
      * MaximumLift:40-110mm
      * ContinuousWorking Life: 500 hours
      * Driving Mode: DC, Magnetic Driving
      * Material: Engineering Plastic
      * Outlet outside, insideDiameters:7.5 mm, 5 mm.

## GSM MODULE

GSM (Global System for Mobile communications) is a cellular network, which means that mobile phones connect to it by searching for cells in the immediate vicinity. GSM networks operate in four different frequency ranges. Most GSM networks operate in the 900 MHz or 1800MHz bands. Some countries in the Americas use the 850 MHz and 1900 MHz bands because the 900 and 1800MHz frequency bands were already allocated. There are 400 and 450MHz frequency bands are assigned in some countries, where these frequencies were previously used for first-generation systems.GSM-900 uses 890–915 MHz to send information from the mobile station to the base station (uplink) and 935–960 MHz for the other direction (downlink), providing 124 RF channels (channel numbers 1 to 124) spaced at 200 kHz.

Duplex spacing of 45 MHz is used. In some countries the GSM-900 band has been extended to cover a larger frequency range. This 'extended GSM', E-GSM, uses 880–915 MHz (uplink) and 925–960MHz (downlink), adding 50 channels (channel numbers 975 to 1023 and 0) to the original GSM-900 band. Time division multiplexing is used to allow eight full-rate or sixteen half-rate speech channels per radio frequency channel. There are eight radio times lots (giving eight burst periods) grouped into what is called a TDMA frame. Half rate channels use alternate frames in the same timeslot. The channel data rate is 270.833 kbit/s, and the frame duration is 4.615ms.

* + 1. **ADVANTAGES**:

GSM also pioneered a low-cost, to the network carrier, alternative to voice calls, the short message service (SMS, also called "text messaging"), which is now supported on other mobile standards as well. Another advantage is that the standard includes one worldwide Emergency telephone number, 112. This makes it easier for international travelers to connect to emergency services without knowing the local emergency number.

* + 1. **THE NETWORK:**

GSM provides recommendations, not requirements. The GSM specifications define the functions and interface requirements in detail but do not address the hardware. The GSM network is divided into three major systems: the switching system (SS), the base station system(BSS), and the operation and support system (OSS).The purpose of OSS is to offer the cost-effective support for centralized, regional and local operational and maintenance activities that are required for a GSM network.

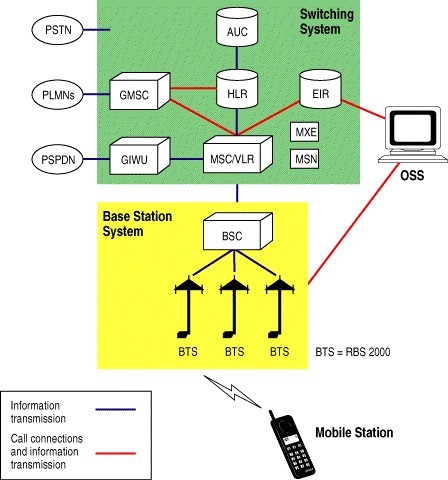


Fig 3.8: GSM Module

### OPERATION AND SUPPORT SYSTEM:

The operations and maintenance center (OMC) is connected to all equipment in the switching system and to the BSC. The implementation of OMC is called the operation and support system (OSS). The OSS is the functional entity from which the network operator monitors and controls the system. The purpose of OSS is to offer the customer cost-effective support for centralized, regional and local operational and maintenance activities that are required for a GSM network. An important function of OSS is to provide a network overview and support them a intense activities of different operation and maintenance organizations.

## DC MOTOR

A dc motor uses electrical energy to produce mechanical energy, very generally through the interaction of magnetic fields and current-containing conductors. There verse process, producing electrical energy from mechanical energy, is carried out by an alternator, source or dynamo. Many types of electric motor scan be run as sources, and vice versa. The input of a DC motor is current/voltage and its output is torque (speed).

The DC motor has two basic parts: the rotating part that is called the armature and the stable part that includes coils of wire called the field coils.

Fig3.9: DC Motor

The stationary part is also called up the stator. Figure shows a depict of a distinctive DC motor, Figure shows a picture of a DC armature, and Figure shows a picture of a distinctive stator. From the picture one can see the armature is made of coils of wire wrapped around the core, and the core has an covered shaft that rotates on charges. One should also notice that the ends of each coil of wire on the armature are finished at one end of the armature. The outcome points are called the commutator, and this is where's brushes make electrical contact to bring electrical current from the stationary part to the rotating part of the machine.

### OPERATION:

The DC motor one will find in modem industrial applications operates very similarly to the simple DC motor described earlier in this chapter. Figure 12-9 demon stares an electrical diagram of a simple DC motor. comment that the DC voltage is applied directly to the field winding and the brushes. The armature and the area are both shown as a coil of wire. In afterward diagrams, a field resistor will be added in series with the field to control the motor speed.

When voltage is applied to the motor, current begins to flow by the field coil from the negative terminal to the positive terminal. This sets up an inviolable magnetic field in the field winding. Current also begins to feed through the brushes into a commutator segment and then through an armature coil. The current goes forward to flow through the coil back to the brush that is attached to other end of the coil and returns to the DC power source. The current flow in the armature coil sets up a strong magnetic field in the armature.

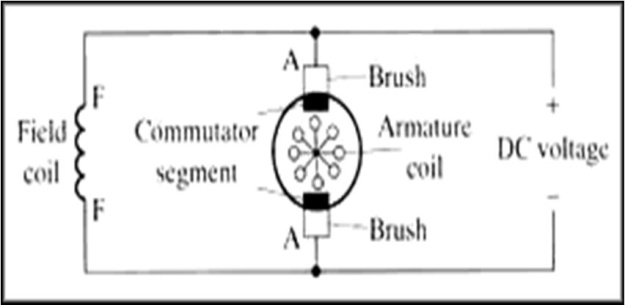


Fig3.10: Electrical Diagram of DC Motor

The magnetic field in the armature and field coil causes the armature to begin to rotate. This occurs by the dissimilar magnetic poles attracting each other and the like magnetic polesgrossingouteachother. As the armature begins to rotate, the commutator sections will also beg into move beneath the brushes. As an individual commutator segment moves under the brushconnected to positive voltage, it will become positive, and when it impresses under a brush connected to negative voltage it will turn negative.

In this way, the commutator segments continually change polarity from positive to negative. Since the commutator segments are associated to the ends of the wires that make up thefield winding in the armature, it induces the magnetic field in the armature to change polarity continually from north pole to south pole. The commutator sections and brushes are aligned in such a way that the switch in polarity of the armature coincides with the location of the armature's magnetic field and the field winding's magnetic field.

The switching accomplish is timed so that the armature will not lockup magnetically with the field. Rather the magnetic fields tend to build on each other and provide additional torque tokeep the motor shaft rotating. When the voltage is de-energized to the motor, the magnetic fields in that armature and the field winding will quickly diminish and the armature shaft's speed will begin to drop to zero. If voltage is enforced to the motor again, the magnetic fields will toughen and the armature will begin to rotate again.

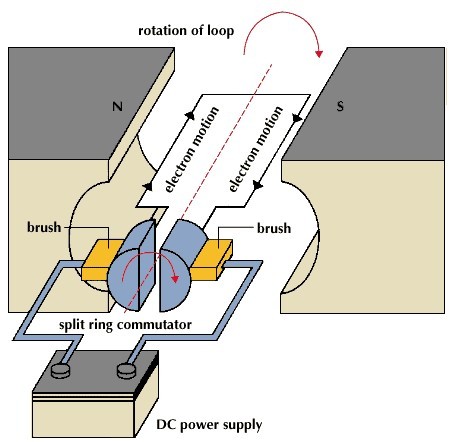


Fig 3.11: Operation of DC Motor

The Types of DC Motors are DC Shunt Motor, DC Series Motor, DC Long Shunt Motor (Compound), and DC Short Shunt Motor (Compound). The rotational energy that user get from any motor is usually the battle between two magnetic fields going after each other. The DC motor has magnetic poles and an armature, to windward DC electricity is flowed, The Magnetic Poles are electromagnets, and when they are energized, they develop a strong magnetic field approximately them, and the armature which is given power with a commutator, invariably repels the poles, and therefore rotates.

# SOFTWARE DESCRIPTION

This project is implemented using the following software:

* + Arduino IDE

## 4.1 ARDUINO IDE

Arduino IDE (Integrated Development Environment) is open-source programmingthat is used to write & compile code using a module that is Arduino. This is an officialprogramming software shown in Fig. 4.1, which makes compiling code simple so a typical man can understand the learning procedure.

Fig 4.1: Arduino IDE

This software is readily available for all operating systems like MAC, Windows, andLinux. Arduino Mega, Arduino Uno, Arduino Leonardo, and more are a range of Arduino modules that are available. Coding on this software mostly uses functions of C/C++. The Arduino Integrated Development Environment or Arduino Software (IDE) contains a texteditor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them. The Arduino IDE is an open source software, easy to use and user-friendly, writing the code to get the desired output and dumping it to on the board can be done easily. Since it is an open-source software users do not need license or special hardware components in their systems for the installation of such software. This software is readily available for all operating systems like MAC, Windows, and Linux. Arduino Mega, Arduino Uno, Arduino Leonardo, and more are a range of Arduino modules that are available. Coding on this software mostly uses functions of C/C++. This project uses Arduino IDE software to interface all the hardware components used for the functioning of the vehicle. Arduino IDE (Integrated Development Environment) is open-source programming that is used to write & compile code using a module that is Arduino. The Arduino IDE (Integrated Development Environment) or Arduino Software consists of a text editor for writing code, a message area, a textual content console, a toolbar with buttons for not unusual place functions, and a sequence of menus. It connects to the Arduino and Genuino hardware to add packages and talk with them.

### Writing a sketch

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ‘.ino’. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow one to verify and upload programs, create, open, and save sketches, and open the serial monitor. The toolbar consists of many icons. The first icon from the left is to verify, the second one is to upload, the third one is for opening a new project, the fourth one is to open a project and the fifth one is to save the project. The icon on the extreme right is for opening the serial monitor. The white area in the middle is the sketch area.

These icons are shown in Table 4.1. Additional commands are found within the five menus: File, Edit, Sketch, Tools, and Help. The menus are context-sensitive, which means only those items relevant to the work currently being carried out are available.

### Sketch book:

The Arduino Software (IDE) uses the concept of a sketch book: a standard place to store user programs (or sketches). The sketches in user sketch book can be opened from the File > Sketchbook menu or the Open button on the toolbar. The first-time user run the Arduino software, it will automatically create a directory for user’s sketch book. User can view or change the location of the sketch book location from with the Preferences dialog.

### Compiling a Sketch:

The Arduino code is just plain old C without the header part. When user press the 'compile' button, the IDE saves the current file as “arduino.c” in the 'lib/build' directory then it calls a “makefile” contained in the 'lib' directory. This makes file copies arduino.c as prog.c into'lib/tmp' adding 'wiringlite.inc' as the beginning of it. this operation makes the Arduino/wiring code into a proper C file (called prog.c).

After this, it copies all the files in the 'core' directory into 'lib/tmp'. these files are the implementation of the various Arduino/wiring commands adding to these files adds commands to the language. The core files are supported by pascal stang's prosy on avr-lib which is contained in the 'lib/avrlib' directory.

At this point, the code contained in lib/tmp is ready to be compiled with the C compiler contained in 'tools'. The Arduino Software (IDE) will display a message when the compiling is complete or shows an error. If the make operation is successful then user will have prog.hex ready to be downloaded into the processor and the display will be shown as in Figure.

### Uploading a Sketch

Before uploading your sketch, you need to select the correct items from the Tools>Board and Tools>Port menus. On the Mac, the serial port is something/dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) – to find out, you look for USB serial device in the ports section of the Windows Device Manager.

On Linux, it should be /dev/ttyACMx, /dev/ttyUSBx or similar. Once user has selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. On most boards, user will see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error. The display for successful uploading of the sketch is shown in Figure.

When user upload a sketch, user is using the Arduino boot loader, a small program that has been loaded onto the microcontroller on the board. It allows one to upload code without using any additional hardware. The boot loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The boot loader will blink the onboard (pin 13) LED when it starts (i.e., when the board resets).

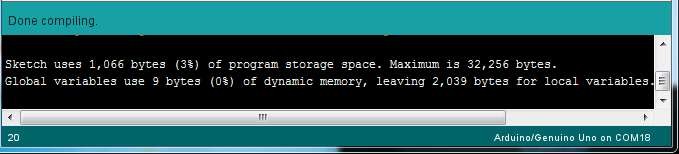


Fig. 4.2: Successful Compiling

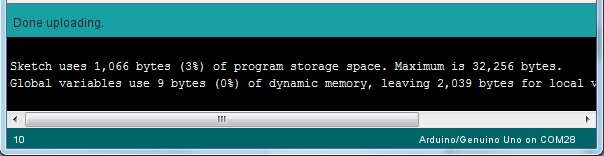


Fig. 4.3: Successful Uploading

Table 4.1: Arduino IDE Tool Bar Icons Description

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Symbol** | **Function** |
| 1 |  | **Verify**  Checks your code for errors compiling it. |
| 2 |  | **Upload**  Compiles your code and uploads it to the configured board.  Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using  Programmer" |
| 3 |  | **New**  Creates a new sketch. |
| 4 |  | **Open**  Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window  overwriting its content. |
| 5 |  | **Save**  Saves your sketch. |
| 6 |  | **Serial Monitor**  Opens the serial monitor |

# WORKING OF PROPOSED SYSTEM

## EXISTING MODEL

Agriculture is the backbone of India, but farmers have been facing a lot of problems lately, only rich farmers can afford all the needs for a crop to grow to meet the consumer's quality standards. Whereas when people take a look at the poor farmers who age more than forty years, who cannot do all the things on their own, and look at the risks of farming, Farmers are avoiding passing it on to their children. Keeping these listed farmers in mind water monitoring systems, seed sowing, and plowing functions are discovered mechanically so that it makes the farmer's job a little bit easy on them.

## PROPOSED MODEL

The Proposed work is to provide farmers a vehicle which supports him in all possibleways like help in ploughing with one switch, help in knowing soil moisture content and pumping water, and give all these alerts to farmers at all times.

## WORKINGPRINCIPLE

The project aims on the design, development and the fabrication of the robot whichcan dig the soil, leveler to close the mud and turn on and turn off the motor depending onwater level in the ground and this whole system of the robot works with a regulated powersupply of 12V. The language input allows a user to interact with the robot which is familiarto most of the people. The advantages of these robots are hands-free and fast data inputoperations. In the field of agricultural autonomous vehicle, a concept is been developed toinvestigate if multiple small autonomous machines could be more efficient than traditional large tractors and human forces. Keeping the above ideology in mind, a unit with the following feature is designed:

* + - Robot has rotor which will destroy the unwanted grasses while moving and also levelthe ground.
    - Allthe operationsare performed withthe help ofArduino UNO.
    - Moisture sensor is used to sense the water level in the ground and turn on and turn off the pumping motor depending on water level.
    - Bluetooth is used to send the message to the farmer about the operation performed byrobot.

The power supply to the robot is switched on. The movement of the robot is control led through a software-Software serial GPRS for Arduino which can be installed into the farmer’s mobile phone. The robot can be moved left, right, front and back through this software. There are two motors used for ploughing action, the first one is used to drive the vehicle and the second one is used to control the plougher. These two motors are also controlled by the L293D motor driver

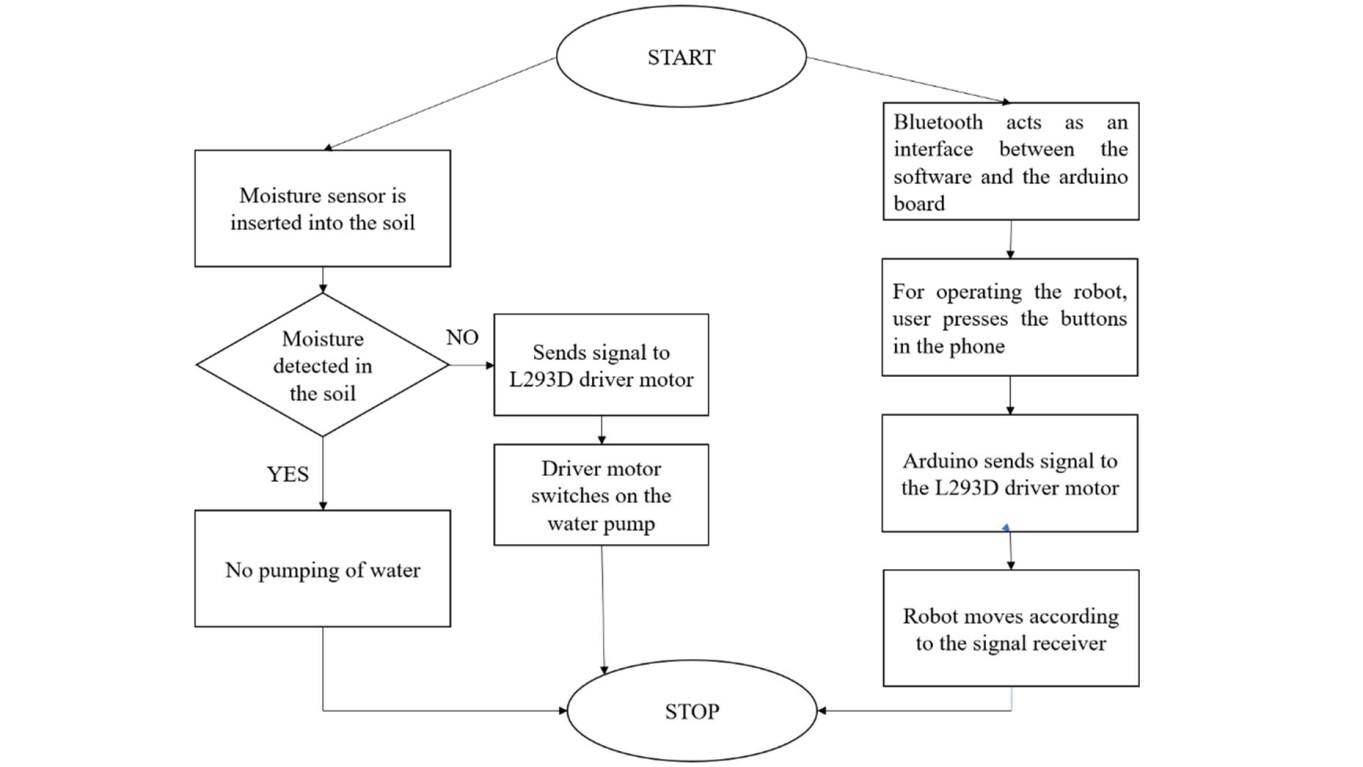
which receives the signals from the mobile phone software through the bluetooth module. Depending on the moisture level in the soil the L293D motor driver sends signals tothe water pump motor to switch on/off the pump. The actions of the water pump are also intimated to the farmer through the bluetooth module itself.

Fig 5.1: Flowchart of the whole system

# RESULTS AND DISCUSSION

## PLOUGHING OUTPUT

The L293D Driver motor is used to drive the vehicle. Here, the system has been setup such that the driver motor will receive signals from the mobile phone of the user, through bluetooth module and arduino. The movement of the vehicle and the movement of the plougher is controlled by the user through the arduino software installed in their mobile phone. The Bluetooth module installed in the vehicle receives the signal from user phone and sends the signal to arduino which controls the L293D motor driving the vehicle wheels accordingly. There are two motors which are used in the vehicle movement. One is to control the wheels of the vehicle and the other is to control the plougher. The mode of control is through the phone.

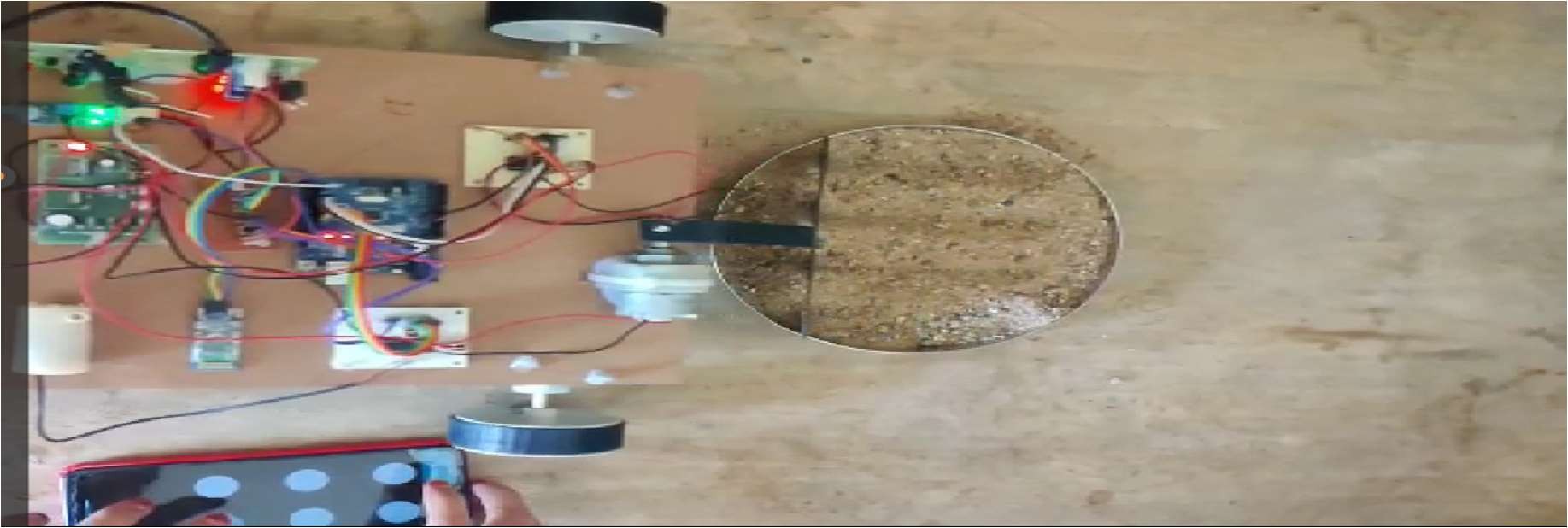


Fig 6.1 Ploughing Output

The control pallet and working of vehicle is shown in the figure below, where the operator is using the arduino software installed in the phone to control the movement of the vehicle and the plougher. There are total of 6 buttons to control the vehicle. The figure above shows the vehicle ploughing the sand being operated through the software installed in the user’s phone.

## MOISTURE SENSING OUTPUT

The moisture sensor and module are used to detect the moisture content in the soil and take necessary actions. The sensing pad of the moisture sensor is inserted into the soil, if the soil is dry the vehicle stops moving and the moisture sensor sends a signal to the water pump through the arduino. Once the soil is irrigated and the moisture sensor senses the moisture the vehicle can be moved again. The vehicle can’t be moved until the moisture sensor sends a signal to arduino saying that the soil has enough water and the water pump can be turned off. This process is repeated until the whole land area is watered enough. The water pump is turned on and off based on the output from the moisture sensor sent to it through the arduino board. The working of the moisture sensor is shown in the figure, here the soil has moisture above the set threshold value so the moisture sensor is showing two lights and the water pump is turned off through arduino.

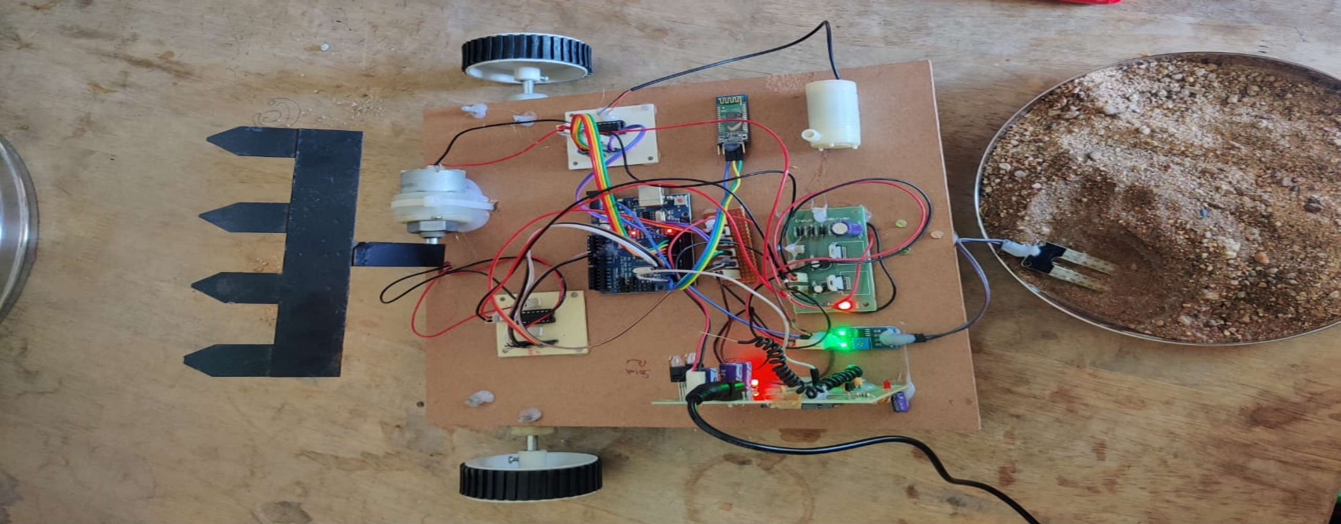


Fig 6.2 Moisture Sensing Output

The above figure shows that the moisture content of soil is normal, so there is no need to switch on the water pump. Since the water pump is turned off the vehicle is free to move based on the instructions received from the user mobile phone through the Bluetooth module on the vehicle.

* 1. **SEED SOWING OUTPUT**

The seed sowing program in this vehicle is controlled manually. The is a switch to turn on or off the seed sowing function and a container is attached to the vehicle to carry and drop the seeds. The switch attached to the vehicle needs to be turned on in order to access the seed sowing facility. The seeds need to be put in the container shown below and the vehicle drop a few number of seeds along it’s path of travel. The seeds are just dropped on to the surface of the soil so this function is suitable for crops like cotton and millets.



Fig 6.3 Seed sowing output

The above figure 6.3 shows the seed sowing container and the motor used to operate the seed sowing function. A D.C motor is used for this function as it is of low cost and is most suitable for this project.

# ADVANTAGES AND DISADVANTAGES

## ADVANTAGES

* + - Reduces the number of labors required for agricultural activity.
    - Due to its quick action, time will be saved.
    - Agribot can able to work in any environmental condition.
    - The robots can work without sleep so they can work 24\*7 for 365 days.
    - Protection against harmful effects of chemicals.

## DISADVANTAGES

* + - As machines will do all the work there will be less scope for employment of labor mainly in rural villages.
    - Robot needs high maintenance.
    - In a way or other emotional appeal of agriculture may be changed.

**8.1 APPLICATIONS**

# APPLICATION

* Robot has rotor which will destroy the unwanted grasses while moving.
* The robot also levels the ground.
* The robot also has a digger to dig the vegetables from the ground.
* The robot automatically turns on and turns off water pumping motor.
* The robot checks the moisture content in the soil.
* The robot intimates the user about the actions.
* It can be used to plough the ground.
* The movement of robot can be controlled through mobile phone.
* Seed Sowing can also be done and the length between each seed can also be improvised with further applications.

# CONCLUSION and FUTURE SCOPE

Internet on things and cloud computing collectively makes a system that control agriculture sector effectively. This system will sense all the environmental parameters and send the data to the user via cloud. User will take controlling action according to that this will be done by using actuator. This asset allows the farmer to improve the cultivation in a waythe plant need. It leads to higher crop yield, prolonged production period, better quality and less use of protective chemicals. This equipment may be in our future, but there are important reasons for thinking that it may not be just replacing the human driver with a computer. It may mean a rethinking of how crop production is done.

Crop production may be done better and cheaper with a swarm of small machines than with a few large ones. One of the advantages of the smaller machines is that they may be more acceptable to the non-farm community. The jobs in agriculture are a drag, dangerous, require intelligence and quick, though highly repetitive decisions hence robot scan be rightly substituted with human operator. The higher quality products can be sensed by machines (color, firmness, weight, density, ripeness, size, shape) accurately. Robots can improve the quality of our lives but there are down sides.

The present situation in our country all the agricultural machine is working on manual operation otherwise by petrol engine or tractor is expensive, farmer can’t work for long timemanually to avoid this problem, people need to have some kind of power source system to operate the digging machine.

There is a need for autonomous and time saving technology in agriculture to have efficient farm management. The prime benefits of development of autonomous and intelligent agricultural robots are to improve repeatable precision, efficacy, reliability and minimization of soil compaction and drudgery. Agricultural robots are designed using different localization techniques which are vision, GPS, and sensor-based navigation control system.

# REFERENCES

1. Blackmore, S. (2007). A systems view of agricultural robotics. Precision Agriculture conference, Wageningen Academic Publishers, the Netherlands. pp. 23-31.
2. Shibusawa, S. 1996. Phyto Technology - An introduction to the concept and topic of a new Project. <http://phytech.ishikawa>c.ac.jp/WhatIs.html.
3. Tillett, N.D., Hague, T. and Marchant, J.A. (1998) A robotic system for plant scalehusbandry. Journal of Agricultural Engineering Research, 69, 169-178.
4. Rajalakshmi, Mrs. S. Devi Mahalakshmi “IOT Based Crop-Field Monitoring and Irrigation Automation” 10th International conference on Intelligent systems and control (ISCO),7-8 Jan 2016 published in IEEE Xplore Nov 2016.
5. Prof. K. A. Patil And Prof N. R. Kale proposes “A Model for Smart Agriculture Using IOT” 2016 International Conference on Global Trends in signal Processing, Information Computing and Communication.
6. Dr.N. Suma, Sandra Rhea Samson, S. Saranya, G. Shanmugapriya, R. Subhashri „IOT Based Smart Agriculture Monitoring System ‟2017 International Journal on Recent and Innovation Trends in Computing and Communication.
7. Mahammadshareef Mekala, Dr.P. Viswanathan„ A Survey: Smart agriculture IoT with cloud Computing ‟978-1-5386-1716-8/17/$31.00 ©2017 IEEE
8. Prathibha S R1, Anupama Hongal 2, Jyothi M P3‟ IOT BASED MONITORING SYSTEM IN SMART AGRICULTURE‟ 2017 International Conference on Recent Advances in Electronics and Communication Technology
9. Ibrahim Mat, Mohamed Rawidean Mohd Kassim, Ahmad Nizar Harun, Ismail MatYusoff “IOT in Precision Agriculture Applications Using Wireless Moisture SensorNetwork” 2016 IEEE Conference on Open Systems (ICOS), October 10-12- 2016, Langkaw, Malaysia.
10. Zhaochan Li, JinlongWang, Russell Higgs, LiZhou WenbinYuan4 “Design of an Intelligent Management System for Agricultural Greenhouses based on the Internet of Things” IEEE International Conference on Embedded and Ubiquitous Computing (EUC)2017.

# APPENDIX

## SOURCE CODE:

#include<SoftwareSerial.h>

SoftwareSerialgprs=SoftwareSerial(2,3);//CONNECTTXOFbluetoothTO2ANDRXOFbluetooth TO 3 //

#definem11 4

#definem12 5

#definem21 6

#definem22 7

#definem31 10

#definem32 11

#definem41 8

#definem42 9

#definesoil12

unsigned int RecievedData,i = 0;intpos=0;

voidSendSMS(StringData){ ////send a message///digitalWrite(13,HIGH);

Serial.print("AT+CMGF=1\r"); //Because we want to send the SMS in text modedelay(2000);

Serial.print("AT+CMGS=\"+919390573630\"\r"); //Startsending msgto thisnumber//

//tobesent tothenumber specified.

//Replace this number with the target mobile number.delay(2000);

Serial.print(Data);

Serial.print("\r");//The text for the messagedelay(1000);

Serial.write(0x1A);//Equivalent to sending Ctrl+ZdigitalWrite(13,LOW);

}

///// robot wheel motors ////////void forward(){digitalWrite(m11,LOW);

digitalWrite(m12,HIGH);digitalWrite(m21,LOW);digitalWrite(m22,HIGH);

}

voidbackward()

{

digitalWrite(m11,HIGH);digitalWrite(m12,LOW);digitalWrite(m21,HIGH);digitalWrite(m22,LOW);

}

void left()

{

digitalWrite(m11,HIGH);digitalWrite(m12,LOW);digitalWrite(m21,LOW);digitalWrite(m22,HIGH);

}

void right()

{

digitalWrite(m11,LOW);digitalWrite(m12,HIGH);digitalWrite(m21,HIGH);digitalWrite(m22,LOW);

}

void water\_on(){digitalWrite(m31,LOW);analogWrite(m32,100);

}

void water\_off(){digitalWrite(m31,LOW);digitalWrite(m32,LOW);

}

void up(){

digitalWrite(m41,HIGH);digitalWrite(m42,LOW);delay(50);digitalWrite(m41,HIGH);digitalWrite(m42,HIGH);

}

void down(){digitalWrite(m41,LOW);digitalWrite(m42,HIGH);delay(50);digitalWrite(m41,HIGH);digitalWrite(m42,HIGH);

}

void Break(){digitalWrite(m11, LOW);digitalWrite(m12, LOW);digitalWrite(m21, LOW);digitalWrite(m22, LOW);digitalWrite(m31, LOW);digitalWrite(m32, LOW);digitalWrite(m41, LOW);digitalWrite(m42, LOW);

}

void setup() {Serial.begin(9600);gprs.begin(9600);pinMode(m11, OUTPUT);pinMode(m12, OUTPUT);pinMode(m21, OUTPUT);pinMode(m22, OUTPUT);pinMode(m31, OUTPUT);pinMode(m32, OUTPUT);pinMode(m41, OUTPUT);pinMode(m42,OUTPUT);

pinMode(soil,INPUT);

}

void loop() {if(digitalRead(soil)==LOW){

water\_off();

}

else{water\_on();

SendSMS("soilisdrymotor ison");

}

if(gprs.available()){RecievedData= gprs.read();Serial.println(RecievedData);

if(RecievedData =='f'){forward();

//delay(200);

//Break();

}

else if(RecievedData =='b'){backward();

// delay(200);

//Break();

}

else if(RecievedData =='r'){right();

delay(200);Break();

}

elseif(RecievedData=='l'){

left();delay(200);Break();

}

elseif(RecievedData=='s'){

Break();

}

else if(RecievedData =='p'){water\_on();

delay(2000);

}

else if(RecievedData =='u'){up();

}

else if(RecievedData =='d'){down();

}

}

}