SOLAR POWERED SMART IRRIGATION SYSTEM

MAJOR PROJECT REPORT

SUBMITTED TO JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD IN PARTIAL FULFILMENT FOR THE AWARD OF BACHELOR OF TECHNOLOGY IN CIVIL ENGINEERING

BY

17211A0104 – ALLE VENNALA

17211A0119 – BONGU PRANATHI

17211A0120 – CHANDRIKA SRAVYA

17211A0138 – EDUPALLI LALITHYA

UNDER THE SUPERVISION OF

Dr. G. SRIDEVI

PROFESSOR & HEAD OF THE DEPARTMENT

DEPARTMENT OF CIVIL ENGINEEERING
B.V. RAJU INSTITUTE OF TECHNOLOGY
(UGC–AUTONOMOUS)
VISHNUPUR, NARSAPUR, MEDAK DISTRICT, TELANGANA STATE

2021

DECLARATION

We hereby declare that the work presented in this project work entitled **SOLAR POWERED SMART IRRIGATION SYSTEM** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of engineering ethics. It contains no material previously published or written by another person nor materials which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Place: Narsapur
Date:

(A. VENNALA)
Roll No. 17211A0104

(B. PRANATHI)
Roll No. 17211A0119

(C. SRAVYA)
Roll No. 17211A0120

(E. LALITHYA)
Roll No. 17211A0138

B.V.RAJU INSTITUTE OF TECHNOLOGY

(UGC–AUTONOMOUS)

Vishnupur, Narsapur, Medak District.

Department of Civil Engineering

CERTIFICATE

This is to certify that the major project work entitled **SOLAR POWERED SMART IRRIGATION SYSTEM** carried out by **ALLE VENNALA (17211A0104)**, **BONGU PRANATHI (17211A0119)**, **CHANDRIKA SRAVYA (17211A0120)**, **EDUPALLI LALITHYA (17211A0138)**, a bonafide students of IV year in partial fulfillment for the award of Bachelor of Technology in Civil Engineering of the JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD, during the year 2021. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The major project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Place: Narsapur

Date:

Dr. G. Sridevi
Project Guide

Dr. G. Sridevi
Professor & HOD

Name and Signature of Examiner with date

TABLE OF CONTENTS

ABSTRACT

In the present scenario, the availability of power and water is insufficient to satisfy the farmer's requirements. Traditionally, implemented techniques of irrigation are proving to be less futile as these are not good at multitasking different concerns, which are a combination of availability of water, sources of energy and timely soil profile analysis. Amidst the merger of automation and the methods of irrigation used earlier, the scope to mitigate issues concerning water and power crisis is huge. To encounter all the above mitigations, we are using the “Internet of Things” and “Solar Energy” to overcome them. In this project, an ESP8266 Wi-Fi Module, Arduino Board, Solar Panel, Soil moisture sensor, Temperature Sensor and Humidity Sensors are used to build a smart irrigation system. This project also includes an SMS alerting system. This Smart Irrigation system paves the way for a better water management level at the field and can also help us save a lot of water.

ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college, ***B. V. Raju Institute of Technology, (Autonomous)*** and Department of ***Civil Engineering*** which gave us the opportunity to have expertise in engineering and profound technical knowledge.

We would like to express our sincere thanks to ***Dr. G. Sridevi***, Project Guide & Head of Civil Engineering Department, for giving us this opportunity to undertake this project.

We wish the gratitude to our principal, ***Dr. K. Lakshmi Prasad,*** for providing us with the environment and means to enrich our skills and motivating us in our Endeavour and helping us realize our full potential.

We also wish to express our indebtedness to our parents as well as our family members whose blessings and support always helped us to face the challenges ahead.

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

(A. VENNALA)
Roll No. 17211A0104

(B. PRANATHI)
Roll No. 17211A0119

(C. SRAVYA)
Roll No. 17211A0120

(E. LALITHYA)
Roll No. 17211A0138

Place: Narsapur
Date:

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

According to Beecham's report entitled “Towards Smart Farming: Agriculture Embracing the IOT Vision” predicts that food production must have to increase by 70 percent in the year 2050 in order to meet our estimated world population of 9.6 billion people. Hence, it is very important to boost up the agricultural productivity to ensure high yield and farm profitability. The major challenge in quality farming is unpredictable weather and environmental conditions such as rainfall, temperature, soil moisture etc. Moreover, humidity is one of the major environmental parameters in farming as it affects the turgor pressure of plants, which is an indicator of the amount of water in plant cells. Due to high rate of transpiration, plants wilt rapidly as too much water is pulled out from plant cells. On the contrary, when amount of moisture in air as well as temperature is high, the rate of transpiration is reduced which in turn restricts evaporative cooling. In order to monitor these environmental conditions and action have been taken accordingly, continuous manual effort was required which is quite impractical and not possible all the times. In this respect, IOT plays a significant role in implementing the concept of smart farming to automate the farming operations. IOT is new computing and communication paradigm in which the objects of everyday life have equipped with sensor, microcontroller and transceiver to sense the surrounding environmental parameters. IOT technology is gaining popularity in agricultural field for its highly scalable, interoperable and pervasive nature. To automate the farming operations, several environmental parameters those have impact on farming, are required to track down at different locations. The important environmental parameters include temperature, moisture and water level. Different types of sensors are deployed over the field to monitor those environmental parameters related to farming and attached with microcontroller. Apart from that these sensed data can be stored in the cloud. Microcontroller attached with wi-fi module sends those sensed parameters to the cloud.

1.2 OBJECTIVES OF THE PROJECT:

- To design and implement a Solar-Powered Smart Irrigation System using IoT Approach.
- To build an energy-efficient and cost-effective system using renewable resources.
- To know soil parameters such as soil moisture content, temperature and humidity values.
- To integrate the system using various sensor parameters such as temperature and humidity sensor, soil moisture sensors and water storage capacity sensor to collect information.
- In this project, the user can water the fields from anywhere using GSM technique.

1.3 LITERATURE REVIEW:

In, M. A. Abdurrahman, et. al. proposed a cost-efficient product for farming where water is scare. The system made up with low-cost sensors and simple circuitry to automatically controls the flow of water. The humidity and temperature level are also sensed and displays in LCD. This system provides water for plants according to the soil moisture level and crop water requirement.

P. A. Bhosale and V. V. Dixit have proposed in an indigenous low cost time depended microcontroller based irrigation scheduler which consists with various sensors for detecting moisture, temperature and wind. This system derives appropriate actuators (relay, solenoid valves, motor) depending on these values. The captured data are conveyed to the user in form of SMS through GSM module and stored into a memory card.

In, F. TongKe proposed smart agriculture based on IoT and cloud computing. Agriculture information cloud is constructed with different resources to achieve dynamic distribution of resource and load balancing. Large amount of data obtained through RFID, wireless communication handled in agriculture information cloud. studied the control network and IoT technology for agricultural production.

CHAPTER 2

BLOCK DIAGRAM

2.1 Block diagram:

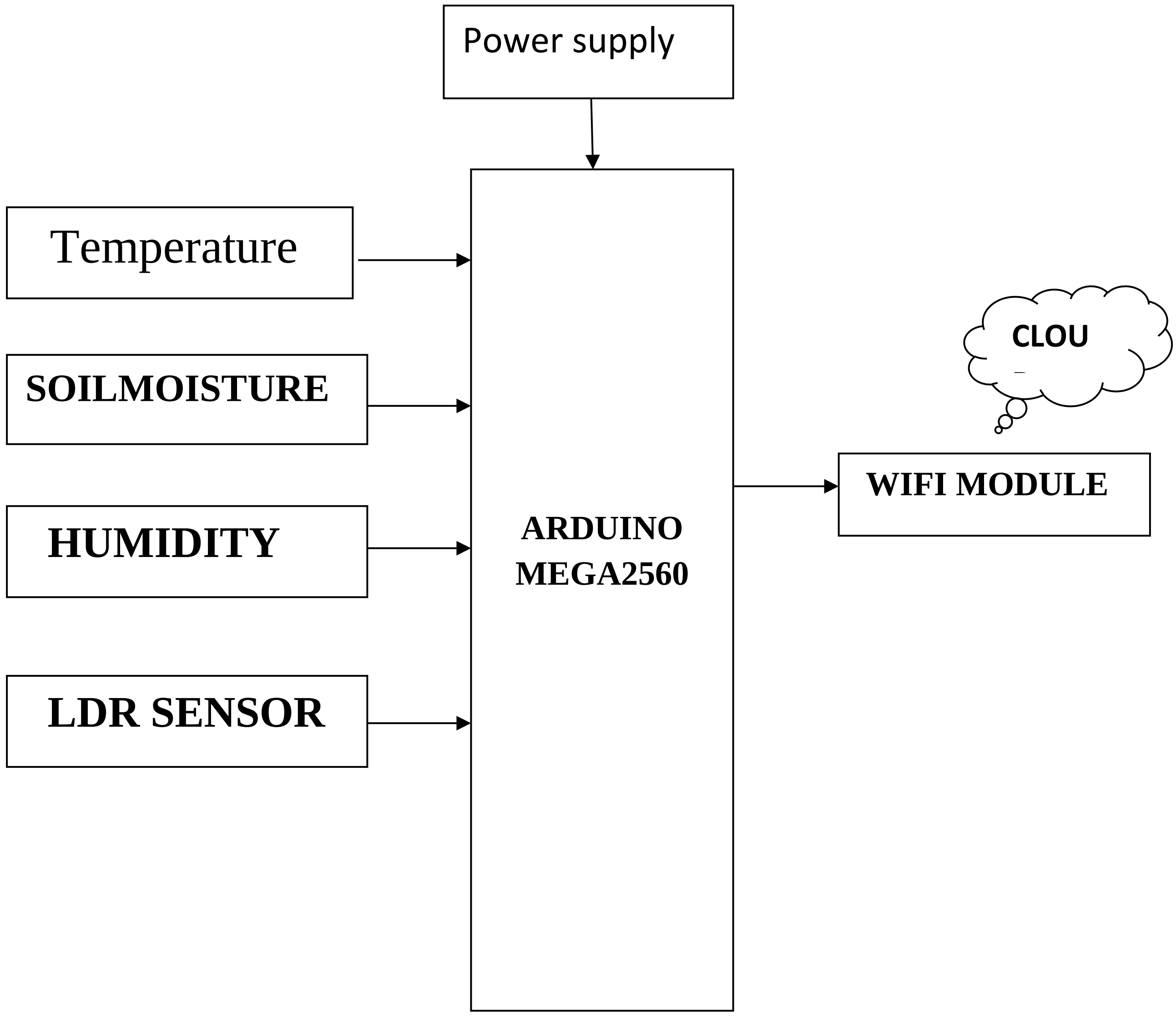


Fig 2.1: Block diagram of the project

2.2 Description of the block diagram:

The ARDUNIO ships as a bare circuit board with standard connections humidity sensor, Temperature, rain sensor, relay, Motor and Lcd, fan with power supply.

2.3 AVR MICROCONTROLLER

2.31 ARDUINO:

Introduction:

Arduino is a computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and Analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

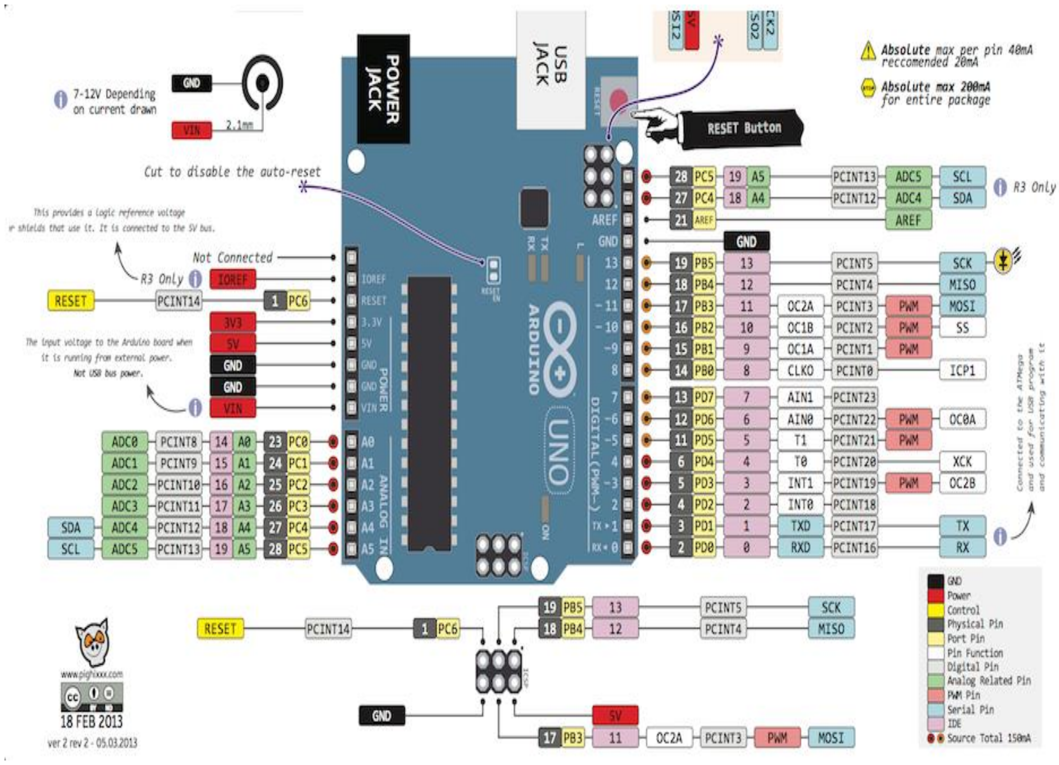


Fig 2.3 Arduino

History:

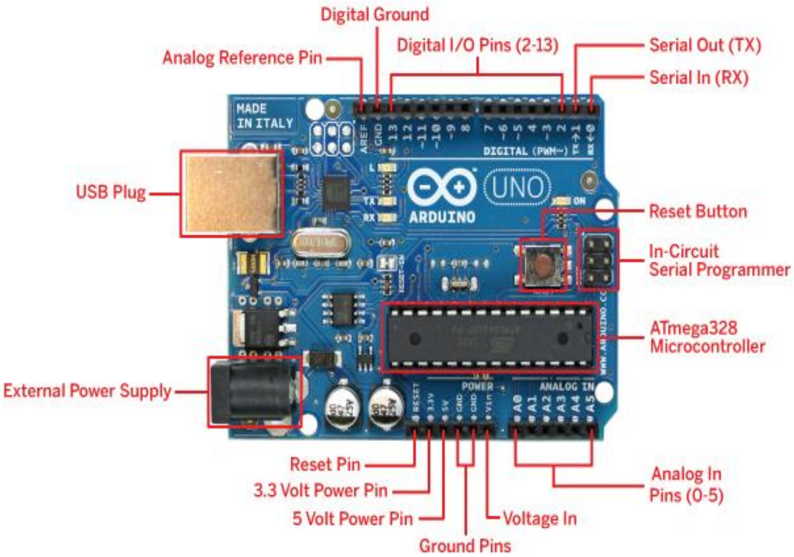
The origin of the Arduino project started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. At that time, the students used a BASIC Stamp microcontroller at a cost of $100, a considerable expense for many students. In 2004, Colombian student Hernando Barragan created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas, who are known for work on the Processing language. The project goal was to create simple, low cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller.

In 2005, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they copied the Wiring source code and renamed it as a separate project, called Arduino.

The initial Arduino core team consisted of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis, but Barragan was not invited to participate. Following the completion of the Wiring platform, lighter and less-expensive versions were distributed in the open-source community.

Adafruit Industries, a New York City supplier of Arduino boards, parts, and assemblies, estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands.

2.31 Hardware:



Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. The source code for the IDE is released under the GNU General Public License, version 2. Nevertheless, an official Bill of Materials of Arduino boards has never been released by Arduino staff.

Although the hardware and software designs are freely available under copyleft licenses,the developers have requested that name Arduino be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in *-duino*. An Arduino board consist of an Atmel 8, 16, 32 bit AVR microcontroller (ATmega8,ATmega168, ATmega328,ATmega1280,ATmega2560), but other makers' microcontrollers have been used since 2015. The boards use single-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple, and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the LilyPad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino UNO is the optibootbootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board orcable, Bluetooth or other methods, when used with traditional microcontroller tools instead of the ArduinoIDE, standard AVR in-system programming (ISP) programming is used.

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current *Uno* provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female

0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The ArduinoNano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

Official boards:

The original Arduino hardware was produced by the Italian company Smart Projects. Some Arduino-branded boards have been designed by the American companies SparkFun Electronics and Adafruit Industries. As of 2016, 17 versions of the Arduino hardware have been commercially produced.

Software development

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension.pde.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Applications:

- Xoscillo, an open-source oscilloscope
- Arduinome, a MIDI controller device that mimics the Monome
- OBDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars
- Ardupilot, drone software and hardware
- Gameduino, an Arduino shield to create retro 2D video games
- Arduino Phone, a do-it-yourself cell phone
- Water quality testing platform
- Automatic titration system based on Arduino and stepper motor
- Low cost data glove for virtual reality applications
- Impedance sensor system to detect bovine milk adulteration
- Homemade CNC using Arduino and DC motors with close loop control by Homofaciens
- DC motor control using Arduino and H-Bridge

Technical specs:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Programming:

The Arduino/Genuino Uno can be programmed with the (Arduino Software (IDE)). Select "Arduino/Genuino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

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

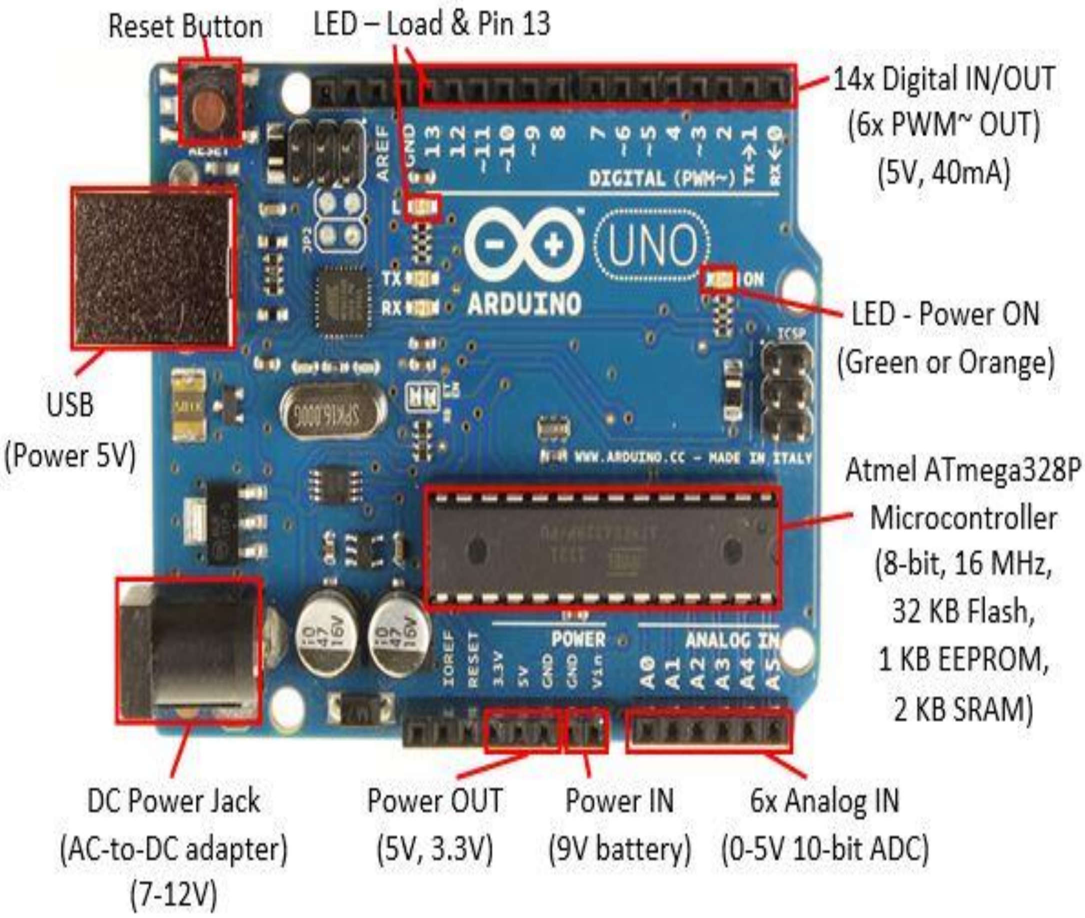
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then reseing the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS

X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.



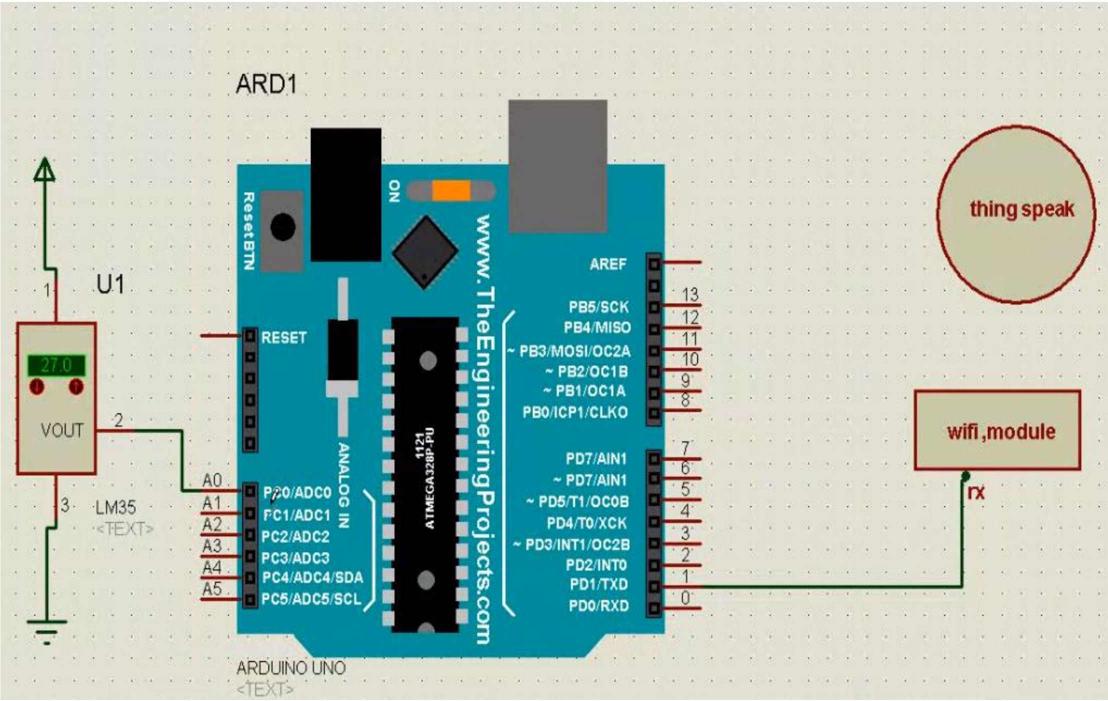
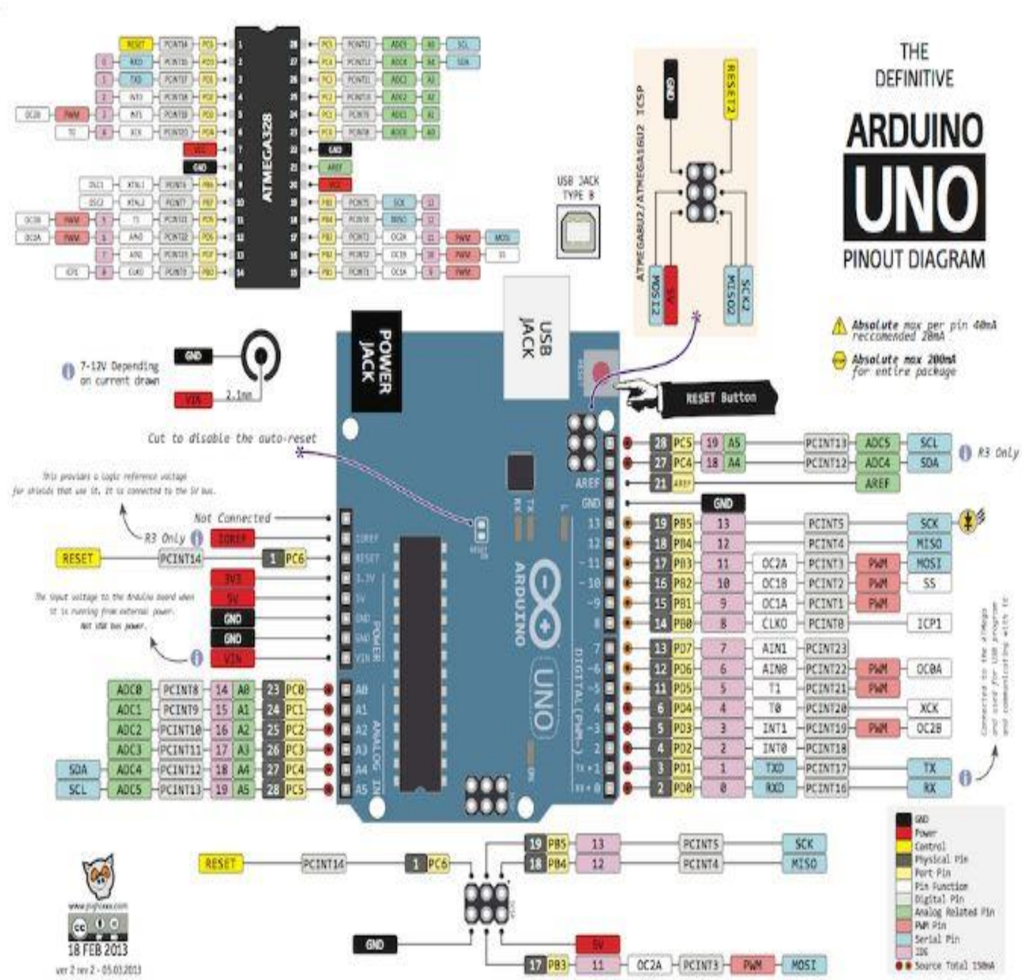
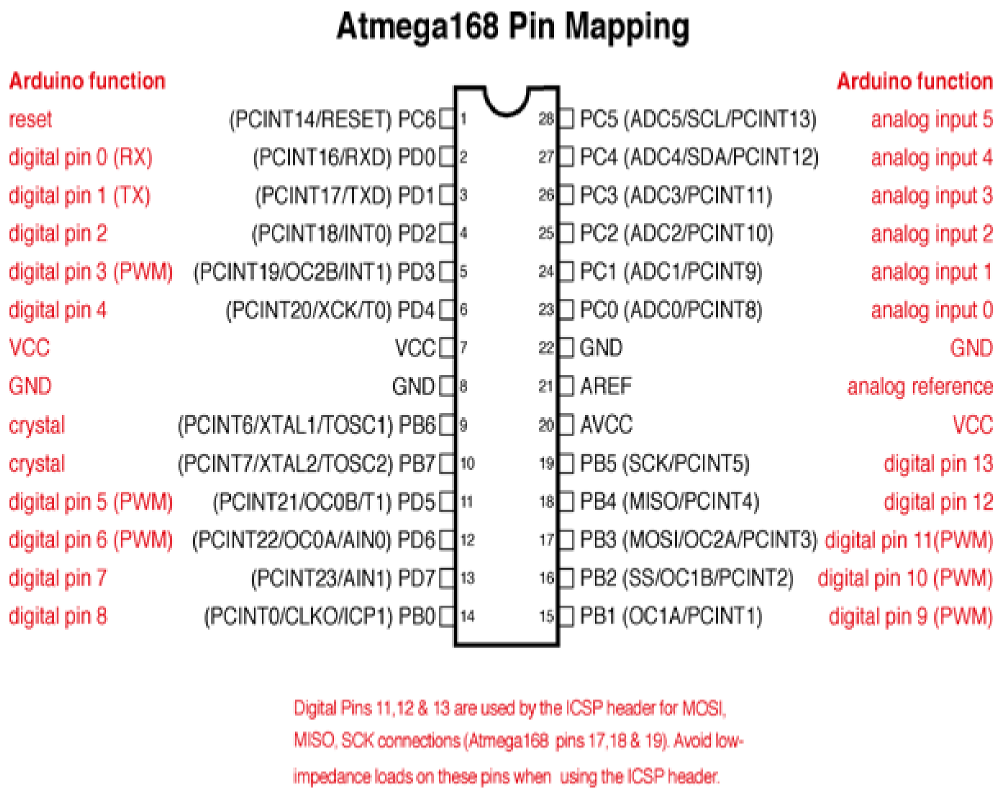


Fig 2.32 Connections





The Arduino/Genuino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Differences with other boards

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Power

The Arduino/Genuino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.040 48576663

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- Vin. The input voltage to the Arduino/genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

Memory:

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical.

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication:

Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

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

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

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

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled"RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

Revisions

Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

CHAPTER 3
FUNCTIONAL MODULES & SOFTWARE MODULES DESCRIPTION

3.1 Temperature Sensor - The LM35

The LM35 is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature (in ^{o}C).

The LM35 - An Integrated Circuit Temperature Sensor

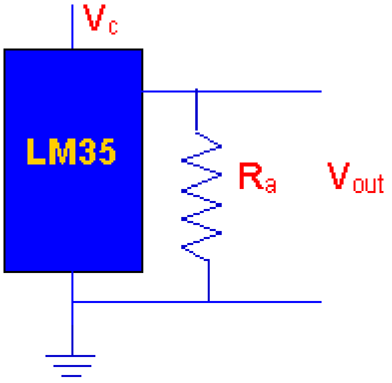
- Use of LM35s To Measure Temperature
  - You can measure temperature more accurately than a using a thermistor.
  - The sensor circuitry is sealed and not subject to oxidation, etc.
  - The LM35 generates a higher output voltage than thermocouples and may not require that the output voltage be amplified.

- LM35 Look Like
  - Here it is.

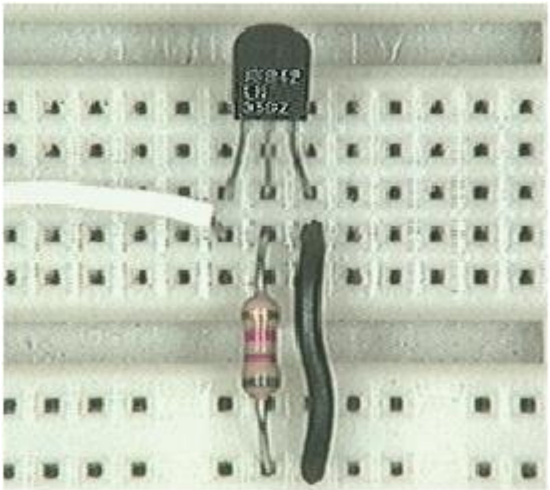
- Working of LM35
  - It has an output voltage that is proportional to the Celsius temperature.

- The scale factor is .01V/°C
- The LM35 does not require any external calibration or trimming and maintains an accuracy of +/-0.4 °C at room temperature and +/- 0.8 °C over a range of 0 °C to +100 °C.
- Another important characteristic of the LM35DZ is that it draws only 60 micro amps from its supply and possesses a low self-heating capability. The sensor self-heating causes less than 0.1 °C temperature rise in still air.

- Use An LM35 (Electrical Connections)
  - Here is a commonly used circuit. For connections refer to the picture above.
  - In this circuit, parameter values commonly used are:
    - V_c = 4 to 30v
    - 5v or 12 v are typical values used.
    - $R_a = V_c / 10^{-6}$
    - Actually, it can range from 80 KW to 600 KW , but most just use 80 KW.



- Here is a photo of the LM 35 wired on a circuit board.
  - The white wire in the photo goes to the power supply.
  - Both the resistor and the black wire go to ground.
  - The output voltage is measured from the middle pin to ground.l



- What Can You Expect When You Use An LM35
  - You will need to use a voltmeter to sense Vout.
  - The output voltage is converted to temperature by a simple conversion factor.
  - The sensor has a sensitivity of 10mV / ^{o}C.
  - Use a conversion factor that is the reciprocal, that is 100 ^{o}C/V.
  - The general equation used to convert output voltage to temperature is:
    - Temperature (^{o}C) = Volt * (100 ^{o}C/V)
    - So if Volt is 1V , then, Temperature = 100 ^{o}C
    - The output voltage varies linearly with temperature.

In order to calculate the Celsius reading from the analog value, we use the following formula to calculate the temperature in Celsius:

$$tempC = \frac{5.0 * val * 100}{1024}$$

Where

Val = is the value send to the computer by the serial port
tempC = is the calculated temperature value (in Celsius)

5 is the reference we are using

1024 is the resolution of the 10bit internal ADC

3.2 Power Supply:

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage. The block diagram of regulated power supply is shown in the figure 3.2

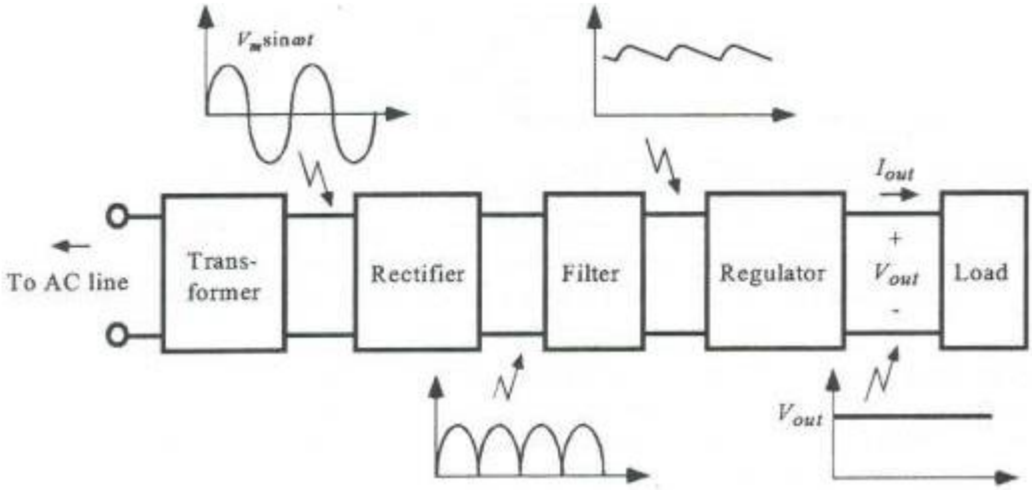


Fig 3.3 components of power supply

Transformer:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

Filter:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore, a regulator is applied at the output stage.

Voltage regulator:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels

3.3 Soil moisture

Soil moisture sensors measure the volumetric water content in soil.[1] Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content.

The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity. Reflected microwave radiation is affected by the soil moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or gardeners.

Soil moisture sensors typically refer to sensors that estimate volumetric water content. Another class of sensors measure another property of moisture in soils called water potential; these sensors are usually referred to as soil water potential sensors and include tensiometers and gypsum blocks.

Technologies commonly used to indirectly measure volumetric water content (soil moisture) include)

- Frequency Domain Reflectometry (FDR): The dielectric constant of a certain volume element around the sensor is obtained by measuring the operating frequency of an oscillating circuit.
- Time Domain Transmission (TDT) and Time Domain Reflectometry (TDR): The dielectric constant of a certain volume element around the sensor is obtained by measuring the speed of propagation along a buried transmission line;[2] (see also: TDR moisture sensor)
- Neutron moisture gauges: The moderator properties of water for neutrons are utilized to estimate soil moisture content between a source and detector probe.
- Soil resistivity: Measuring how strongly the soil resists the flow of electricity between two electrodes can be used to determine the soil moisture content.
- Galvanic cell: The amount of water present can be determined based on the voltage the soil produces because water acts as an electrolyte and produces electricity. The technology behind this concept is the galvanic cell

Accurately measures soil moisture using patented modified TDT technology

• Self-calibrates to all soil types and conditions

• Soil moisture readings are within ±3% of the actual volumetric soil moisture content

• Measurement ranges from 5% moisture to fully saturated soil

• Capable of measuring changes of less than 0.1%

• Measures soil temperature

• Moisture readings are consistent in salty conditions

• Sensor is completely sealed – no electrical contact with soil eliminates any electrostatic degradation or Galvanic corrosion of the sensing element

• Power surge resistant

• Shock resistant

• Not affected by salts or fertilizers

• Provides automatic and user-requested measurements of soil moisture and soil temperature

• Has true two-way communication using a 9-byte packet for commands and replies

• Capable of self-identifying to the two-wire controller and will report pre-configured unique serial numbers

• Has one pre-assigned serial number

• Has standard error collision detections and will resend messages on the two-wire

3.4 Humidity Sensor

Humidity sensor works on the principle of relative humidity and gives the output in the form of voltage. This analog voltage provides the information about the percentage relative humidity present in the environment.

A miniature sensor consisting of a RH sensitive material deposited on a ceramic substrate. The AC resistance (impedance) of the sensor decreases as relative humidity increases.

The relative humidity is defined as:

The analog output of sensor is connected to ADC to get its corresponding digital value. For calibration of digital values, the reference voltage of ADC is set to 1.5 volts. The digital values are received at port of microcontroller. These digital values are used to calculate percentage relative humidity of environment. The calculated data is sent to LCD to display the percentage relative humidity.

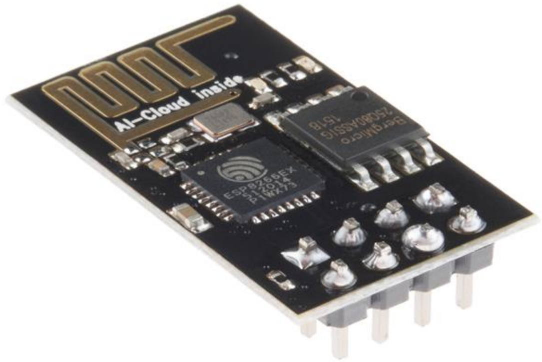
1. High sensitivity and reliability in a small package
2. Fast response time
3. High resistance to chemicals and contaminants
4. Enclosed in a moulded cream colored body
5. 5mm pitch terminations

3.41 Solar panel

- A solar panel, or photo-voltaic (PV) module, is an assembly of photo-voltaic cells mounted in a framework for installation. Solar panels use sunlight as a source of energy and generate direct current electricity. A collection of PV modules is called a PV panel, and a system of panels is an array. Arrays of a photovoltaic system supply solar electricity to electrical equipment.
- Solar panels can be used for many different applications (street lighting, heating systems, machine installations, charging of phones, cameras, signage and many other energy driven devices)
- The pumps used for the transport of the water are equipped with solar cells. The solar energy absorbed by the cells is then converted into electrical energy via a generator which then feeds an electric motor driving the pump.

Fig 3.41 Solar panel

3.42 WiFi Module



Introduction

When power is applied to the module you should see the red power light turn on and the blue serial
indicator light flicker briefly.
If you have a 3.3V FTDI Serial to USB board you can get started without fear of destroying your new
ESP8266 WiFi module. Do note that many FTDI boards have a solder jumper to convert from 3.3V to
5V operation so ensure it is set to enable 3.3V operation.
Here are the connections required to enable communication with the module over serial:

What is this ESP8266

- It's a wireless SoC
- It has GPIO, I2C, ADC, SPI, PWM and some more
- It's running at 80MHz
- 64KBytes of instruction RAM
- 96KBytes of data RAM
- 64KBytes boot ROM
- It has a Winbond W25Q40BVNIG SPI flash
- It's a RISC architecture

- The core is a 106micro Diamond Standard core (LX3) made by Tensilica
- The ESP8266 chip is made by Espressif
- Modules bearing this chip are made by various manufacturers

Description:

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! 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. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the Documents section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution!

Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Please do not power it directly from your 5V dev board.

Note: This new version of the ESP8266 WiFi Module has increased the flash disk size from 512k to 1MB.

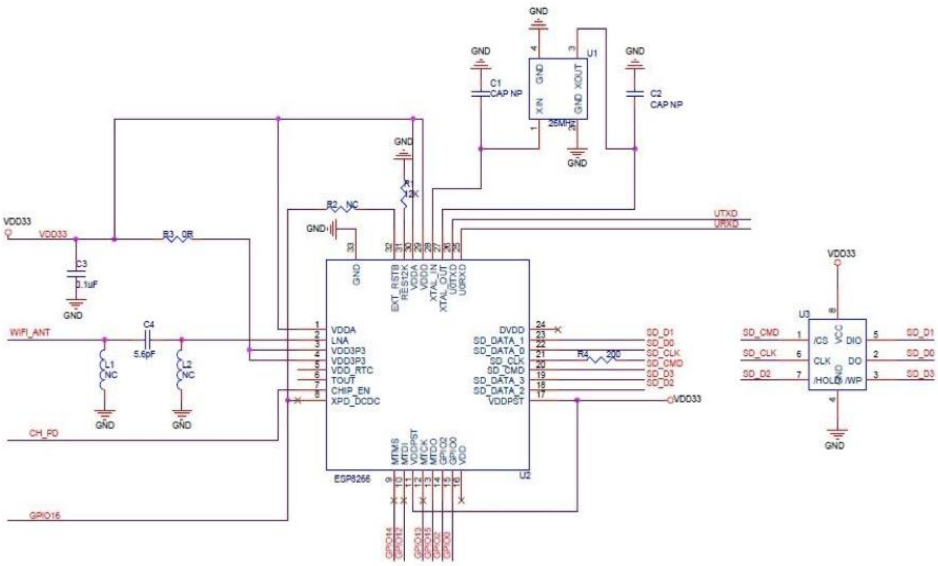
Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)

Technical Overview:

ESP8266 is a complete and self-contained Wi-Fi network solutions that can carry software applications, or through Another application processor uninstall all Wi-Fi networking capabilities. ESP8266 when the device is mounted and as the only application of the application processor, the flash memory can be started directly from an external Move. Built-in cache memory will help improve system performance and reduce memory requirements. Another situation is when wireless Internet access assume the task of Wi-Fi adapter, you can add it to any microcontroller-based design, the connection is simple, just by SPI / SDIO interface or central processor AHB bridge interface. Processing and storage capacity on ESP8266 powerful piece, it can be integrated via GPIO ports sensors and other applications specific equipment to achieve the lowest early in the development and operation of at least occupy system resources. The ESP8266 highly integrated chip, including antenna switch balun, power management converter, so with minimal external circuitry, and includes front-end module, including the entire solution designed to minimize the space occupied by PCB. The system is equipped with ESP8266 manifested leading features are: energy saving

VoIP quickly switch between the sleep / wake patterns, with low-power operation adaptive radio bias, front-end signal processing functions, troubleshooting and radio systems coexist characteristics eliminate cellular / Bluetooth / DDR / LVDS / LCD interference.



Ultra-low power technology:

ESP8266 specifically for mobile devices, wearable electronics and networking applications design and make the machine to achieve the lowest energy consumption, together with several other patented technology. This energy-efficient construction in three modes: active mode, sleep mode and deep sleep mode type. When ESP8266 using high-end power management technology and logic systems to reduce non-essential functions of the power conversion regulate sleep patterns and work modes, in sleep mode, it consumes less than the current 12uA, is connected, it consumes less power to 1.0mW (DTIM = 3) or 0.5mW (DTIM = 10). Sleep mode, only calibrated real-time clock and watchdog in working condition. Real-time clock can be programmed to wake ESP8266 within a specific period of time. Through programming, ESP8266 will automatically wake up when detected certain to happen. ESP8266 automatic wake-up in the shortest time, this feature can be applied to the SOC for mobile devices, so before you turn Wi- Fi SOC are in a low-power standby mode. To meet the power requirements of mobile devices and wearable electronics products, ESP8266 at close range when the PA output power can be reduced through software programming to reduce overall power consumption in order to adapt to different applications.

Maximum integration:

ESP8266 integrates the most critical components on the board, including power management components, TR switch, RF balun, a peak power of + 25dBm of PA, therefore, ESP8266 only guarantee the lowest BOM cost, and easy to be embedded in any system. ESP8266 BOM is the only external resistors, capacitors, and crystal.

ESP8266 application subject:

- Smart Power Plug
- Home Automation
- mesh network
- industrial wireless control
- Baby Monitor
- Network Camera
- sensor networks
- wearable electronics
- wireless location-aware devices
- Security ID tag
- wireless positioning system signals

Specifications:

Power:

The following data are based on a 3.3V power supply, ambient temperature 25C and use the internal regulator measured. [1] All measurements are made in the absence of the SAW filter, the antenna interface is completed. [2] all transmit data based on 90% duty cycle, continuous transmission mode in the measured.

CPU and memory

CPU Interface

The chip embedded in an ultra-low-power 32-bit micro-CPU, with 16 compact mode. Can be connected to the CPU via the following interfaces:

- connecting storage controllers can also be used to access external code memory RAM / ROM interface (iBus)
- Also attached storage controller data RAM interface (dBus)
- Access Register of AHB interface
- JTAG debug interface

Storage Controller:

Storage controller contains ROM and SRAM. CPU can iBus, dBus and AHB interface to access the storage controller. Any one of these interfaces can apply for access to ROM or RAM cells, memory arbiter to determine the running order in the order of arrival.

AHB and AHB module:

AHB module acts as arbiter, through the MAC, and SDIO host CPU control AHB interface. Since sending Address different, AHB data requests may arrive the following two slaves in one: APB module, or flash memory controller (usually in the case of off-line applications) to the received request is a high speed memory controllers often request, APB module receives register access is often Request. APB module acts as a decoder, but only you can access the ESP8266 main module programmable registers. Since the sending address different, APB request may reach the radio receiver, SI / SPI, hosts SDIO, GPIO, UART, real-time clock (RTC), MAC or digital baseband.

Interface:

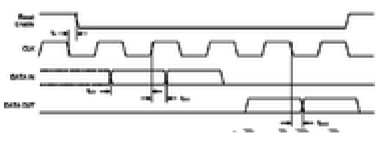
ESP 8266 contains multiple analog and digital interfaces, as follows:

Main SI / SPI control (optional):

Main Serial Interface (SI) can run at two, three, four-wire bus configuration, is used to control the EEPROM or other I2C / SPI devices. Multiple devices share the two-wire I2C bus. Multiple SPI devices to share the clock and data signals, and according to the chip select, each controlled by software alone GPIO pins. SPI can be used to control external devices, such as serial flash, audio CODEC or other slave devices, installation, effectively giving it three different pins, making it the standard master SPI device.

- SPI_EN0
- SPI_EN1
- SPI_EN2

SPI slave is used as the primary interface, giving SPI master and slave SPI support. In the built-in applications, SPI_EN0 is used as an enable signal, the role of external serial flash, download firmware and / or MIB data to baseband. In host-based applications, the firmware and you can choose one MIB data downloaded via the host interface both. This pin is active low when not should be left unconnected. SPI_EN1 often used for user applications, such as controlling the built-in applications or external audio codec sensor ADC. This pin is active low when not should be left unconnected. SPI_EN2 often used to control the EEPROM, storing individual data (individual data), such as MIB information, MAC address, and calibration data, or for general purposes. This pin is active low when not should be left unconnected.



General Purpose IO:

A total of up to 16 GPIO pins. The firmware can assign them different functions. Each GPIO can be configured internal pullup / pulldown resistors available software registers sampled input, triggering edge or level CPU interrupt input, trigger level wake-up interrupt input, open-drain or complementary push-pull output drivers, software register output source or sigma-delta PWM DAC. These pins are multiplexed with other functions, such as the main interface, UART, SI, Bluetooth co-existence and so on.

3.43 Digital IO pins:

Digital IO pad is two-way, three states. It includes a three-state control input and output buffers. In addition, for low-power operation, IO can be set to hold state. For example, when we reduce the chip's power consumption, all the output enable signal can be set to maintain a low-power state. Hold function can be selectively implanted IO in need. When the IO help internal and external circuit driving, hold function can be used to hold last state. Hold function to pin introduce some positive feedback. Therefore, the external drive pin must be stronger than the positive feedback. However, the required driving force size is still small, in the 5uA of.

Variables	Symbol	Min	Max	Units
Input Low Voltage	Vil	-0.3	0.25xV10	V
Input High Voltage	Vih	0.75xV10	3.6	V
Input leakage current	IIL	-	50	nA
Output Low Voltage	VOL	-	0.1xV10	V
Output High Voltage	VOH	0.8xV10	-	V
Input pin capacitance	Cpad	-	2	pF
VDDIO	V10	1.7	3.6	V
Current	Imax	-	12	mA
Temperature	Tamb	-20	100	C

All digital IO pins must add an overvoltage protection circuit (snap back circuit) between the pin and ground. Usually bounce (snap back) voltage is about 6V, while maintaining the voltage is 5.8V. This prevents excessive voltage and generating ESD. Diodes also avoid reverse voltage output devices.

Firmware and software tools development kit:

The firmware running on the ROM and SRAM chip, when the device is awake, firmware via SDIO sector Download the instructions from the host side. Firmware is fully compliant with 802.11 b / g / n / e / i WLAN MAC protocol and Wi-Fi Direct specification only supports basic services unit distributed control function (DCF) under (BSS) operation, but also follow the latest Wi-Fi P2P protocol to support P2P groups operating (P2P group operation). Low level protocol functions automatically run by ESP8266, such as

- RTS / CTS
- Confirm
- fragmentation and reassembly
- polymerization
- frame package (802.11h /RFC 1042)
- automatic beacon monitoring / scanning
- P2P WiFi direct

With P2P discovery procedures, passive or active scanning once in the host command start, it will be done automatically. Perform power management, interaction with the host at least, this way, the task of effectively minimized.

Power Management:

Chip can tune into the following states:

- off (OFF): CHIP_PD pin is in a low power state. RTC failure. All registers are emptied.
- deep sleep (DEEP_SLEEP): RTC open, other parts of the chip are closed. RTC internal recovery memory to save the basic WiFi connection information.
- sleep (SLEEP): Only RTC running. Crystal oscillator stops. Any part of the wake (MAC, host, RTC timer, external interrupt) will make the wake of the chip.
- Wake (WAKEUP): In this state, the system from a sleep state to start (PWR) status. Crystal oscillator and PLL are converted enabled state.

on state (ON): High-speed clock can run, And sent to each clock control register is enabled Modules. Each module, including the CPU, including the implementation of relatively low-level clock gating. When the system works, you can WAITI instructions to turn off the CPU's internal clock.

Clock Management:

3.44 High Frequency Clock:

ESP8266 on high frequency clock is used to drive two Tx and Rx mixer, which is generated by the internal oscillator and an external oscillator. Crystal frequency between 26MHz to 52MHz float. Although the internal crystal oscillator of the calibration range of the crystal so that the clock generator to meet the conditions, but in general, the crystal quality is still obtained a proper phase noise factors to be considered. When the crystal is used, or because of the frequency offset, rather than the best choice for quality, the maximum capacity of the data processing system and will reduce the sensitivity of the wifi. Please refer to the following instructions to measure the frequency offset.

Variables	Symbol	Min	Max	Units
Frequency	Fxo	52	MHz	
Load capacitance	Cl		32	pF
Dynamic capacitance	Cm	2	5	pF
Serial resistance	Rs	0	65	Ω
Frequency tolerance	Fxo	-15	15	ppm
Frequency vs Temperature (-25C ~ 75C)	Fxo,Temp	-15	15	ppm

3.45 External Reference Requirements:

At 26MHz external clock frequency between 52MHz. In order to make a well-functioning radio receiver, the clock will Must have the following characteristics:

Variables	Symbol	Min	Max	Units
Clock amplitude	Vxo	0.2	1	Vpp
External clock accuracy	Fxo,EXT	-15	15	ppm
Phase Noise @ 1kHz offset, 40MHz clock			-120	dBc/Hz
Phase Noise @ 10kHz offset, 40MHz clock			-130	dBc/Hz
Phase Noise @ 100kHz offset, 40MHz clock			-138	dBc/Hz

Radio receivers:

ESP8266 radio receiver mainly includes the following modules:

- 2.4GHz receiver
- 2.4GHz transmitter
- High-speed clock generator and crystal oscillator
- Real-time clock
- bias and regulators
- Power Management

3.46 Channel Frequency:

According IEEE802.11bgn standard, RF transceiver supports the following channels:

Channel	Freq.	Channel	Freq.
1	2412	8	2447
2	2417	9	2452
3	2422	10	2457
4	2427	11	2462
5	2432	12	2467
6	2437	13	2472
7	2442	14	2484

GHz receiver:

2.4GHz RF signal receiver down into quadrature baseband signal, with two high-resolution, high-speed ADC and the latter into a digital signal. In order to accommodate different signal channels, a radio receiver integrated RF filters, automatic gain control (AGC), DC offset compensation circuit and a baseband filter.

GHz transmitter:

2.4 GHz transmitter orthogonal frequency baseband signals up to 2.4GHz, using high-power CMOS power amplifier to drive the antenna. Further use of the digital calibration improves the linearity of the power amplifier to achieve the average power of + 19dBm in 802.11b transmission, the transmission reaches + 16dBm 802.11n average power, features super. To offset defects in the radio receiver is also calibrated by other measures such as:

- carrier leakage
- I / Q phase matching, and
- baseband nonlinear

This will reduce the time and equipment required for testing.

Clock generator:

The clock generator generates the receiver and transmitter 2.4GHz clock signal all of its components are integrated on the chip, Include:

- inductor
- varactor

Closed-loop filter:

Clock generator contains a built-in calibration circuit and self-test circuitry. Clock phase and quadrature phase noise through the optimal calibration algorithm processing patent on the chip, in order to ensure that the receiver and transmitter to achieve the best performance.

3.47 Commands:

- carefully there are must be no any spaces between the " and IP address or port

Commands	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
AT+RST	restart the module	basic	-	-	-	-	
AT+CWMODE	wifi mode	wifi	AT+CWMODE=<mode>	AT+CWMODE?	AT+CWMODE=?	1= Sta, 2= AP, 3=both	
AT+CWJAP	join the AP	wifi	AT+ CWJAP =<ssid>,< pwd >	AT+ CWJAP?	-	ssid = ssid, pwd = wifi password	
AT+CWLAP	list the AP	wifi	AT+CWLAP				
AT+CWQAP	quit the AP	wifi	AT+CWQAP	-	AT+CWQAP=?		
AT+ CWSAP	set the parameters of AP	wifi	AT+ CWSAP= <ssid>,<pwd>,<chl>, <ecn>	AT+ CWSAP ?		ssid, pwd, chl = channel, ecn = encryption	Connect to your router: ：AT+CWJAP="YOURSSID","helloworld"; and check if connected: AT+CWJAP?
AT+ CIPSTATUS	get the connection status	TCP/IP	AT+ CIPSTATUS				

Commands	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
AT+CIPSTART	set up TCP or UDP connection	TCP/IP	1)single connection (+CIPMUX=0) AT+CIPSTART= <type>,<addr>,<port>; 2) multiple connection (+CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	-	AT+CIPSTART=?	id = 0-4, type = TCP/UDP, addr = IP address, port= port	Connect to another TCP server, set multiple connection first: AT+CIPMUX=1; connect: AT+CIPSTART=4,"TCP","X1.X2.X3.X4",9999
AT+CIPSEND	send data	TCP/IP	1)single connection(+CIPMUX=0) AT+CIPSEND=<length>; 2) multiple connection (+CIPMUX=1) AT+CIPSEND= <id>,<length>		AT+CIPSEND=?		send data: AT+CIPSEND=4,15 and then enter the data
AT+CIPCLOSE	close TCP or UDP connection	TCP/IP	AT+CIPCLOSE=<id> or AT+CIPCLOSE		AT+CIPCLOSE=?		
AT+CIFSR	Get IP address	TCP/IP	AT+CIFSR		AT+ CIFSR=?		

Commands	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
AT+ CIPMUX	set mutiple connection	TCP/ IP	AT+ CIPMUX=<mode>	AT+ CIPMUX ?		0 for single connection 1 for mutiple connection	
AT+ CIPSERVER	set as server	TCP/ IP	AT+ CIPSERVER = <mode>[,<port>]			mode 0 to close server mode, mode 1 to open; port = port	turn on as a TCP server: AT+CIPSERVER=1,8888, check the self server IP address: AT+CIFSR=?
+IPD	received data						

3.48 Thing Speak:

According to its developers, "Thing Speak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates"

ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications.

Thing Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Thing Speak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks.

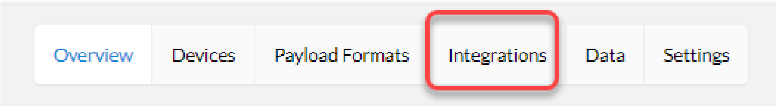
ThingSpeak has a close relationship with Mathworks, Inc. In fact, all of the ThingSpeak documentation is incorporated into the Mathworks' Matlab documentation site and even enabling registered Mathworks user accounts as valid login credentials on the ThingSpeak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and Mathworks, Inc.

ThingSpeak has been the subject of articles in specialized "Maker" websites like Instructables, Codeproject, and Channel 9.

To forward data to ThingSpeak, you must have an application on the Things Network with a registered device and a payload formatter. Create a ThingSpeak integration to forward the data.

1. Log in to The Things Network Console.
2. Select Applications, and then select the application you want to forward data to ThingSpeak from.

3.Click the Integrations tab.

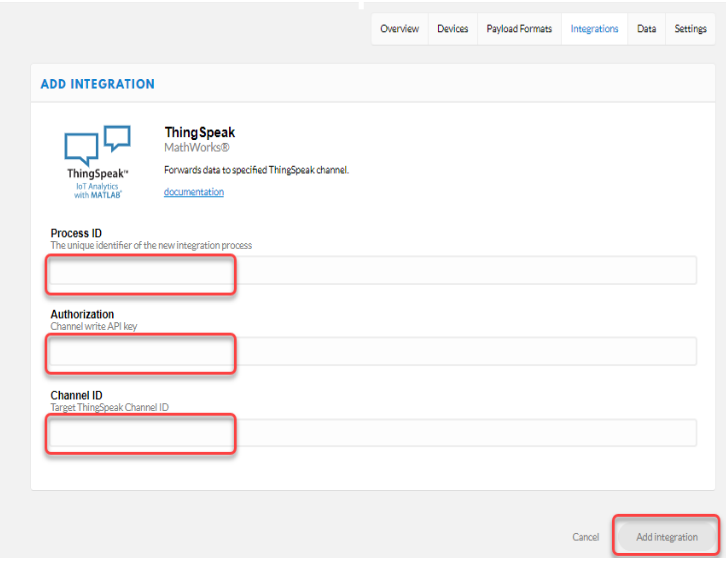


4.Select ThingSpeak.

5.In the Process ID field, name your integration.

6.In the Authorization field, enter the write API key for the channel that you want to store your data in. The API key is available from the 'API keys' tab of your ThingSpeak channel.

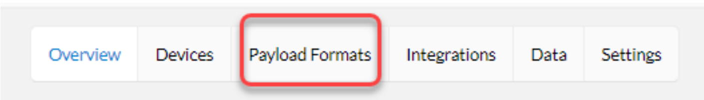
7.In the Channel ID field, enter the channel ID for the ThingSpeak channel you want to forward data into. The channel ID is available on your ThingSpeak channel's page.



Payload Format:

Next, set up the payload formatter. The payload formatter converts bytes sent from your device into a data format that can be stored and visualized on ThingSpeak. The example payload format is for a payload of 20 bytes where the output variables are one, two, three bytes. For an example of how to send a particular payload from a device and format it for the ThingSpeak integration, see Collect Agricultural Data over The Things Network.

1. Click Payload Formats.



3.5 SOFTWARE MODULES DESCRIPTION

3.51 ARDUNIO INSTALLATION:

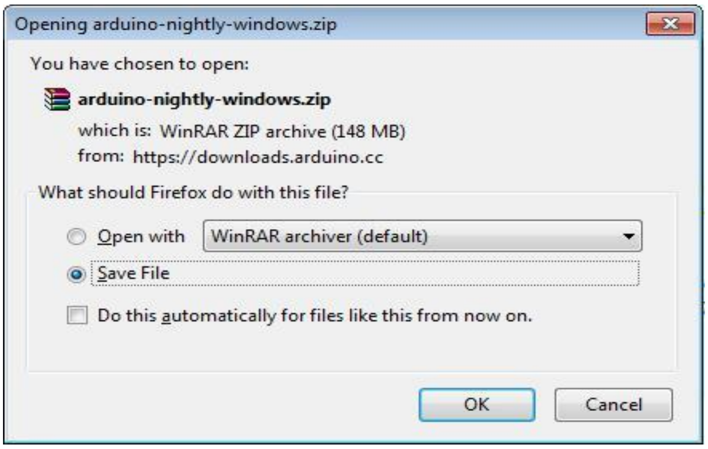
After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1: First you must have your Arduino board (you can choose your favorite board) and aUSB cable. In case you use Arduino UNO, ArduinoDuemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Step 2: Download Arduino IDE Software:

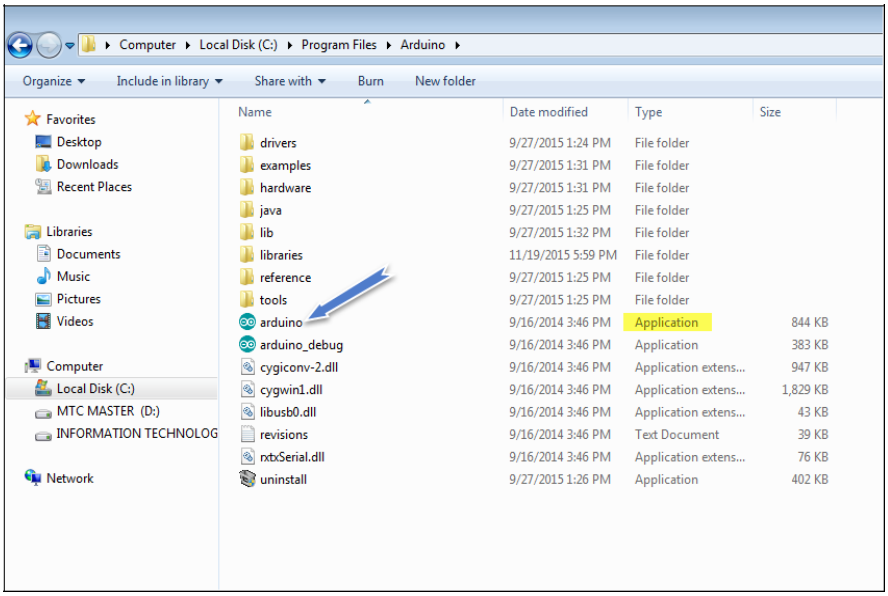
You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with



your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

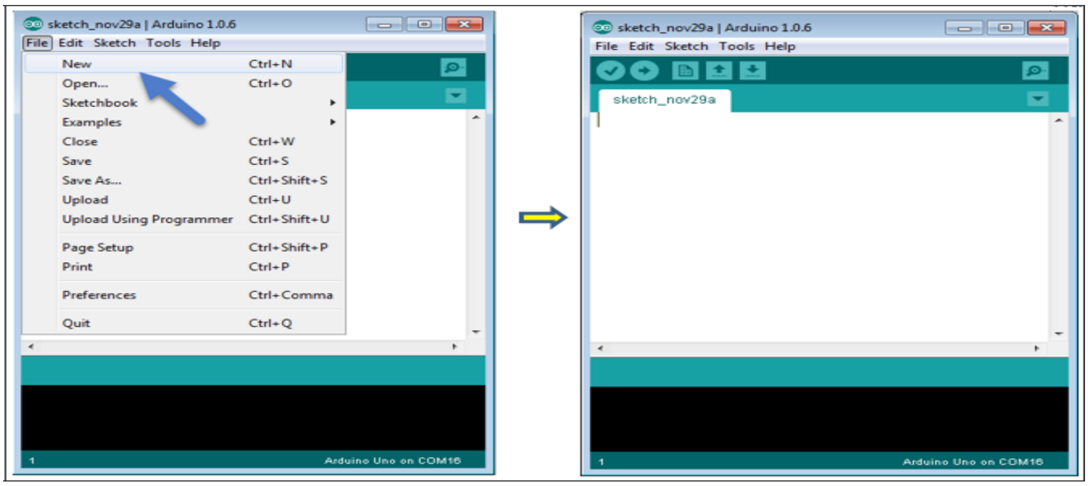
Step 3: Power up your board:

The Arduino Uno, Mega, Duemilanove and ArduinoNano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an ArduinoDiecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should glow.



Step 4: Launch Arduino IDE:

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.



Step 5: Open your first project:

Once the software starts, you have two options:

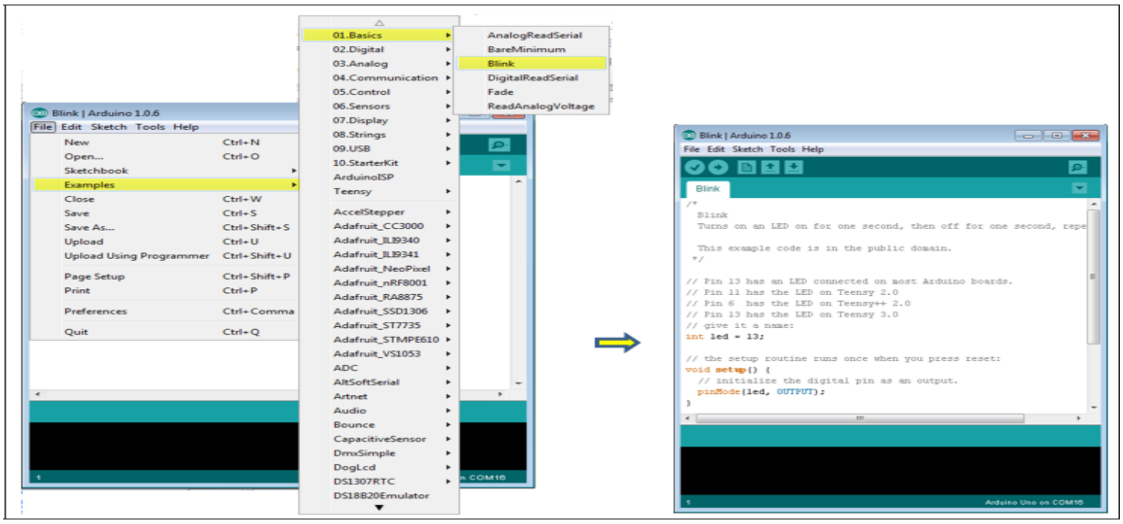
Create a new project.

Open an existing project example.

To create a new project, select File --> New

To open an existing project example, select File -> Example -> Basics -> Blink.

Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.



Step 6: Select your Arduino board:

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools -> Board and select your board

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 Design and implementation:

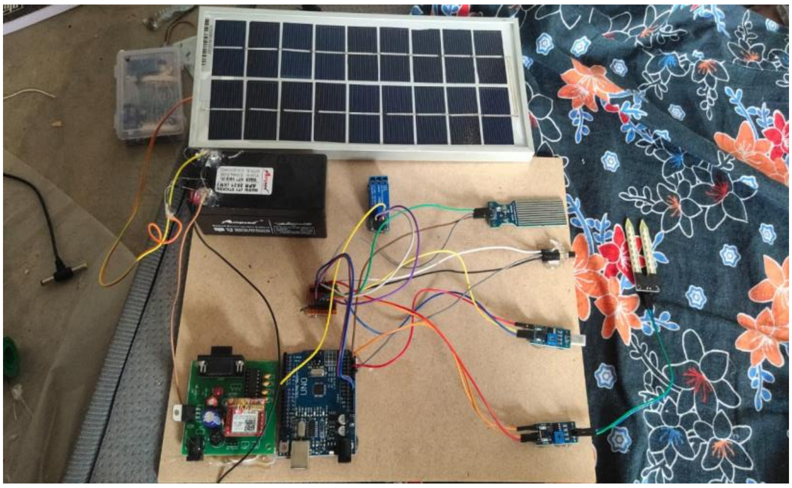


Fig 4.1 Experimental implementation of the project

The Figure 4.1 shows the experimental setup of our smart agriculture monitoring system. Our main objective of this work is to design an IoT based smart farming to control high voltage electrical devices like pump, flap of playhouses etc. without human intervention depending on environmental parameters like soil moisture and temperature. These parameters are stored in cloud for future data analysis. Farming is done within playhouses for better controlled environment. The proposed system is consisting of different layer as represented. It is divided into four modules: Sensor layer, Middleware, Communication Layer and Cloud & Application Layer.

1. Sensor Layer

This is the first layer of our proposed system. It is responsible for capturing and monitoring different environmental parameters. For sensing or collecting the parameters different kinds of sensors are deployed over the agriculture field. For this research work, two types of sensor have used: soil moisture sensor to monitor soil humidity level and temperature sensor to observe temperature level within poly houses. These sensors are attached with Arduino based microcontroller. The microcontroller attached with sensors formed basic IoT objects those are deployed over the agriculture field.

2. Middleware design

This is the second layer of our proposed system. The middleware is needed to automate the farming process and it controls the actuators. It is to be designed for microcontroller. Sensed values are fed into the microcontroller and depending upon the threshold values of different parameters of monitoring field it acts accordingly. This layer carefully monitors temperature and soil moisture level as these two parameters directly affects the crop yield and following decisions are made.

• If soil moisture level is less than the threshold value then microcontroller will turn on the pump machine for watering the field as inadequate moisture content in soil will decrease the crop production. The threshold value of soil moisture content is different for different types of soil. The recommended threshold values of soil moisture content for different types of soil at which irrigation takes place are given values according to. Proposed system considers 15% soil moisture content as a threshold. Once the moisture level reaches the threshold, pump will automatically turn off and thus avoids unnecessary electric power consumption.

• If temperature level is greater than the threshold value then microcontroller will open the flap of the polyhouse. Proposed system considers 40° C temperature as a threshold. Increase in temperature results in reduction in crop duration and affects the equilibrium between crops and pests. It also increases the crop respiration rates and decreases the efficiency of fertilisers. Apart from controlling the actuators, microcontroller sends the sensed data to the ThingSpeak cloud from the field through a gateway.

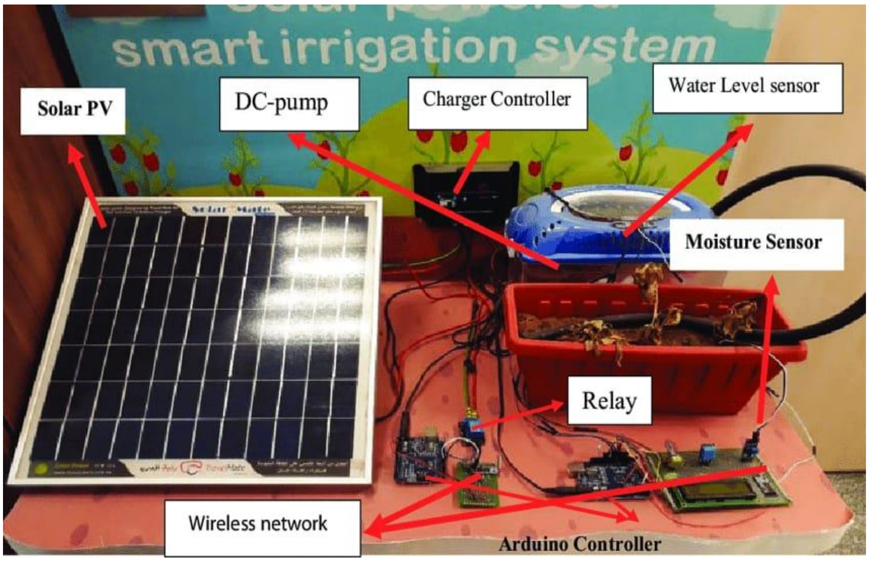
3. Communication Layer

In this layer microcontroller communicates with the gateway wirelessly through Wi-Fi module as it gives advantage over Bluetooth. Bluetooth provides short range communication than Wi-Fi as gateway may be far away from the monitoring field. Ethernet based communication is avoided due to huge cabling. Here, microcontroller is equipped with sensors deployed over monitoring field and sending the sensed soil moisture and temperature value to the cloud through a gateway. IP based protocol is running on the gateway. Microcontroller sends HTTP request to the ThingSpeak cloud for writing sensed value to the corresponding channel.

4. Cloud & Application layer

Cloud computing is an emerging technology and can be used effectively in smart farming. The proposed model uses the cloud computing platform for recording different agricultural field data. In this layer different channels are created, each corresponds to specific parameter field in the ThingSpeak cloud for storing field data (temperature, soil moisture). Microcontroller sends the sensed data to the respective channel periodically through communication protocol. These data (soil moisture value, temperature value) are plotted with respect to time and can be used for future analysis. Agricultural field status (temperature, soil moisture) can be monitored remotely in terms of graph in ThingSpeak web service. Applications can be created related to farming which is deployed in the cloud and can be used by farmers or researchers.

4.2 SNAP SHOT OF THE PROJECT:



4.3 ADVANTAGES AND APPLICATIONS

4.31 Advantages:

1) Low design time.

2) Low production cost.

3) This system is applicable for both the indoor and outdoor environment.

4) Setting the destination is very easy.

5) It is dynamic system.

6) Less space.

7) Low power consumption.

4.32 Applications:

The applications of IOT-based smart agriculture not only target Conventional, large farming operations, but could also be new levers to uplift other growing or common trend in agricultural like organic farming, family farming etc., and enhance highly transparent farming. Some of the applications of smart agriculture system are given below:

i. Precision Farming:

It makes agricultural practice more controlled and accurate when it comes to raising live stock and growing crops. It is one of the most famous applications of IOT in the agriculture sector and numerous organizations are leveraging this technique around the world.

ii. Smart Greenhouses:

Greenhouse farming is a methodology that helps in enhancing the yield of vegetable fruits, crops etc. Greenhouses controls the environmental parameters through manual intervention or a proportional control mechanism.

iii. Illuminum Greenhouses:

It is a drip installation and AgriTech greenhouse organization and uses new modern technologies for providing services.

iv. Agricultural Drones:

Drones are being used in agriculture in order to enhance various agricultural practices like crop health imaging, integrated GIS mapping, ease of use, saves time, and the potential to increase yields.

v. Livestock Monitoring:

Large farm owners can utilize wireless IOT applications to collect data regarding the locations, well-being, and health of their cattle. It also lowers labour costs as ranchers can locate their cattle with the help of IOT based sensors.

CHAPTER 5

PROJECT CODE

5.1 PROJECT CODE

```
#include <SoftwareSerial.h>

SoftwareSerial Serial1(2,3); // RX, TX

int t,h,L,S;

void setup()

{

Serial1.begin(115200);

Serial.begin(9600);

 pinMode(A0,INPUT);

 pinMode(A1,INPUT);

 pinMode(A2,INPUT);

  pinMode(A3,INPUT);

  pinMode(13,OUTPUT);

 }

void loop()

{

t=analogRead(A0);

t=t/2-1;

h=analogRead(A1);

h=h+15;

L=analogRead(A2);

S=analogRead(A3);

Serial.println(L);
```

```
Send_Wifi();
if(S<600)
{
  digitalWrite(13,1);
  init_sms();
}
if(S>600)
{
  digitalWrite(13,0);
  //init_sms();
}
if(L<400)
{
    digitalWrite(13,0);
  init_sms1();
}
}
void Send_Wifi()
{
      Serial1.print("AT\r\n");
      delay(1000);
      Serial1.print("AT+CWMODE=3\r\n");
      delay(3000);
      Serial1.print("AT+CIPMUX=1\r\n");
      delay(3000);
          Serial1.print("AT+CWJAP=\"VITS\",\"12345678\"\r\n"); //ssid and password
```

```
    delay(10000);
    Serial1.print("AT+CIPSTART=4,\"TCP\",\"184.106.153.149\",80\r\n");
    delay(5000);
    Serial1.print("AT+CIPSEND=4,106\r\n");
    delay(1000);
    Serial1.print("GET /update?key=7BPFZ8PK4V782CBF&field1=");
    UARTWriteInt(t,4);
    Serial1.print("&field2=");
    UARTWriteInt(h ,4);
    Serial1.print("&field3=");
    UARTWriteInt(L,4);
     Serial1.print("&field4=");
    UARTWriteInt(S,4);
    delay(300);
    Serial1.print("\r\n");
}
void UARTWriteInt(long val,unsigned int field_length)
{
  char str[10]={0,0,0,0,0,0,0,0,0,0};
  int i=9,j=0;
  while(val)
  {
  str[i]=val%10;
  val=val/10;
  i--;
  }
```

```
  j=10-field_length;
  if(val<0) Serial1.write('   ');
  for(i=j;i<10;i++)
  {
  Serial1.write(48+str[i]);
  }
}
void init_sms()
{
  Serial.println("AT+CMGF=1");
  delay(400);
  Serial.println("AT+CMGS=\"9491038180\"");   // use your 10 digit cell no. here
  delay(400);
  Serial.print("water motor on");
  Serial.write(26);
  delay(2000);
  Serial.println("AT+CMGF=1");
  delay(400);
  Serial.println("AT+CMGS=\"9491038180\"");   // use your 10 digit cell no. here
  delay(400);
  Serial.print("water motor on ");
  Serial.write(26);
  delay(2000);
}
void init_sms1()
{
```

```
  Serial.println("AT+CMGF=1");
  delay(400);
  Serial.println("AT+CMGS=\"9491038180\"");   // use your 10 digit cell no. here
  delay(400);
  Serial.print("water levels low");
  Serial.write(26);
  delay(2000);
  Serial.println("AT+CMGF=1");
  delay(400);
  Serial.println("AT+CMGS=\"9491038180\"");   // use your 10 digit cell no. here
  delay(400);
  Serial.print("water levels low ");
  Serial.write(26);
  delay(2000);
}
```

CHAPTER 6

RESULT

6.1 RESULT:

The solar powered smart irrigation system using IoT demonstrates a collection of data using sensors for productivity and efficiency. This generates a clean energy by utilizing the solar generation technology which improves cost management and waste reduction for overall improved system performance.

After completing the plan and the collection of the components of the smart Irrigation system, it's been met the goal. Also, all of the requirements were implemented in order to finish this smart Irrigation system, so that it becomes full production and finalize. After that, the system became tested, and the end result became as required.

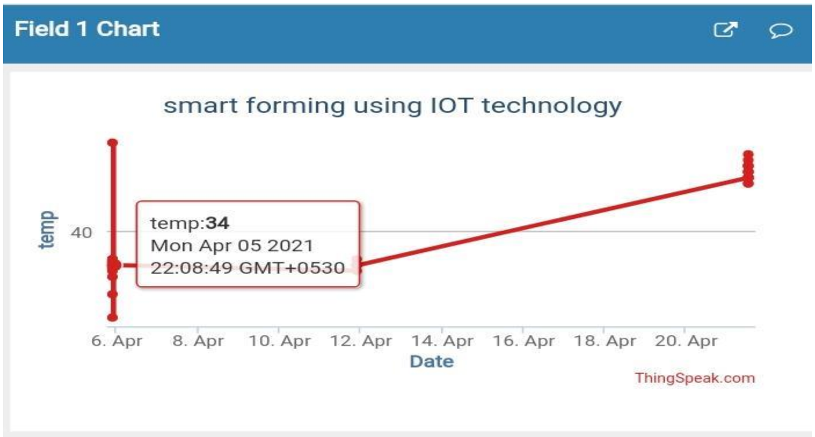
Moisture sensors from any line of the 3 fields send a sign to the Arduino that the soil is dry and crop needs water. After the sign reaches the Arduino, it will send a command to the relay of that specific line field valve to be energized to open the valve and a command to the relay of the pump to exchange it directly to irrigate that field.

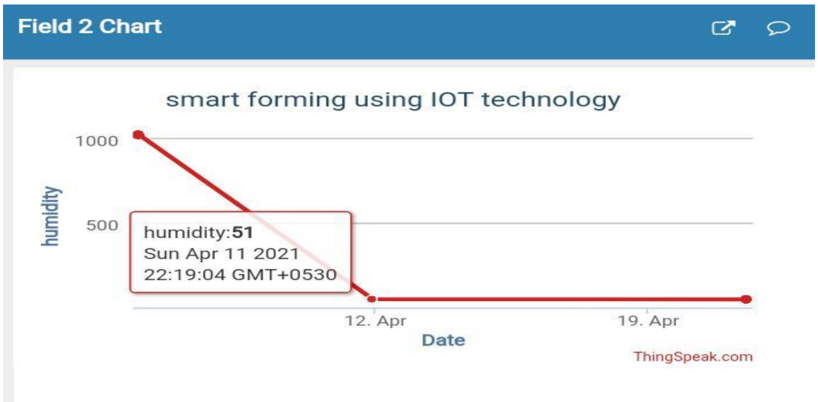
- The smart irrigation system was tested on a garden plant. The plant's water requirement is 15% - 45% a day and temperature requirement of the soil ranges from 50°C- 100°C.
- In the Arduino code, the moisture and temperature range were set respectively. The water level sensor correspondingly sends a sign that checks the amount of water present in order for the operation of supply.

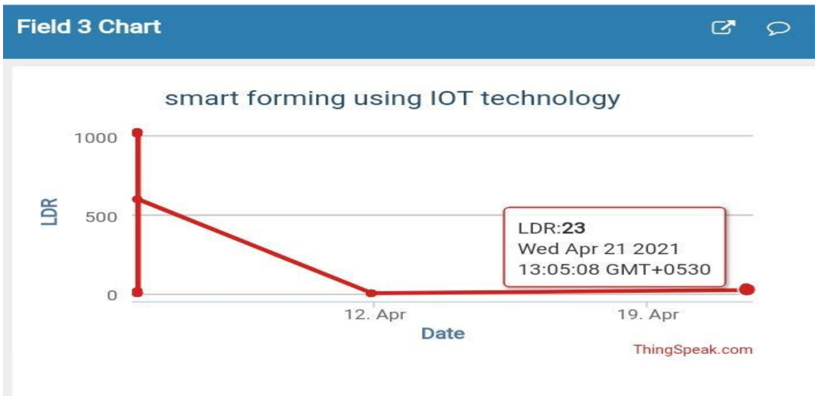
The automatic irrigation control using Arduino uno has been experimentally proven to work satisfactorily and we could successfully set the timer and managed to control the motor over time.

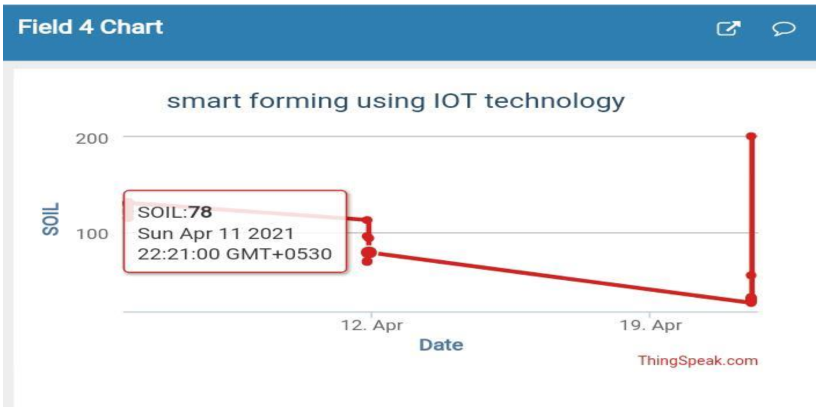
This process not only records values of temperature and humidity it also controls the motor accordingly. Analyzing the weather condition motor will automatically maintain water supply making it possible to maintain greenery without human intervention.

GRAPHICAL ANALYSIS:









CHAPTER 7

CONCLUSION

7.1 CONCLUSION:

The entire system gives the field automation in agriculture, which makes farmer's work easier. It helps in increasing the agricultural production and reduces the time and money of the farmer. Rooftop is useful for smaller farms as it is costly to implement. Graphs are used to analyse present conditions and take necessary actions in future. Android application can be further developed for easier access to all elements in the field and can be used to control the field. Temperature and Humidity values can be used to make statistical analysis regarding the weather conditions in the past and predict the future.

7.2 SCOPE FUTURE:

The future scope of this project could be including variety of soil sensors like pH sensor, Rain sensor and then collecting and storing the data on cloud server. This would make the predicting and analyzing processes more accurate. It also includes making different data mining algorithms suitable for data analysis in agriculture.

These Project can be made more innovative by adding features like checking the faults in the agricultural network and correcting them remotely. Visualization of live working of integrated system in the fields by using a camera. There by enabling the user to take immediate action in case of any problem.

7.3 BIBLIOGRAPHY:

1. K. Lakshmisudha, swathi hegde, neha cole, shruti iyer, " good particularity most stationed cultivation spinning sensors", state-of-the-art weekly going from microcomputer applications (0975- 8887), number 146-no.11, july 2011

2. Nikesh gondchawar, dr. r.complexion.kawitkar, "iot based agriculture", all-embracing almanac consisting of contemporary analysis smart minicomputer additionally conversation planning (ijarcce), vol.5, affair 6, june 2016. Overall Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 5 Issue: 2 177 – 181.

3. M.K.Gayatri, J.Jayasakthi, Dr.G.S.Anandhamala, "Giving Smart Agriculture Solutions to Farmers for Better Yielding Using IoT", IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural

4. Lustiness. r. nandurkar, slant. r. thool, r. tumor. thool, "plan together with situation coming from rigor horticulture technique executing trans - missions sensor network", ieee world consultation toward telemechanics, regulate, intensity also wiring (aces), 2014. Development (TIAR 2015).

5. Paparao Nalajala, D. Hemanth Kumar, P. Ramesh and Bhavana Godavarthi, 2017. Design and Implementation of Modern Automated Real Time Monitoring System for Agriculture using Internet of Things (IoT). Journal of Engineering and Applied Sciences, 12: 9389-9393.

6. Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra Nieto-Garibay, and Miguel Ángel PortaGándara, "Computerized Irrigation System Using a Wireless Sensor Network and GPRS Module", IEEE Transactions on Instrumentation and Measurements, 0018-9456,2013

7. Paparao Nalajala, P Sambasiva Rao, Y Sangeetha, Ootla Balaji, K Navya," Design of a Smart Mobile Case Framework Based on the Internet of Things", Advances in Intelligent Systems and Computing,Volume 815, Pp. 657-666, 2019.