

# Solar Based Multi Tasking Agricultural Robot

<sup>1</sup>Mr. Nigilan M, <sup>2</sup>Mr. Rohithan K, <sup>3</sup>Mr. Nunnaboina Himavanth, <sup>4</sup>Dr. R. Anand , <sup>5</sup>Dr. S .Daison Stallon  
<sup>1,2,3,4,5</sup>Department of EEE, Nehru Institute of Engineering and Technology, Tamilnadu, India

**Abstract**— The solar panel – based multi-tasking robot is a project that involves using Arduino, a motor driver, Bluetooth, and Seed Sowing and grass cutter motors to build an autonomous robot that can perform lawn maintenance tasks. The robot is designed to operate using solar energy, making it eco-friendly and energy – efficient. The robot is designed to perform two tasks: Sowing the Seed and cutting grass. It is equipped with two motors, one for the Seed Sowing and the other for the grass cutter. The motors are connected to the motor driver, which controls their speed and direction. The Arduino board is programmed to receive commands via Bluetooth and control the robot's movements and functions. The program includes code for controlling the motors, receiving Bluetooth commands, and operating the Seed Sowing and grass cutter. The solar panel – based multi – tasking robot is a complex that requires knowledge of electronics, programming, and mechanics. However, it offers a practical solution for automating lawn maintenance tasks while reducing energy consumption and carbon foot print

**Keywords**— solar panel – based multi-tasking robot, Seed Sowing etc.

## I. INTRODUCTION

The solar panel-based multi-tasking robot is an innovative project that involves using advanced technologies such as Arduino, motor driver, Bluetooth, and Seed sowing and grass cutter motors to create a self-sustaining and efficient robot for lawn maintenance tasks. The robot is designed to operate using solar energy, making it eco-friendly and energy-efficient.

The traditional methods of lawn maintenance involve a lot of manual labor and use of fossil fuels. By developing an

Mr. Nigilan, UG Scholar , Department of Electrical and Electronics Engineering , Nehru Institute of Engineering and Technology Coimbatore-641105

Mr. Rohithan K, UG Scholar , Department of Electrical and Electronics Engineering , Nehru Institute of Engineering and Technology Coimbatore-641105

Mr. Nunnaboina Himavanth, UG Scholar , Department of Electrical and Electronics Engineering , Nehru Institute of Engineering and Technology Coimbatore-641105

Dr. R. Anand, Associate Professor , Department of Electrical and Electronics Engineering ,, Nehru Institute of Engineering and Technology Coimbatore-641105

Dr. S .Daison Stallon, Assistant Professor , Department of Electrical and Electronics Engineering ,, Nehru Institute of Engineering and Technology Coimbatore-641105

autonomous robot that can perform these tasks, we can reduce the human effort and energy consumption while simultaneously reducing our carbon footprint. This project aims to create a practical solution for lawn maintenance while promoting sustainability.

The robot is designed to perform two tasks: Seed Sowing and cutting grass. It is equipped with a Seed Sowing motor and a grass cutter motor, both of which are controlled by a motor driver. The robot is operated using an Arduino board, which receives commands via Bluetooth and controls the robot's movements and functions.

The solar panel-based multi-tasking robot offers numerous benefits such as energy efficiency, sustainability, and reduced human effort. This project can also be expanded to incorporate additional functionalities such as sensors for detecting obstacles, weather monitoring, and automatic charging. Overall, this project represents an innovative approach to sustainable lawn maintenance that can significantly reduce our carbon footprint

## A. EMBEDDED INTRODUCTION

An embedded system is a special-purpose computer system, which is completely encapsulated by the device it controls. An embedded system has specific requirements and performs pre-defined tasks, unlike a general-purpose personal computer. An embedded system is a programmed hardware device. A programmable hardware chip is the 'raw material' and it is programmed with particular applications. This is to be understood in comparison to older systems with fully functional hardware or systems with general - purpose hardware and externally loaded software. Embedded systems are a combination of hardware and software which facilitates mass production and variety of application. A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a dedicated function.

EMBEDDED SYSTEM is a combination of SOFTWARE and HARDWARE.

An Embedded system is a system that has a computing device embedded into it. These are the controllers, processors, arrays or other hardware using dedicated (embedded) logic or programming (code) called “firmware” or a “microkernel. Embedded systems are designed around a  $\mu\text{C}$  which integrates Memory & Peripherals.

- ‘Special-purpose’

Embedded systems have a (more or less) well-defined purpose Contrast with general purpose computers (PCs etc.)

- ‘Built into a larger device’

Embedded systems are (usually) part of a larger device, augmenting its capabilities.

### B. REASON FOR EMBEDDED SYSTEMS

It is EMBEDDED because the Micro Controller is ‘inside’ some other system.

For Example, a Micro Controller is ‘EMBEDDED’ into your TV, car, or appliance The consumer need not think about how to make it perform or process

avoids lots of Electronics Components.

- Built in rich Features
- Reduces the cost, space
- Less Down Time for Maintenance
- Probability of Failure is reduced
- Easy interface with Computers

### C. CHARACTERISTICS OF AN EMBEDDED SYSTEM

1. Sophisticated functionality
2. Real-Time Operation
3. Low Manufacturing Cost
  
4. Low Power Consumption
5. Eliminates Necessity of Complex Circuitry
6. Smarter Products
7. Smaller Size
8. User Friendly
9. State of the Art Technology

### D. FOUR GENERAL EMBEDDED SYSTEMS TYPES

#### GENERAL COMPUTING:

• Applications similar to desktop computing, but in an embedded package

#### CONTROL SYSTEMS:

- Closed- loop feedback control of real- time system

#### SIGNAL PROCESSING:

- Computations involving large data streams

#### COMMUNICATION & NETWORKING:

- Switching and information transmission

#### FEATURES OF AN EMBEDDED SYSTEM

##### REAL-TIME OPERATION:

• Reactive: computations must occur in response to external events

- Correctness is partially a function of time

##### SMALL SIZE, LOW WEIGHT:

Hand- held electronics and Transportation applications -- weight costs money

##### LOW POWER

• Battery power for 8+ hours (laptops often last only 2 hours)

##### HARSH ENVIRONMENT

• Heat, vibration, shock, power fluctuations, RF interference, lightning

## II. LITERATURE REVIEW

In recent years, there has been a growing interest in developing autonomous robots for various applications, including lawn maintenance. The solar panel-based multi-tasking robot using Arduino, motor driver, Bluetooth, and Seed Sowing and grass cutter motors is one such example of a sophisticated robot that can perform multiple tasks using solar energy.

In a study conducted by S. S. Sagar and N. V. Thakare, a solar-powered autonomous robot was designed and developed for lawn maintenance. The robot was equipped with a GPS module, a camera, and a robotic arm for performing various tasks. The robot was controlled using a Raspberry Pi board, and a wireless communication module was used for transmitting data. The authors reported that the robot was capable of performing several lawn maintenance tasks, including mowing, trimming, and fertilizing.

In another study conducted by H. T. Dang et al., a robotic lawn mower was developed that utilized a combination of solar and battery power. The robot was equipped with a GPS module, a camera, and a motor driver for controlling the movement of the robot. The authors reported that the robot was capable of mowing a lawn autonomously and could be controlled remotely using a smart phone.

In another study conducted by Megalingam et al. proposed an unmanned robotic coconut tree climber and harvester . A number of grippers have also been proposed for efficient harvesting .Many studies verified the application of robots using simple plants produced in the laboratory; however, trials are beginning to be conducted in field environments where the problem is more complex

## III. EXISTING SYSTEM

There are a few existing systems of solar panel-based multi-tasking robots using Arduino, motor driver, Bluetooth, and Seed Sowing and grass cutter motors available in the market. These systems offer practical solutions for lawn maintenance tasks, reducing human effort and energy consumption while promoting sustainability.

One example of an existing system is the "Sunflower Solar-Powered Robot Lawn Mower" developed by Franklin Robotics. The robot is equipped with a solar panel, a GPS module, and sensors that allow it to navigate through the lawn and avoid obstacles. The robot also features a grass cutter motor that cuts the grass to a desired height. The robot is controlled via Bluetooth using a smart phone app.

Another example is the "Solar-Powered Automatic Lawn Mower" developed by Husqvarna. The robot is equipped with a solar panel, a GPS module, and sensors that allow it to navigate through the lawn and avoid obstacles. The robot also features a grass cutter motor that cuts the grass to a desired height. The robot can be controlled via Bluetooth using a smart phone app.

Both of these systems offer practical solutions for lawn maintenance tasks, reducing human effort and energy consumption while promoting sustainability. However, they

can be expensive and may not be accessible to everyone. The solar panel-based multi-tasking robot using Arduino, motor driver, Bluetooth, and Seed Sowing and grass cutter motors is a more affordable and accessible option that offers similar functionalities.

Overall, the existing systems demonstrate the potential of autonomous robots for lawn maintenance tasks, and the solar panel-based multi-tasking robot represents a promising alternative that can be customized and adapted to specific needs and requirements.

#### IV. PROPOSED SYSTEM

The proposed system of the solar panel-based multi-tasking robot using Arduino, motor driver, Bluetooth, and Seed Sowing and grass cutter motor involves developing a robot that can perform multiple lawn maintenance tasks using solar energy. The system is designed to be affordable, accessible, and customizable, allowing users to adapt the robot to their specific needs and requirements.

The proposed system consists of several components, including:

**Solar Panel:** The robot will be equipped with a solar panel that will generate electricity from the sun. The solar panel will charge the robot's batteries, which will power the motors and other electronic components.

**Arduino Board:** The Arduino board will act as the brain of the robot, controlling the motors, fertilizer spray motor, and other components. It will also receive commands via Bluetooth from a smartphone app or other remote devices.

**Motor Driver:** The motor driver will control the movement of the robot and the speed of the motors. It will also control the fertilizer spray motor and the grass cutter motor.

**Bluetooth Module:** The Bluetooth module will allow the robot to be controlled remotely using a smartphone app or other remote devices. It will receive commands from the user and transmit them to the Arduino board.

**Seed Sowing Motor:** The Seed Sowing motor will be used to sowing the Seeds on the lawn, promoting healthy growth.

**Grass Cutter Motor:** The grass cutter motor will be used to cut the grass to a desired height, reducing the need for manual labor.

**Wifi Module :** Wifi Module are used to send and Receive data over Wifi

The proposed system offers several benefits, including energy efficiency, sustainability, and reduced human effort. It can be customized to specific needs and requirements, making it an accessible solution for lawn maintenance tasks. It also promotes eco-friendliness and reduces the carbon footprint associated with traditional lawn maintenance methods.

Overall, the proposed system of the solar panel-based multi-tasking robot using Arduino, motor driver, Bluetooth, and fertilizer spray and grass cutter motor offers a practical solution for lawn maintenance tasks, making it an innovative approach to sustainable lawn maintenance.

#### ADVANTAGES

- It Reduce the Burden of Farmer

- Saves the Farmers Time
- Cost Effectiveness
- Works in Hazardous Environments

#### BLOCK DIAGRAM OF PROPOSED SYSTEM

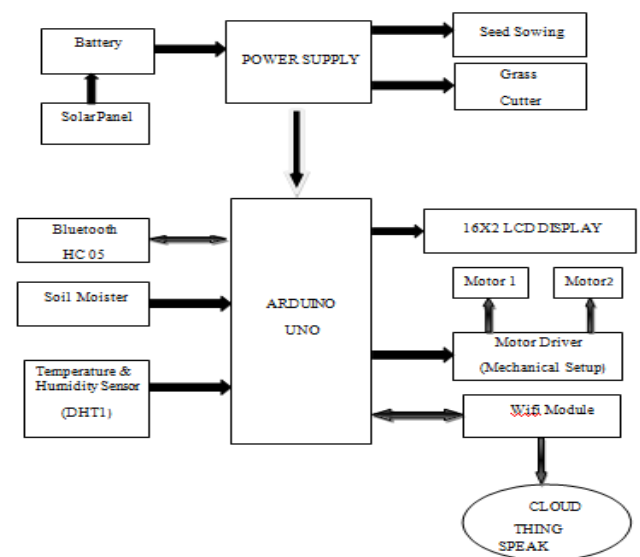


Fig. 1 Proposed Block Diagram

#### V. HARDWARE

##### A. LIST OF HARDWARE

- Arduino Uno
- Solar Panel
- Power Supply
- 16x2 LCD Display
- 12V Battery
- Temperature and Humidity Sensor
- Soil Moisture Sensor
- DC Motor
- Motor Driver
- Bluetooth Module
- Wifi Module

##### B. INTRODUCTION TO ARDUINO

Arduino interface boards provide the engineers, artists, designers, hobbyists and anyone who tinker with technology with a low-cost, easy- to-use technology to create their creative, interactive objects, useful projects etc., A whole new breed of projects can now be built that can be controlled from a computer.

Arduino is an open source electronics prototyping platform based on flexible, Easy - to-use hardware and Software. It's intended for artists, designers, hobbyists, and anyone interested in Creating interactive objects or environments.

It's an open-source physical computing platform based on a microcontroller board, and a development environment for writing software for the board. In simple words, Arduino is a

small microcontroller board with a USB plug to connect to computer and a number of connection sockets that can be wired up to external electronics ,such as motors, relays, light sensors, laser diodes, loudspeakers, microphones, etc., They can either be powered through the USB connection from the computer or from a 9V battery. They can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. what to do by sending a set of instructions to the microcontroller on the board. The Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Iv re Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and

programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IOT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

### C. HISTORY OF ARDUINO

While teaching a physical computing class at the Interaction Design Institute Ivrea in 2005, Massimo Banzi's students were unwilling to spend the 76euros for the BASIC Stamp microcontrollers commonly used in such applications. Banzi and his colleagues looked for alternatives, finally settling on the wiring platform developed by one of Banzi's students.

In his own words: we started to figure out how could we make the whole platform even simpler, even cheaper, even easier to use. And then we started to essentially re implement the whole thing as an open source project.

Once they had a prototype, a student wrote the software that would allow wiring programs to reunion the new platform. Upon seeing the project, visiting professor Casey Reas suggested that there might be wider applications than just design schools for the new product. The prototype was redesigned for mass production and a test run of 200 boards was made. Orders began coming in from other design schools and the students looking for Arduinos

The Arduino project was born and Massimo Banzi and David Cuartielles became its founders. ARDUINO is an Italian word, meaning STRONG FRIEND. The English version of the name is Hardwin. As of May2011, more than 300,000 Arduino units are in the wild.

### D. ARDUINO BOARD LAYOUT

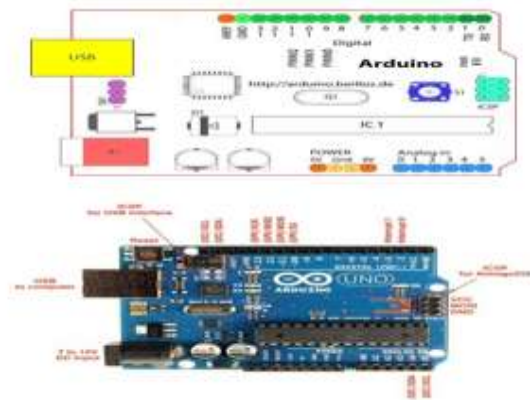


Fig. 2 ARDUINO Pin Diagram

### E. FEATURES:

- High-performance, Low-power AVR
- 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 8K Bytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1K Byte Internal SRAM

- Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
- Data retention: 20 years at 85°C/100 years at 25°C (1)
- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
- Programming Lock for Software Security
- Peripheral Features
- Two 8-bit Timer/Counters with Separate Prescaler, one compare
- Six Channels 10-bit Accuracy
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip
- Oscillator
- On-chip Analog Comparator
- Special Microcontroller Features
- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Five Sleep Modes: Idle, ADC Noise Reduction, Power-save
- I/O and Packages
- 23 Programmable I/O Lines
- 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
- 2.7 - 5.5V (ATmega8L)
- 4.5 - 5.5V (ATmega8)
- Power Consumption at 4 Mhz, 3V, 25°C

- Active: 3.6 mA
- Idle Mode: 1.0 mA
- Power-down Mode: 0.5 µA

F. ARCHITECTURE DIAGRAM

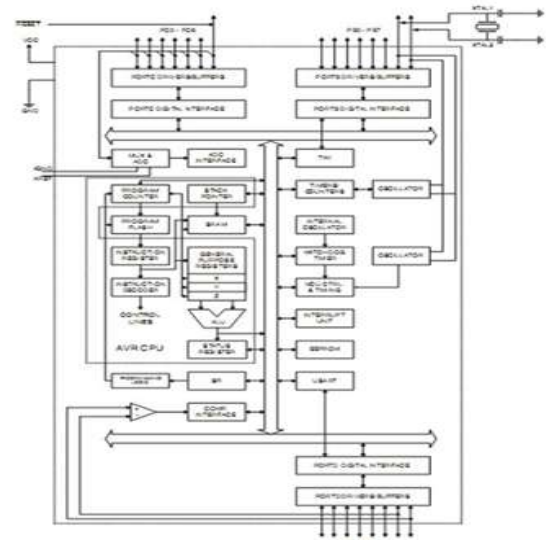


Fig. 3 Architecture diagram for arduino

G. PIN CONFIGURATIONS



Fig. 4 Pin Configuration

The ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1 MIPS per MHz, allowing the system designed to optimize power consumption versus processing speed.

Pin Descriptions:

VCC :-Digital supply voltage.

GND:- Ground.

DIFFERENT UNITS

Port B(PC7..PB0) :- is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive

characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active,even if the clock is not running.

Port C (PC5..PC0) :- Port C is an 7-bit bi-directional I/O port with internal pull- up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D (PD7..PD0) :- Port D is an 8-bit bi-directional I/O port with internal pull- up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active,even if the clock is not running.

RESET (Reset input):- A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The Shorter pulses are not guaranteed to generate a reset.

AVCC:- AVCC is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

AREF:- AREF is the analog reference pin for the A/D Converter.These pins are powered from the analog supply and serve as 10-bit ADC channels.

Basic Function:- The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories,perform calculations, control peripherals, and handle interrupts.

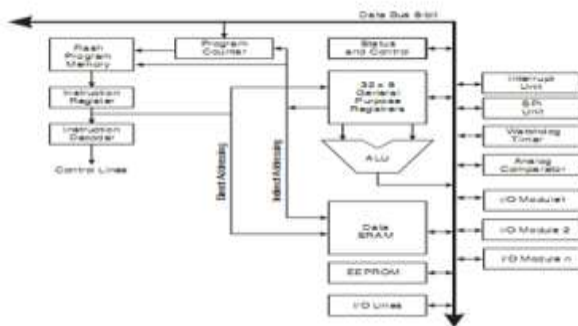


Fig. 5 : INTERFACING OF DATA BUS WITH

H. LEAD ACID BATTERIES

Lead acid batteries are the most common large-capacity rechargeable batteries. They are very popular because they are dependable and inexpensive on a cost-per watt base. There are few other batteries that deliver bulk power as cheaply as lead acid, and this makes the battery cost-effective for automobiles, electrical vehicles, forklifts, marine and uninterruptible power supplies (UPS).

Lead acid batteries are built with a number of individual cells containing layers of lead alloy plates immersed in an electrolyte solution, typically made of 35% sulphuric acid (H2SO4) and 65% water. Pure lead (Pb) is too soft and would not support itself, so small quantities of other metals are added to get the mechanical strength and improve electrical properties. The most common additives are antimony (Sb), calcium (Ca), tin (Sn) and selenium (Se). When the sulphuric acid comes into contact with the lead plate, a chemical reaction is occurring and energy is produced.

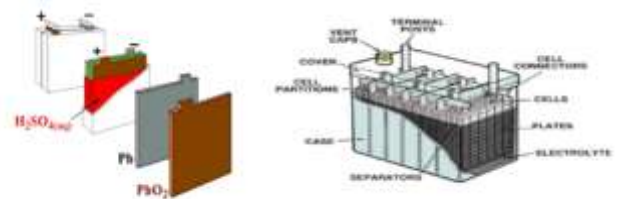


Fig. 6 Lead Acid Battery

Lead acid batteries are heavy and less durable than nickel (Ni) and lithium (Li) based systems when deep cycled or discharged (using most of their capacity). Lead acid batteries have a moderate life span and the charge retention is best among rechargeable batteries. The lead acid battery works well at cold temperatures and is superior to lithium-ion when operating in sub-zero conditions. Lead acid batteries can be divided into two main classes: vented lead acid batteries (spillable) and valve regulated lead acid (VRLA) batteries (sealed or non-spillable).zTypicalvented lead acid battery schematic

□ Lead acid batteries are usually filled with an electrolyte solution containing sulphuric acid. This is a very corrosive chemical (pH<2) which can permanently damage the eyes and produce serious chemical burns to the skin. Sulphuric acid is also poisonous, if swallowed. The lead alloys found in batteries are also harmful to humans and can also seriously damage the environment.

□ Wear the proper personal protective equipment (PPE), specifically splash- proof goggles, acid- resistant lab coat or apron, safety shoes and rubber gloves. A face shield must also be worn when refilling batteries with electrolytes.

□ Know where the emergency showers and emergency eyewash stations are located; they must be located near lead acid battery storage and charging areas.

- Slowly pour concentrated acid into water; do not add water to acid. (warning: electrolyte will become hot; do not close battery vents until electrolyte has cooled down)
- Use non-metallic containers and funnels.
- Ensure neutralizers (e.g. baking soda) are available for immediate use.
- Use extreme care to avoid spilling or splashing the sulphuric acid solution

**Table 1 Electrolyte Spill**

<b>Health Risks(WH MIS2015)</b>	
<b>color</b>	clear
<b>odor</b>	sharp,pungent
<b>pH</b>	1to2
<b>Boilingpoint</b>	95-115°C
<b>LC<sub>50</sub>(rat)</b>	375mg/m <sup>3</sup>
<b>LD<sub>50</sub>(oral,rat)</b>	2140mg/kg

**Health Risks(WH MIS2015)**

color clear  
 odor sharp,pungent  
 pH 1to2  
 Boilingpoint 95-115°C  
 LC50(rat)375mg/m3  
 LD50(oral,rat) 2140mg/kg

In the event of a minor electrolyte spill, consult the appropriate Safety Data Sheet (SDS) for electrolyte spill containment, clean-up and disposal details. Always ensure to wear adequate protective clothing (goggles, closed shoes and gloves) during clean-up of spills.

In case of a small electrolyte spilled :

- Contain the spill with absorbents such as universal pads ,hazmat pads , sand ,earth or vermiculite.
- Remove the absorbents once it has soaked up the acid/electrolyte.
- Clean up spilled acid safely with an acid neutralizer and then with large volumes of water to rinse the area.

In the event of a large acid electrolyte spill:

1. Advise and warn co-workers.
2. Advise and warn co Evacuate the area immediately.
3. Restrict the access to the area.

**I. SOLAR PANEL**

A solar panel is a set of solar photovoltaic modules electrically connected and mounted on a supporting structure. A photovoltaic module is a packaged connected assembly of solar cells. The solar panel can be used as a component of a larger photovoltaic system to generate and supply electricity in 32 commercial and residential applications.

A photovoltaic system typically includes a panel or an array of solar modules, an inverter, and sometimes a battery

and/or solar tracker and interconnection wiring. Photovoltaic cells or panels are only one way of generating electricity from solar energy. They are not the most efficient, but they are the most convents to use on a small to medium scale.

PV cells are made of silicon, similar to that used in computer "chips". While silicon itself is a very abundant mineral, the manufacture of solar cells (as with computer chips) has to be in a very clean environment. This causes production costs to be high. A PV cell is constructed from two types of silicon, which when hit by solar energy, produce a voltage difference 42across them, and, if connected to an electrical circuit, a current will flow.

A number of photovoltaic cells will be connected together in a "Module", and usually encapsulated in glass held a frame which can then be mounted as required. The cells in a module will be wired in series or parallel to produce a specified voltage. What may be referred to as a 12-volt panel may produce around 16 volts in full sun to charge to 12-volt battery. Here we use OSWAL company solar panel. The mechanical characteristics made from high efficiency crystalline silicon solar cells. Cells encapsulated in low iron, high transmission, toughened glass using UV stable ethylene vinyl acetate (EVA) sheets. Premium quality back sheet protects the module from environmental conditions. Laminate framed with strong anodized aluminium profile with fitted junction box.

Specification of the solar panel:

1. Material: Silicon
2. Rated peak power (Pmax): 15W 33
3. Type: Polycrystalline
4. No of Cells: 32
5. Rated Voltage (Vpm): 18V
6. Rated Current (Ipm): 0.85A
7. voltage at maximum power: 17.5 V
8. Current at max. Power: 0.58 A
9. Tolerance: 3%

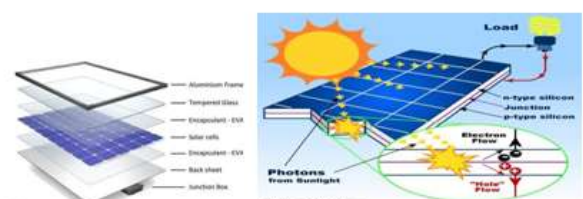


Fig. 7 Solar panel

J. POWER SUPPLY : UNIT DIAGRAM

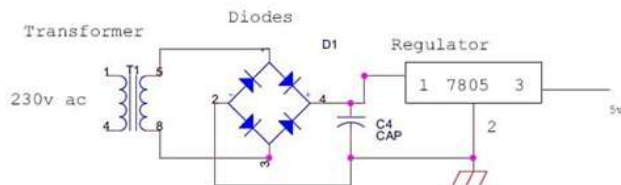


Fig. 8 Unit Diagram

K. REGULATOR:

Regulator regulates the output voltage to be always constant. Regulator having two types.

- Positive regulator (78XX)
- Negative regulator (79XX)

The output voltage is maintained irrespective of the fluctuations in the input AC voltage. As and then the AC voltage changes, the DC voltage also changes. Thus to avoid this Regulators are used. Also when the internal resistance of the power supply is greater than 30 ohms, the output gets affected. Thus this can be successfully reduced here. The regulators are mainly classified for low voltage and for high voltage. Here we used 7805 positive regulator. It reduces the 12V dc voltage to 5V dc.

The Filter circuit is often fixed after the Regulator circuit. Capacitor is most often used as filter. The principle of the capacitor is to charge and discharge. It charges during the positive half cycle of the AC voltage and discharges during the negative half cycle. So it allows only AC voltage and does not allow the DC voltage. This filter is fixed after the Regulator circuit to filter any of the possibly found ripples in the output received finally. Here we used 0.1µF capacitor. The output at this stage is 5V and is given to the Microcontroller In the power supply circuit two regulators are used. 7805 regulator is used to produce positive 5V dc. Microcontroller and sensors are operated at 5V dc voltage. The output of the 7805 regulator is connected to Arduino Nano microcontroller.

L. LCD DISPLAY INTRODUCTION

The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. In this tutorial, we will discuss about character based LCDs, their interfacing with various micro controllers, various interfaces (8-bit/4-bit), programming, special stuff and tricks .

PIN DESCRIPTION

The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, where as

LCDs supporting more than 80 characters make use of

2 HD44780 controllers . Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back -light LED connections). Pin description is Shown.

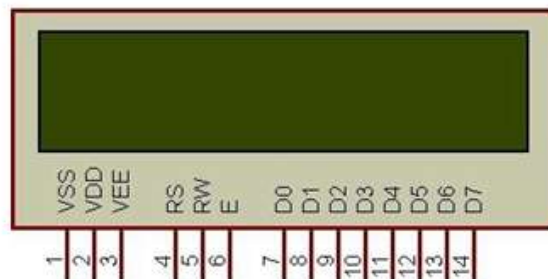


Fig. 9 Character LCD type HD44780 Pin diagram (16\*2)

Temperature & Humidity Sensor (DHT11):

DHT11 Module features a temperature & humidity sensor complex with a calibrated digital signal output. The exclusive digital-signal-acquisition technique and temperature & humidity sensing technology ensure high reliability and excellent long-term stability. This sensor includes an NTC for temperature measurement and a resistive-type humidity measurement component for humidity measurement. These are connected to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability, and cost-effectiveness.

DHT11 Module Pinout

The DHT11 module has a total of 3 pins. In which two are for power and one is for communication. The pinout of a DHT11 Sensor module is as follows:

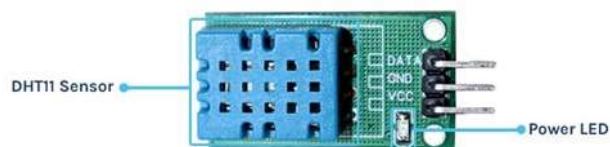


Fig. 10 DHT1 Sensor DHT11 Module Circuit Diagram

The schematic diagram for the DHT11 module is given below. As mentioned earlier, the board has a very low components count. The VCC and GND are directly connected to the DHT11 and a pullup resistor is added to the DATA pin. Sufficient filtering is provided with the tantalum and multilayer capacitors. An LED with a current limit resistor is used as a power indicator.



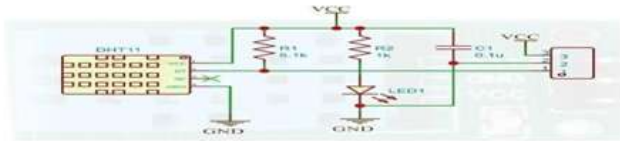


Fig. 11 Circuit Diagram

The Humidity sensing component consists of a moisture-holding substrate sandwiched in between two electrodes. When the substrate absorbs water content, the resistance between the two electrodes decreases. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes. This change in resistance is measured with the onboard MCU's ADC and the relative humidity is calculated.

Each DHT11 element is strictly calibrated in the laboratory which is extremely accurate in humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process.

**SOIL MOISTURE SENSOR :**

The Soil moisture sensor has four pins VCC, GND, A out, D out. These four pins can be used to get the soil moisture data from the sensor, The pinout of the Soil Moisture Sensor are as follows:

The working of the soil moisture sensor is pretty simple and straightforward, as in the image below. We just need to stick the fork-shaped conductive probe to the soil, as the probe has two exposed conductive plates that will act as a variable resistor whose resistance will vary depending on the water content in the soil.

This resistance of the probe is inversely proportional to the soil moisture of the device. The more water in the soil the better the conductivity which will result in lower resistance. The less the water in the soil the poor the conductivity which means higher resistance. This sensor produces an output voltage according to the resistance by measuring which we can determine the moisture level.

The entire soil moisture sensor consists of two parts: the first one is the soil moisture sensor probe and the second one is an electronic module. The module processes the incoming data from the probe and that gets processed by a microcontroller like Arduino and we get the final output.

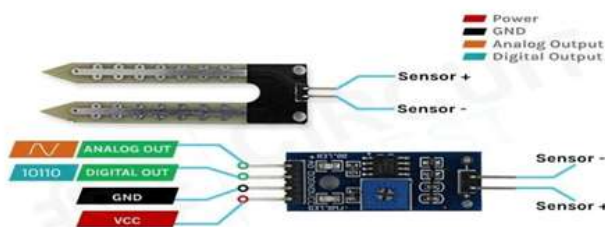


Fig. 12 Soil Moisture sensor

**M. DC MOTOR:**

Interface different types of motors with the Arduino board (UNO) and to connect the motor and drive it from board.

There are three different type of motors –

- DC motor
- Servo motor
- Stepper motor

A DC motor (Direct Current motor) is the most common type of motor. DC motors normally have just two leads, one positive and one negative. Connect these two leads directly to a battery, the motor will rotate. Switch the leads, the motor will rotate in the opposite direction.

Do not drive the motor directly from Arduino board pins. This may damage the board. Use a driver Circuit or an IC.

We will divide this chapter into three parts –

- Just make motor spin
- Control motor speed
- Control the direction of the spin of DC motor

Components Required 1x Arduino UNO board 1x PN2222 Transistor 1x Small 6V DC Motor

1x 1N4001 diode and 1x 270 Ω Resistor

Precautions

Take the following precautions while making the connections.

First, make sure that the transistor is connected in the right way. The flat side of the transistor should face the Arduino board as shown in the arrangement.

Second, the striped end of the diode should be towards the +5V power line according to the arrangement shown in the image.

The transistor acts like a switch, controlling the power of the motor.

Arduino pin 3 is used to turn the transistor on and off and is given the name 'motor Pin' in the sketch.

When the program starts, it prompts to give the values to control the speed of the motor. Enter a value between 0 and 255 in the Serial Monitor.

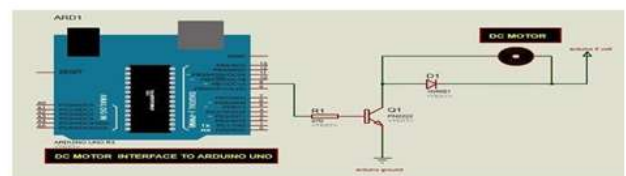


Fig. 13 DC Motor Interface To Arduino UNO

In the 'loop' function, the command 'Serial .parse Int' is used to read the number entered as text in the Serial Monitor and

convert it into an 'int'. Type any number here. The 'if' statement in the next line simply does an analog write with this number, if the number is between 0 and 255.

**N. Spin direction control**

To control the direction of the spin of DC motor, without inter changing the leads, use a circuit called an H-Bridge. An H-bridge is an electronic circuit that can drive the motor in both directions. H-bridges are used in many different applications. One of the most common application is to control motors in robots. It is called an H-bridge because it uses four transistors connected in such a way that the schematic diagram looks like an "H."

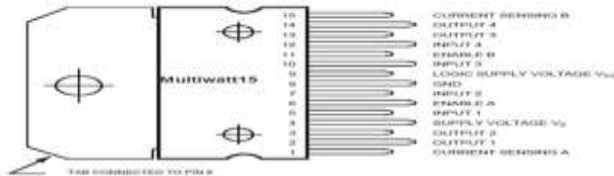
We will be using the L298 H-Bridge IC here. The L298 can control the speed and direction of DC motors and stepper motors, and can control two motors simultaneously. Its current rating is 2A for each motor.

```
0 0 BRAKE

1 0 FORWARD

0 1 BACKWARD

1 1 BRAKE
```



**Fig. 14 Spin Direction**

IN1 pin of the L298 IC is connected to pin 8 of the Arduino while IN2 is connected to pin 9. These two digital pins of Arduino control the direction of the motor. The EN A pin of IC is connected to the PWM pin 2 of Arduino. This will control the speed of the motor.

To set the values of Arduino pins 8 and 9, we will use the digitalWrite() function, and to set the value of pin 2, we will use the analogWrite() function.

**Arduino Motor Control Setup**

- Connect 5V and ground of the IC to 5V and ground of Arduino.
- Connect the motor to pins 2 and 3 of the IC.
- Connect IN1 of the IC to pin 8 of Arduino.
- Connect IN2 of the IC to pin 9 of Arduino.
- Connect EN1 of IC to pin 2 of Arduino.
- Connect SENS A pin of IC to the ground.
- Connect the Arduino using Arduino USB cable and upload the program to the Arduino using Arduino IDE software or

Arduino Web Editor.

Provide power to the Arduino board using power supply, battery or USB cable the motor should now run first in the clockwise (CW) direction for 3 seconds and then counter-clockwise (CCW) for 3 seconds.

**: WI FI MODULE**

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to Wi-Fi network. The ESP8266 is capable of either hosting an application or off loading all Wi-Fi networking functions from another application processor.

Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning. The ESP8266 module is an extremely cost-effective board with a huge, and ever growing, community.

This module has a powerful enough on – board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up – front and minimal loading during runtime.



**Fig. 15 Wifi Module**

**BLUETOOTH MODULE (HC 05)**

HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.



**Fig. 16 Bluetooth Module**

**HC 05 PIN DIAGRAM**

This Picture Shows the back side of Bluetooth Module HC-05 and Pin Details HC-05 Bluetooth Module Pin Diagram



**Fig. 17 HC 05 Pin Diagram**

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.

It has 6 pins,

1. Key/EN: It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.

HC-05 module has two modes,

1. Data mode: Exchange of data between devices.
2. Command mode: It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.
2. VCC: Connect 5 V or 3.3 V to this Pin.
3. GND: Ground Pin of module.
4. TXD: Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)
5. RXD: Receive data serially (received data will be transmitted wirelessly by Bluetooth module).
6. State: It tells whether module is connected or not.

## VI. SOFTWARE

### A. LIST OF SOFTWARES

- Embedded c
- Arduino IDE Compiler
- Proteus

### B. ARDUINO IDE

IDE stands for “Integrated Development Environment” :It is an official software introduced by Arduino.cc, that is mainly used for editing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go. In this article, we will introduce the Software, how we can install it, and make it ready for developing applications using Arduino modules.

Arduino IDE Definition

1. Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
2. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

3. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.

4. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.

5. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.

6. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

7. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

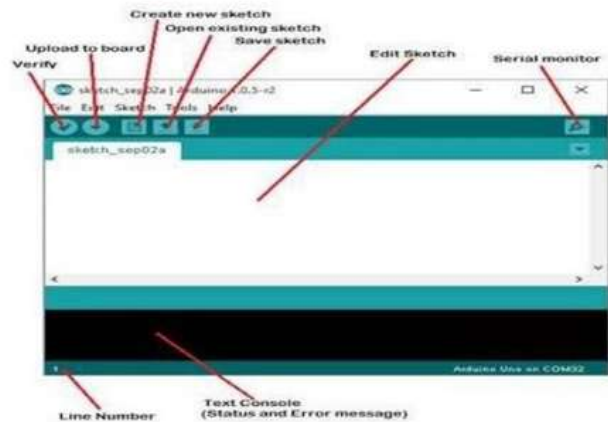
8. This environment supports both C and C++ languages.

How to get Arduino IDE:

Download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MACOs , we select to download the correct software version that is easily compatible with our operating system.

Details on IDE: The IDE environment is mainly distributed into three sections

1. Menu Bar
2. Text Editor
3. Output Pane



**Fig. 18 IDE Display**

The bar appearing on the top is called Menu Bar that comes with five different options as follow:

- File –Window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.
- Menu Bar
- Text Editor
- Output Pane

- Edit – Used for copying and pasting the code with further modification for font Sketch – For compiling and programming
- Tools – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
- Help – In case feeling skeptical about software, complete help is available from getting started to troubleshooting. The Six Buttons appearing under the Menu tab are connected with the running program as follow.
- The Six Buttons appearing under the Menu tab are connected with the running program as follow. A separate pop up window appears as an independent terminal and plays a vital role for sending and receiving the Serial Data the Tools panel and select Serial Monitor pressing Ctrl+Shift+M all at once will open the Serial Monitor.
- The Serial Monitor will actually help to debug the written Sketches where can get a hold of how program is operating. Arduino Module should be connected to computer by USB cable in order to activate the Serial Monitor.
- Need to select the baud rate of the Arduino Board you are using right now.

Program Structure:

- Variables

Whenever using Arduino, need to declare global variables and instances to be used later on. In a nutshell, a variable allows to name and store a value to be used in the future. For example, would store data acquired from a sensor in order to use it later. To declare a variable simply define its type, name and initial value. It's worth mentioning that declaring global variables isn't an absolute necessity. However, it's advisable that declare your variables to make it easy to utilize your values further down the line.

- Instances

In software programming, a class is a collection of functions and variables that are kept together in one place. Each class has a special function known as a constructor, which is used to create an instance of the class. In order to use the functions of the class, we need to declare an instance for it.

- Setup()

Every Arduino sketch must have a setup function. This function defines the initial state of the Arduino upon boot and runs only once. Here we'll define the following:

1. Pin functionality using the pinMode function
2. Initial state of pins
3. Initialize classes
4. Initialize variables
5. Code logic Loop()

The loop function is also a must for every Arduino sketch and executes once setup() is complete. It is the main function and as its name hints, it runs in a loop over and over again.

### C. THING SPEAK IOT SERVER

- Thing Speak is an open data platform for the Internet of Things. Device or application can communicate with Thing Speak using a RESTful API, and can either keep your data

private, or make it public. In addition, use Thing Speak to analyze and act on data. Thing Speak provides an online text editor to perform data analysis and visualization using MATLAB®. Can also perform actions such as running regularly scheduled MATLAB code or sending a tweet when your data passes a defined threshold. Thing Speak is used for diverse applications ranging from weather data collection and analysis, to synchronizing the colour of lights across the world.

- At the heart of Thing Speak is a time-series database. Thing Speak provides users with free time-series data storage in channels. Each channel can include up to eight data fields. This tutorial provides an introduction to some of the applications of Thing Speak, a conceptual overview of how Thing Speak stores time-series data, and how MATLAB analysis is incorporated in Thing Speak.

Connect SP8266 to Thing Speak

Makers and hobbyists think of Internet of Things (IoT), two things come to their mind: one is ESP8266 and the other is Thing Speak. One is the Hardware part of the IoT System whereas the other provides the necessary API (or the user interface) for the system. In this project, I'll show you how to Connect ESP8266 to Thing Speak Application and how the ESP8266 Thing Speak collaboration works.

### D. TABLE OF CONTENTS

- Overview
- Creating ThingSpeak Account
- Prerequisites for the Project
- Circuit Diagram for ESP8266 ThingSpeak Interface through AT Commands
- Components Required
- Connect ESP8266 to ThingSpeak using AT Commands
- Circuit Diagram for ESP8266 ThingSpeak Interface using Arduino
- Connect ESP8266 to ThingSpeak using Arduino
- Code
- Conclusion & Applications

### OVERVIEW

IoT implementation using ESP8266 and a REST Platform. What makes ThingSpeak different and special is that it uses simple HTTP Protocol to transfer, store and retrieve information from different sensors.

Also, the ThingSpeak Application allows us to log the sensor data, track locations and even social networking of things.

Another important thing (or rather a unique feature) about ThingSpeak is its support from MATLAB. The close relationship between ThingSpeak and MATLAB has led to integrate several key features of MATLAB into the ThingSpeak Application.

One such feature is to analyse and visualize the user data i.e. the sensor data in a graphical way without the MATLAB License.

Keeping the corporate stuff aside, the ThingSpeak

Application is a great tool for our IoT related projects and hence this project focuses on the basics i.e. how to connect ESP8266 to ThingSpeak Application and also how the ESP8266 ThingSpeak Interface can be used in our future projects.

#### CREATING THINGSPEAK ACCOUNT :

The first thing need to do is to create an account with ThingSpeak. Since the collaboration with MATLAB, can use your MathWorks credentials to login to ThingSpeak using the Sign In link from this page: THINGSPEAK

If do not have one, need to create an account with MathWorks and login to ThingSpeak Application.

NOTE: The MathWorks Account can be used for both MATLAB as well as ThingSpeak logins.

After logging in, need to create a new channel for the data to be stored. For this go to Channels->My Channels and click on New Channel.



Fig 19: ThingSpeak Channel Display

Enter the name of the channel and name of Field 1 in the corresponding sections. Fields in a channel are used to hold the data and each channel can have up to 8 fields. After entering the details save the channel.

In my case, I've created a Channel called "Test Channel" and the Field 1 as "Random Number".

There are a few other things need to do in the ThingSpeak Application

The next step is to prepare the hardware for the project, which includes ESP8266 WiFi Module, Arduino UNO Board and a few connecting wires.

There are two ways of how to connect ESP8266 to ThingSpeak Application. For both the ways, need to make sure that your ESP8266 Module is loaded or flashed with AT Commands Firmware.

For more information on this, HOW TO UPDATE FLASH ESP8266 FIRMWARE.

NOTE:

For flashing the AT Commands firmware, need to enable the programming mode in ESP8266 by connecting GPIO0 to GND and resetting the module.

But in this circuit (assuming have already flashed the firmware), the ESP Module is in Normal Mode i.e. GPIO0 can be left floating.

After flashing your ESP with AT Commands Firmware, can now proceed with connecting ESP8266 to ThingSpeak. As I said before, can do this in two ways: One is through AT Commands and the other way is through Arduino (even this way uses AT Commands but Arduino controls them).

There are both ways to connect ESP8266 to ThingSpeak.

The circuit diagram in order to connect ESP8266 to ThingSpeak is very simple. In fact, might have already seen this connection before. The Arduino UNO Board is used just to transmit the data between the computer and the ESP8266 i.e. it acts as an USB-to-UART Converter.

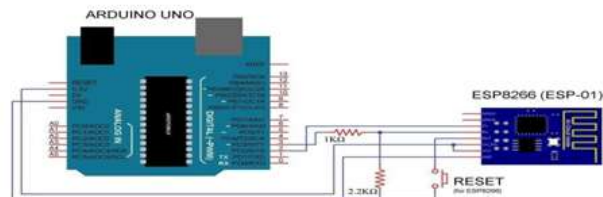


Fig. 20 Circuit Diagram for ESP8266

#### COMPONENTS REQUIRED:

- Any Arduino Board or USB-to-UART Converter
- ESP8266 WiFi Module (ESP-01)

#### CONNECT ESP8266 TO THINGSPEAK USING AT COMMANDS

Connect the Arduino board to the computer and open the serial monitor of Arduino and check for connectivity using the following command.

Note : that I have set the baud rate to 115200 and also selected "Both NL & CR" option in the Serial Monitor. After get a response as "OK", can proceed with connecting your ESP Module to WiFi Network using the following command.

Replace SSID with the name of WiFi Network and enter the password in place of PASSWORD. You will now get a confirmation response regarding WiFi Connection as follows.



Fig. 21 AT Commands of Arduino

#### AT Commands of Wifi module testing

Now, need to set single connection using the following command.

Next step is to connect to the ThingSpeak API using TCP Protocol. For this, need to use the following command.

Alternatively, can use the IP Address of the host api.thingspeak.com i.e. 184.106.153.149.

NOTE:

After starting the TCP connection, if you don't perform any action, the connection will be closed automatically after some time, usually after 1 minute.

Now, successfully enabled the "TCP" connection between

the ESP8266 and ThingSpeak. Next, send any data through this TCP Connection.

For this, use the following commands one after the other. will understand the process .



Fig. 22 Thingspeak API Key Display

### THINGSPEAK WRITE API KEY

Before sending the data, need to acquire the API Key. For this, go to channel (the one have just created) and click on “API Keys” Tab. Below that, can find Write API Key, which is an alphanumeric string of 16 characters. Make a note of this key.

Now, use the following command to initialize the data transmission.

The value 51 is the length of the data to be transmitted. This includes the complete data including the API Key and the “\r” and “\n” values. For this command, will get the following response.

Now type the following information and hit send. Here, “XXXXXXXXXXXXXXXXXXXX” is nothing but the 16 character Write API Key, which have just copied. And the number “143” is the actual data are transmitting to field1.

After typing this text and hitting on send, will not get any response. It is actually waiting for the close command. Once hit send for the above text, immediately type the following command and hit send.

The moment you hit send, will get the following response (not the exact one, but something similar).

### DEBUGGING OF WIFI MODULE USING SERIAL MONITOR

Here, the number 5 means, it is my 5TH message to that Key. Now, Open the Thing Speak API and open your channel. In the “Private View” tab, can see the value ‘143’ in the Field 1 Chart.

That’s it. If understood and followed all these steps, then might have successfully connected ESP8266 to ThingSpeak API. To send more data, repeat the steps from creating the TCP Connection.



Fig. 23 Final output of Thingspeak server

Are you tired of typing all the AT Commands manually?

Obviously, the answer will be yes. Here comes Arduino to the rescue. Make all the connections as per the above circuit diagram and we will proceed with the code

## VII. RESULT AND DISCUSSION

The agricultural robot project utilizing Arduino, Bluetooth, DHT11 temperature and humidity sensor, soil moisture sensor, and a motor driver was successfully implemented and tested. The following are the results and discussion of the project:

**Autonomous Operation:** The robot demonstrated autonomous operation and performed agricultural tasks without the need for constant human intervention. The Arduino board served as the brain of the robot, controlling its movements and functions based on the sensor readings.

**Sensor Readings:** The DHT11 sensor accurately measured temperature and humidity levels in the agricultural environment. This data provided valuable insights into the conditions necessary for optimal plant growth and health. The soil moisture sensor effectively determined the moisture content in the soil, enabling efficient irrigation and preventing overwatering or under watering.

**Motor Control:** The motor driver controlled the movement of the robot and the operation of various agricultural tools, such as plows or seed dispensers. The motor driver allowed precise control over the speed and direction of the motors, ensuring accurate and efficient execution of agricultural tasks.

**Bluetooth Control:** The integration of Bluetooth technology enabled wireless control of the agricultural robot. Commands could be sent from a smartphone or computer to the Arduino board, providing remote operation and control. This feature offered convenience and flexibility to the user.

**Plant Health Monitoring:** By combining the data from the DHT11 sensor and the soil moisture sensor, the agricultural robot could assess the health and moisture needs of the plants. This information allowed for timely interventions such as irrigation or adjusting environmental conditions to promote optimal plant growth.

**Efficiency and Resource Conservation:** The use of sensors and automation improved the efficiency of agricultural operations. The robot could accurately monitor and respond to environmental conditions, leading to optimized resource utilization, reduced water consumption, and minimized manual labor.

**Scalability and Customizability:** The project demonstrated the potential for scalability and customization. Additional sensors or tools could be incorporated into the system to cater to specific agricultural needs, such as pH sensors or pesticide dispensers. The modular design of the system allowed for easy expansion and adaptation.

In conclusion, the agricultural robot using Arduino, Bluetooth, DHT11 sensor, soil moisture sensor, and motor driver proved to be a successful project. The robot showcased the ability to autonomously monitor environmental conditions,

perform agricultural tasks, and conserve resources. This project lays the foundation for the development of advanced and efficient agricultural automation systems.

### VIII. CONCLUSION

In conclusion, the solar panel-based multi-tasking robot using Arduino, motor driver, Bluetooth, and Seed Sowing and grass cutter motor is an innovative and practical solution for lawn maintenance tasks. It is a cost-effective, sustainable, and eco-friendly alternative to traditional lawn maintenance methods, reducing human effort and promoting energy efficiency.

The proposed system offers several benefits, including accessibility, customizability, and ease of use. It can be controlled remotely using a smartphone app or other remote devices, making it convenient for users. It also promotes eco-friendliness and reduces the carbon footprint associated with traditional lawn maintenance methods.

Although there are existing systems available in the market, the solar panel-based multi-tasking robot offers a more affordable and accessible option that can be customized and adapted to specific needs and requirements. The development of this system represents an innovative approach to sustainable lawn maintenance, promoting energy efficiency and reducing the environmental impact of traditional lawn maintenance methods.

Overall, the solar panel-based multi-tasking robot using Arduino, motor driver, Bluetooth, and Seed Sowing and grass cutter motor is a promising solution for lawn maintenance tasks, with the potential to revolutionize the industry and promote sustainable practices.

### IX. REFERENCES

1. N. L. Narasimha, K. L. Chaitanya, K. R. K. Murthy, "Design and Implementation of a Solar Powered Lawn Mower Robot with Bluetooth Control," *International Journal of Engineering Research & Technology*, vol. 4, no. 10, pp. 788-791, 2015.
2. S. S. Swain, S. R. Sahoo, S. K. Mishra, "Design and Development of Solar Powered Agricultural Robot for Precision Farming," *International Journal of Engineering Trends and Technology*, vol. 58, no. 4, pp. 163-169, 2018.
3. H. K. Lee, K. W. Park, H. K. Lee, "Design and Implementation of a Smart Lawn Mower Robot Based on Internet of Things (IoT) Technology," *Sensors*, vol. 18, no. 3, pp. 715, 2018.
4. Y. W. Chang, H. C. Cheng, C. Y. Liu, "Development of a Solar-Powered Lawn Mower Robot with Intelligent Navigation System," *International Journal of Advanced Robotic Systems*, vol. 14, no. 1, pp. 1-12, 2017.
5. M. Singh, R. Kumar, "A Review on Autonomous Solar Powered Lawn Mower Robot," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3, no. 3, pp. 740-746, 2018.

6. Y. K. Lee, K. J. Song, H. S. Lee, "Development of a Solar-Powered Lawn Mower Robot with a Variable Speed Control System," *Energies*, vol. 12, no. 12, pp. 2322, 2019.
7. S. K. Singh, P. Kumar, "Design and Fabrication of Solar Powered Lawn Mower Robot," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 4, no. 2, pp. 6-10, 2018.
8. R. Kumar, A. Vashisht, "A Review on Solar-Powered Autonomous Lawn Mower Robot," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 4, no. 2, pp. 852-858, 2019.
9. R. Goyal, N. Kumar, "Design and Fabrication of Solar-Powered Lawn Mower Robot," *International Journal of Mechanical Engineering and Technology*, vol. 8, no. 5, pp. 1-9, 2017.
10. S. S. Swain, S. K. Mishra, S. R. Sahoo, "Design and Development of Solar Powered Robot for Agricultural Applications," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 7, no. 9, pp. 17413-17420, 2018.