SMART IOT MONITORING SYSTEM FOR AGRICULTURE WITH PREDICTIVE ANALYSIS

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

N.AhamedLimras (RegNo: 37117001)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119

MARCH - 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **N.AhamedLimras (Reg No:** 37117001**)** who carried out the project entitled "**Smart IOT Monitoring System For Agriculture With Predictive Analysis**" under the supervision from **August 2020** to **March 2021**.

Internal Guide

Ms. T. Aandhi, M.E.,

Head of the Department

Dr. S.Vigneshwari, M.E., Ph.D.,

Dr.L.LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I **N.AhamedLimras** hereby declare that the Project Report entitled " **Smart IOTMonitoring System For Agriculture With Predictive Analysis "** is done by me under the guidance of **Ms. T. Anadhi, M.E,** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering**.**

DATE:

N. AhamedLimras

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.T.Aanadhi, M.E.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Smart agriculture is an emerging concept, because IOT sensors are capable of providing information about agriculture fields and then act upon based on the user input. In this Paper, it is proposed to develop a Smart agriculture System that uses advantages of cutting edge technologies such as Arduino, IOT and Wireless Sensor Network. The paper aims at making use of evolving technology i.e. IOT and smart agriculture using automation. Monitoring environmental conditions is the major factor to improve yield of the efficient crops. The feature of this paper includes development of a system which can monitor temperature, humidity, moisture and even the movement of animals which may destroy the crops in agricultural field through sensors using Arduino board and in case of any discrepancy send a SMS notification as well as a notification on the application developed for the same to the farmer's Smartphone using Wi-Fi/3G/4G. The system has a duplex communication link based on a cellular-Internet interface that allows for data inspection and irrigation scheduling to be programmed through an android application. Because of its energy autonomy and low cost, the system has the potential to be useful in water limited geographically isolated area. This system also detects the plant disease. Plant diseases can be detected by image processing technique. Disease detection involves steps like image acquisition, image pre-processing, image segmentation, feature extraction and classification. Our project is used to detect the plant diseases and provide solutions to recover from the disease. It shows the affected part of the leaf in percentage. We planned to design our project with voice navigation system, so a person with lesser expertise in software should also be able to use it easily.

TABLE OF CONTENTS

LIST OF ABBREVATIONS

BCM2837	Broadcom2835
VA	Visual Analytics
IPL	Image Processing Library
USB	Universal Serial Bus
DMA	Direct Memory Access
ARM	Advanced RISC Machine
RISC	Reduced Instruction Set Computer

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Precision agriculture is a new concept in agriculture, it is defined as the farm management system using information technology to identify, analyze and manage the variability of fields to ensure profitability, sustainability, and protection of the environment. In precision agriculture, new information technologies can be used to make better decisions about many aspects of crop production. It is obvious that precision agriculture increases the efficiency that can be realized by understanding and dealing with the natural variability found within a field. Late Blight is a dangerous disease in agriculture field that affects many crops such as potatoes and tomatoes, it is identified by black/brown lesions on leaves and stems that may be small at first but as many lesions accumulate, the entire plant can be destroyed in only a few days after the first lesions are observed. The late blight caused the Irish potato famine of the 1840s which resulted in the death or emigration of over two million people from Ireland and it has been a continuous threat to potato cultivations in Egypt. The potato cultivation in Egypt occupying about 15% of the total vegetable cultivation area.

Machine learning is using precision agriculture benefits to prevent the late blight disease spreading, by using data collected about the climatologically and soil conditions of the plants, the algorithm gives advice to the farmer regarding what should be done to prevent the infection as soon as possible. IoT is also used to collect the sensors data from the field so that the data and the advice of ML algorithm can be available on a GUI platform, which makes it easier to have continuous monitoring of the field. In precision agriculture is used to provide agriculture solution using Artificial Neural Network machine learning algorithm which is used for performing data prediction on data collected by sensors. The use of IoT devices system provides an automated solution for data prediction. The produced result will be helpful for the farmer to take an accurate decision. The system will give all prior knowledge in advance to the farmer for taking proper decision. The proposed system will be used to improve the detection of diseases and predict how the disease will spread in the crop field.In this paper is discus the agriculture sensor values for Predictive Analysis process. Sensor nodes are interface in Robot system. The robot is wireless control robot. It will take sensor reading and send to cloud. This robot is controlled by one wireless gesture system. Using request switches, if that switch is activate sensor data will send to

cloud otherwise data will not update.

CHAPTER 2
LITERATURE SURVEY

1. Project Title: Data Mining for Internet of Things: A Survey

Author Name : Chun-Wei Tsai, Chin-Feng Lai, Ming-Chao Chiang, **Year of publishing:** 2014

Abstract:

It sounds like mission impossible to connect everything on the earth together via internet, but Internet of Things (IoT) will dramatically change our life in the foreseeable future, by making many "impossibles" possible. To many, the massive data generated or captured by IoT are considered having highly useful and valuable information. Data mining will no doubt play a critical role in making this kind of system smart enough to provide more convenient services and environments. This paper begins with a discussion of the IoT. Then, a brief review of the features of "data from IoT" and "data mining for IoT' is given. Finally, changes, potentials, open issues, and future trends of this field are addressed.

2. Project Title: IoT based Smart Agriculture

Author Name : Nikesh Gondchawar, Prof. Dr. R. S. Kawitkar,

Year of publishing: 2016

Abstract:

Agriculture plays vital role in the development of agricultural country. In India about 70% of population depends upon farming and one third of the nation's capital comes from farming. Issues concerning agriculture have been always hindering the development of the country. The only solution to this problem is smart agriculture by modernizing the current traditional methods of agriculture. Hence the project aims at making agriculture smart using automation and IoT technologies. The highlighting features of this project includes smart GPS based remote controlled robot to perform tasks like weeding, spraying, moisture sensing, bird and animal scaring, keeping vigilance, etc. Secondly it includes smart irrigation with smart control and intelligent decision making based on accurate real time field data. Thirdly, smart warehouse management which includes temperature maintenance, humidity maintenance and theft detection in the warehouse. Controlling of all

these operations will be through any remote smart device or computer connected to Internet and the operations will be performed by interfacing sensors, Wi-Fi or ZigBee modules, camera and actuators with micro-controller and raspberry pi.

3. Project Title: IMPLEMENTATION OF SMART FARM MONITORING USING IOT

Author Name : M.Mahendran1, G. Sivakannu2, Sriraman Balaji3

Year of publishing: 2017

Abstract:

An agricultural environment monitoring system provides monitoring services and facility controlling services. This system maintains the crop growth rate in an optimal status. This system also reduces the manpower, time consumption and improves the convenience. The existing monitoring systems are used in an indoor only which isnot used in outdoor environment because

lagging of IT technology. In addition, when users want to check the monitored information in existing monitoring systems, the user must manually check the status through installed sensors or other terminals. In order to solve these issues, the agricultural monitoring system must be designed such a way that can monitor environmental information and soil information closely and reports the status to remote location. The proposed system monitors the environmental status and the status is sent to agricultural monitoring server then the server sends the data to user. The user analyse the data and if the received data is below the specified value then necessary action will be taken. The whole environment is implemented using IoT. The implementation shown here is simulated. Keywords: IOT, Smart Farm.

4. Project Title: Sensor Data Validation

Author Name : 1Dr M Suchithra, 2Asuwini T, 3Charumathi M C, 4Ritu N Lal

Year of publishing: 2018

Abstract:

Agriculture is the primary occupation in our country for ages. In India about 70% of population depends upon farming and one third of the nation's capital comes from farming. But now due to migration of people from rural to urban there is hindrance in agriculture. To overcome this problem we go for smart agriculture techniques using IOT. The Internet of things (IOT) is remodeling the agriculture enabling the farmers with the wide range of techniques such as precision and sustainable agriculture to face challenges in the field. IOT interconnects human to thing, thing to thing and human to human. IOT enables the objects to be sensed and controlled remotely across existing network model. The paper comprises of sensors that sense the field parameters such as temperature,humidity,moisture and fertility in the farm. The sensed values are validated and later sent to the WI-FI module and from WI-FI module the validated data are sent to the farmer's mobile or laptop using cloud. The farmers are also notified by SMS if the field needs a care. An algorithm is developed with threshold values of temperature, humidity, moisture and fertilitythat are programmed into a node MCU to control water quantity. Farmer can automate the motor from anywhere in the world.

CHAPTER 3 SYSTEM ANALYSIS

EXISTING SYSTEM:

In this existing system is discus smart system based on the integration between IoT and machine learning to predict the disease in potatoes and tomatoes. Machine learning algorithm is implemented in this paper. This system is get sensor Field values to apply the machine learning algorithm. It is displaying both data and warning message.

PROPOSED SYSTEM:

In this proposed system is monitoring agriculture land with IoT based predicting analyzing system. Robot system is used to collect data from more sensor nodes and this field data is send to real time cloud and irrigation analyzing process. This system is used to the entire agriculture field. This robot system is reduce overall agriculture field maintain cost and time.

REQUIREMENT SPECIFICATIONS

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

HARDWARE AND SOFTWARE SPECIFICATION

Software Requirement:

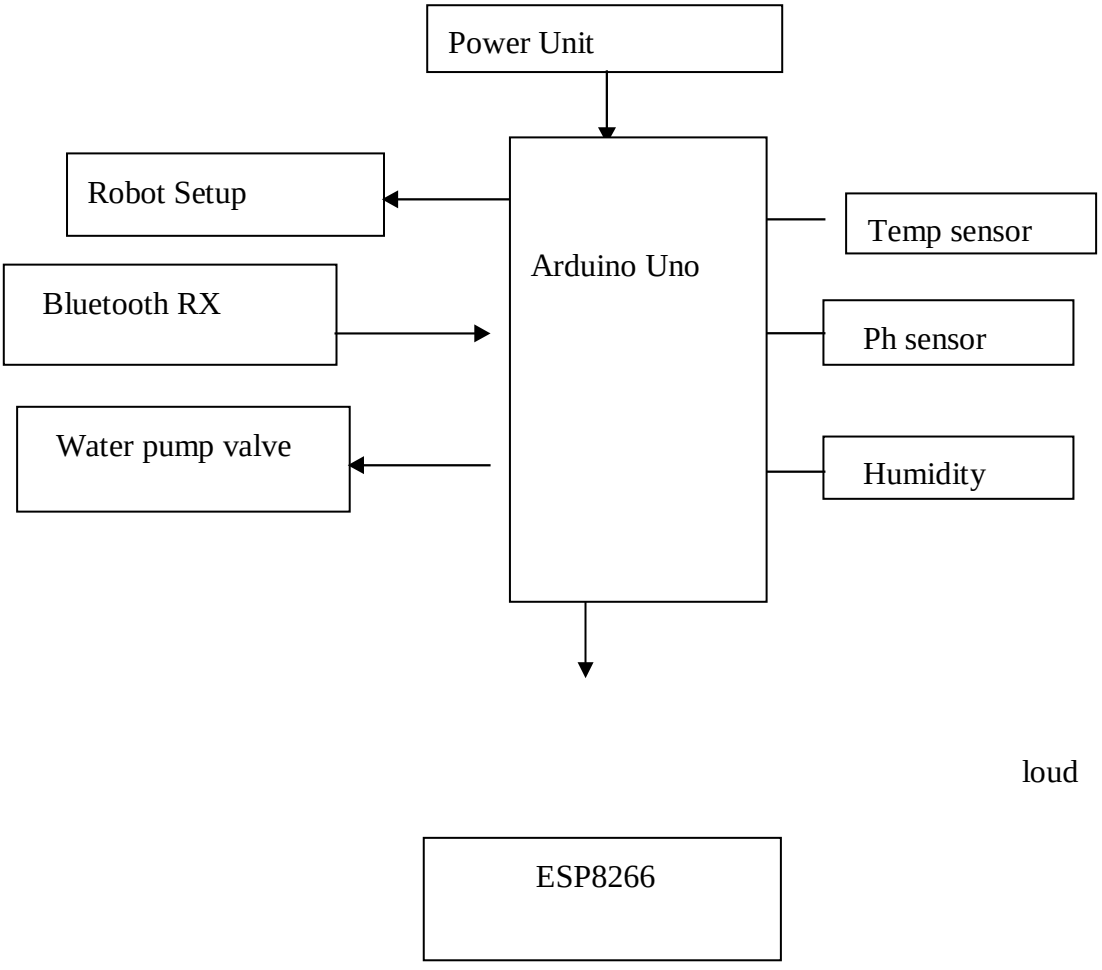
Language	:	Python, c++
Compiler	:	Arduino IDE

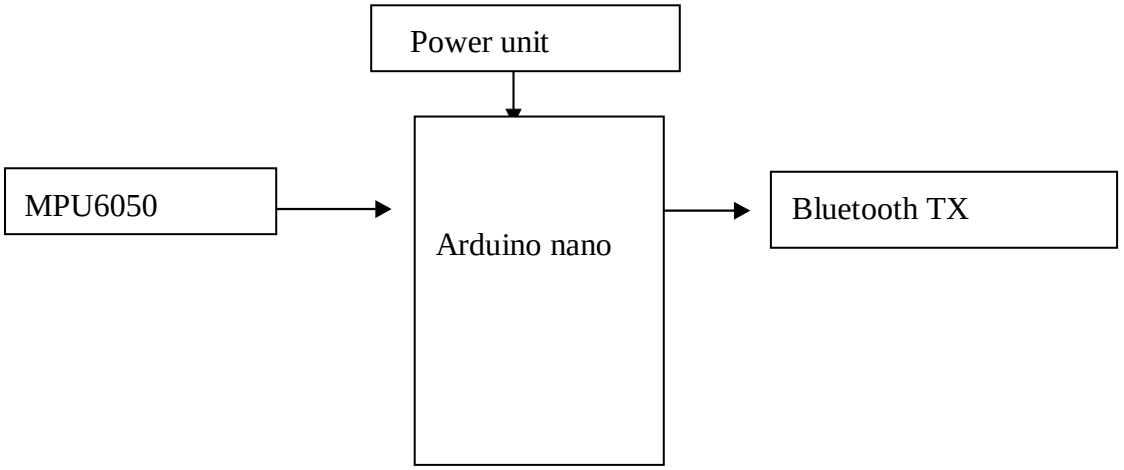
OS : LINUX

Hardware Requirement:

- Arduino uno
- Arduino Nano
- Temperature sensor
- Humidity sensor
- ph sensor
- ESP8266-1
- Power supply
- bluetooth-2
- water pump valve

BLOCK DIAGRAM





BLOCK DIAGRAM DESCRIPTION:

Above the block diagram is contain raspberry pi 3, water level sensor, PH sensor, DHT11, Relay, Pump motor and MCP3008 power supply unit. MCP3008 is nothing but analog to digital convertor. Raspberry pi has only digital input and output pins. Water level sensor and PH sensor is connected with MCP3008. MCP3008 is connected to raspberry pi by using SPI protocol. DHT11 is directly connected to GPIO pin of raspberry pi. Relay switch is nothing but electrical switch. If Sensor value will reached abnormal level pump motor automatically switch on by using relay.

CHAPTER 4

SYSTEM DESIGN

HARDWARE MODULES DETAILS:

- Arduino Nano
- Temperature sensor
- Humidity sensor
- ph sensoArduino uno
- ESP8266-1
- Power supply
- bluetooth-2
- water pump valve

ARDUINO UNO:

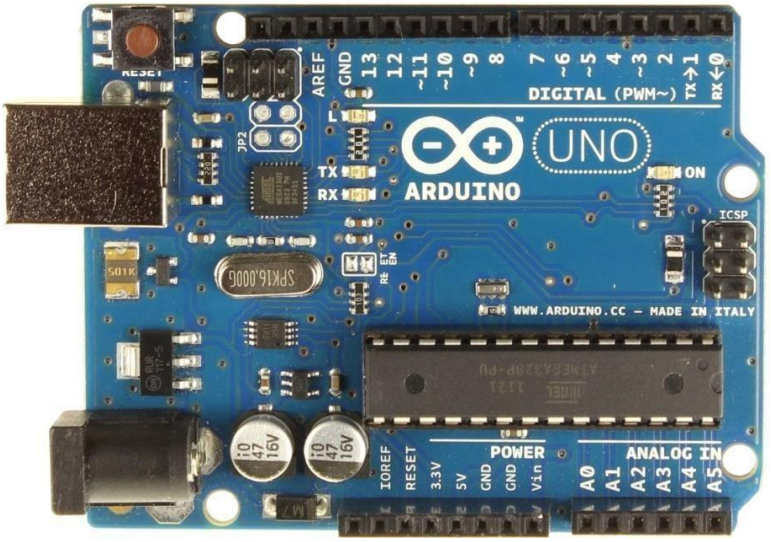


FIG 4.1

Specifications:

- The ATmega2560 is a Microcontroller
- The operating voltage of this microcontroller is 5volts
- The recommended Input Voltage will range from 7volts to 12volts
- The input voltage will range from 6volts to 20volts
- The digital input/output pins are 54 where 15 of these pins will supply PWM o/p.

- Analog Input Pins are 16
- DC Current for each input/output pin is 40 mA
- DC Current used for 3.3V Pin is 50 mA
- Flash Memory like 256 KB where 8 KB of flash memory is used with the help of boot loader
- The static random access memory (SRAM) is 8 KB
- The electrically erasable programmable read-only memory (EEPROM) is 4 KB
- The clock (CLK) speed is 16 MHz
- The USB host chip used in this is MAX3421E
- The length of this board is 101.52 mm
- The width of this board is 53.3 mm
- The weight of this board is 36 g

Key Benefits:

- Low cost
- Consistent board format
- 10x faster processing
- Added connectivity

Key Applications:

- Low cost PC/tablet/laptop
- IoT applications
- Media centre
- Robotics
- Industrial/Home automation

- Server/cloud server

- Print server
- Security monitoring
- Web camera
- Gaming
- Wireless access point
- Environmental sensing/monitoring (e.g. weather station)

PIN DIAGRAM:

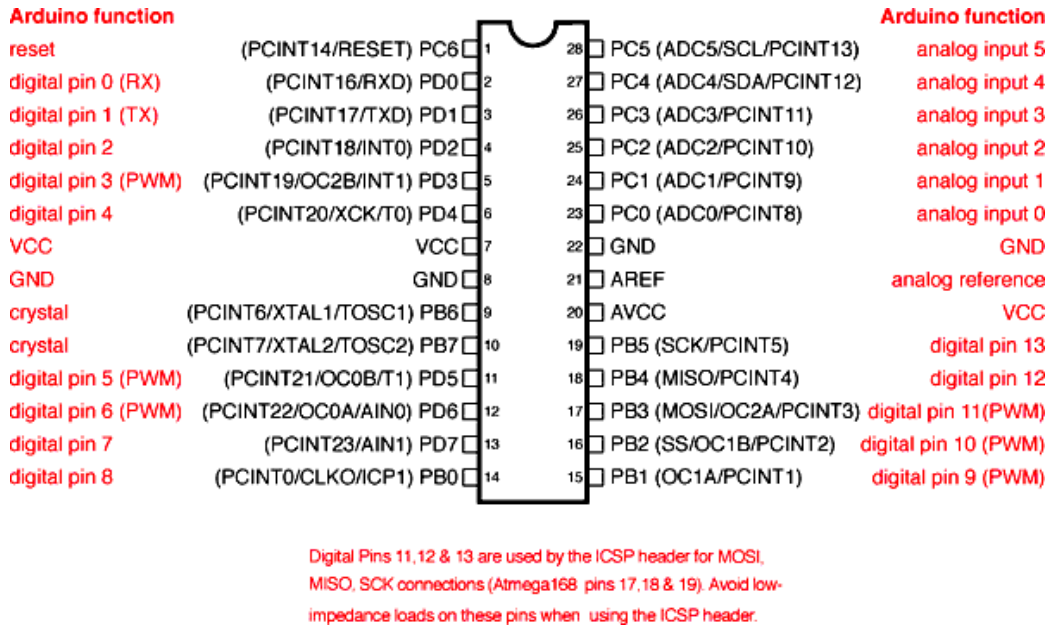


FIG 4.2

ARDUINO UNO PERIPHERALS

1. Introduction

Overview

The Arduino uno is a microcontroller board based on the ATmega1280 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog

inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support

the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The uno is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: arduino-mega-reference-design.zip

Schematic: arduino-mega-schematic.pdf

Summary

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino uno 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 center-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.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be 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 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.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega1280 has 128 KB of flash memory for storing code (of which 4 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 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 a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** 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: 2 to 13 and 44 to 46.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove or Diecimila.

The uno has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and 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

The Arduino uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega1280 provides four hardware UARTs for TTL (5V) serial communication. An FTDI FT232RL on the board channels one of these over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

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

The ATmega1280 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega1280 datasheet.

Programming

The Arduino uno can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The ATmega1280 on the Arduino uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

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

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the

reset line of the ATmega1280 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 uses

this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader 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 boot loader 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 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.

USB Overcurrent Protection

The Arduino 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.

Physical Characteristics and Shield Compatibility

The maximum length and width of the uno PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The uno is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0

to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1

(pins 2 and 3 respectively). SPI is available through the ICSP header on both the uno and Duemilanove

4.1.2 ESP8266:

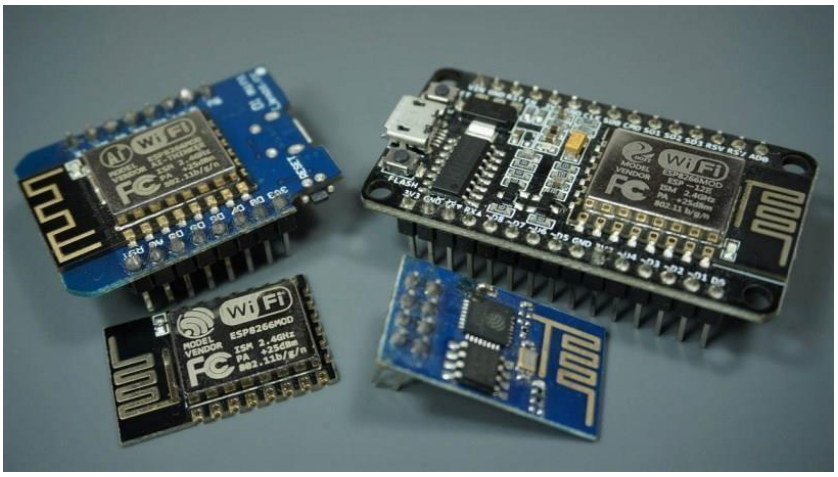


Fig 4.3

The ESP8266 12-E chip comes with 17 GPIO pins. Not all GPIOs are exposed in all ESP8266 development boards, some GPIOs are not recommended to use, and others have very specific functions. With this guide, you'll learn how to properly use the ESP8266 GPIOs and avoid hours of frustration by using the most suitable pins for your projects.

ESP8266 Peripherals

The ESP8266 peripherals include:

- 17 GPIOs
- SPI
- I2C (implemented on software)
- I2S interfaces with DMA
- UART
- 10-bit ADC

Best Pins to Use – ESP8266

One important thing to notice about ESP8266 is that the GPIO number doesn't match the label on the board silkscreen. For example, D0 corresponds to GPIO16 and D1 corresponds to GPIO5.

The following table shows the correspondence between the labels on the silkscreen and the GPIO number as well as what pins are the best to use in your projects, and which ones you need to be cautious.

The pins highlighted in green are OK to use. The ones highlighted in yellow are OK to use, but you need to pay attention because they may have unexpected behavior mainly at boot. The pins highlighted in red are not recommended to use as inputs or outputs.

GPIOs connected to the Flash Chip

GPIO6 to GPIO11 are usually connected to the flash chip in ESP8266 boards. So, these pins are not recommended to use.

Pins used during Boot

The ESP8266 can be prevented from booting if some pins are pulled LOW or HIGH. The following list shows the state of the following pins on BOOT:

- **GPIO16:** pin is high at BOOT
- **GPIO0:** boot failure if pulled LOW
- **GPIO2**: pin is high on BOOT, boot failure if pulled LOW
- **GPIO15**: boot failure if pulled HIGH
- **GPIO3**: pin is high at BOOT
- **GPIO1**: pin is high at BOOT, boot failure if pulled LOW
- **GPIO10**: pin is high at BOOT
- **GPIO9**: pin is high at BOOT

Pins HIGH at Boot

There are certain pins that output a 3.3V signal when the ESP8266 boots. This may be

problematic if you have relays or other peripherals connected to those GPIOs. The following GPIOs output a HIGH signal on boot:

- GPIO16
- GPIO3
- GPIO1
- GPIO10
- GPIO9

Additionally, the other GPIOs, except GPIO5 and GPIO4, can output a low-voltage signal at boot, which can be problematic if these are connected to transistors or relays. You can read this article that investigates the state and behavior of each GPIO on boot.

Analog Input

The ESP8266 only supports analog reading in one GPIO. That GPIO is called **ADC0** and it is usually marked on the silkscreen as **A0**.

The maximum input voltage of the ADC0 pin is 0 to 1V if you're using the ESP8266 bare chip. If you're using a development board like the ESP8266 12-E NodeMCU kit, the voltage input range is 0 to 3.3V because these boards contain an internal voltage divider.

You can learn how to use analog reading with the ESP8266 with the following guide:

On-board LED

Most of the ESP8266 development boards have a built-in LED. This LED is usually connected to GPIO2.

WATER LEVEL SENSOR:

The sensor has a series of ten exposed copper traces, five of which are power traces and five are sense traces. These traces are interlaced so that there is one sense trace between every two power traces. Usually these traces are not connected but are bridged by water when submerged.

How Water Level Sensor Works:

The working of the water level sensor is pretty straightforward. The series of exposed parallel conductors, together acts as a variable resistor (just like a potentiometer) whose

resistance varies according to the water level. The change in resistance corresponds to the distance from the top of the sensor to the surface of the water. The resistance is inversely proportional to the height of the water: The more water the sensor is immersed in, results in better conductivity and will result in a lower resistance. The less water the sensor is immersed in results in poor conductivity and will result in a higher resistance. The sensor produces an output voltage according to the resistance, which by measuring we can determine the water level.

Water Level Sensor Pin out:

The water level sensor is super easy to use and only has 3 pins to connect.

OUT pin is an analog output that will be connected to one of the analog inputs on your Arduino.

VCC pin supplies power for the sensor. It is recommended to power the sensor with between 3.3V – 5V. Please note that the analog output will vary depending on what voltage is provided for the sensor.

GND pin is a ground connection.

PH SENSOR:

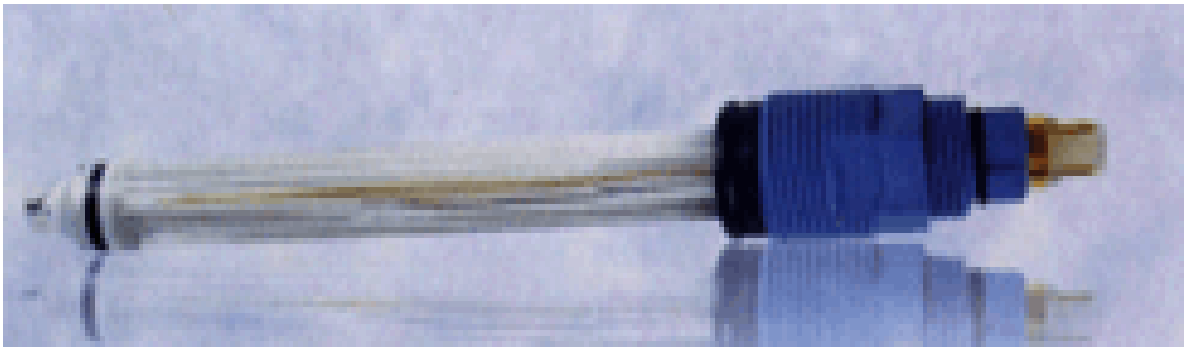


FIG 4.4

1. Preface

In the process world, pH is an important parameter to be measured and controlled. The pH

of a solution indicates how acidic or basic (alkaline) it is. The pH term translates the values of the hydrogen ion concentration which ordinarily ranges between about 1 and 10 x -14 gram-equivalents per litre - into numbers between 0 and 14. On the pH scale a

very acidic solution has a low pH value such as 0, 1, or 2 (which corresponds to a large concentration of hydrogen ions; 10 x 0, 10 x -1, or 10 x -2 gram-equivalents per litre) while a very basic solution has a high pH value, such as 12, 13, or 14 which corresponds to a small number of hydrogen ions (10 x -12, 10 x -13, or 10 x -14 gram-equivalents per litre). A neutral solution such as water has a pH of approximately 7.

A pH measurement loop is made up of three components, the pH sensor, which includes a measuring electrode, a reference electrode, and a temperature sensor; a preamplifier; and an analyzer or transmitter. A pH measurement loop is essentially a battery where the positive terminal is the measuring electrode and the negative terminal is the reference electrode. The measuring electrode, which is sensitive to the hydrogen ion, develops a potential (voltage) directly related to the hydrogen ion concentration of the solution.

Typical pH sensor:

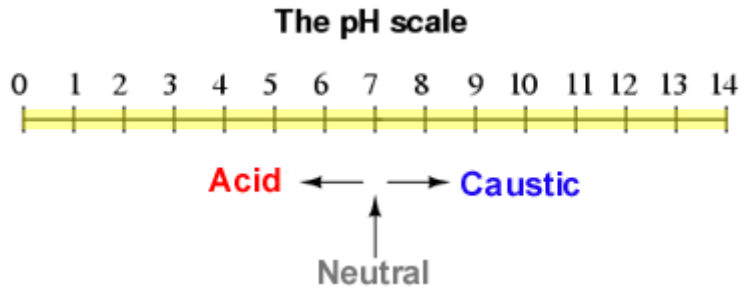
When immersed in the solution, the reference electrode potential does not change with the changing hydrogen ion concentration. A solution in the reference electrode also makes contact with the sample solution and the measuring electrode through a junction, completing the circuit. Output of the measuring electrode changes with temperature (even though the process remains at a constant pH), so a temperature sensor is necessary to correct for this change in output.

This is done in the analyser or transmitter software.
The pH sensor components are usually combined into one device called a combination pH electrode. The measuring electrode is usually glass and quite fragile. Recent developments have replaced the glass with more durable solid-state sensors. The preamplifier is a signal-conditioning device. It takes the high-impedance pH electrode signal and changes it into low impedance signal which the analyser or transmitter can accept. Additionally, the analyzer or transmitter has a man machine interface for calibrating the sensor and configuring outputs and alarms, if pH control is being done.

Keep in mind, application requirements should be carefully considered when choosing a pH electrode. Accurate pH measurement and the resulting precise control that it can allow,

can go a long way toward process optimization and result in increased product quality and consistency. Accurate, stable pH measurement also controls and often lowers chemical usage, minimizing system maintenance and expense. Keeping the system up and running. A system's pH electrodes require periodic

maintenance to clean and calibrate them. The length of time between cleaning and calibration depends on process conditions and the user's accuracy and stability expectations. Cleaning of the measuring sensor and reference junction will also help. However, just as batteries have a limited life, a pH electrode's lifetime is also finite. The measurement of hydrogen ion concentration in a liquid solution. A solution with a low pH value is called an "acid," while one with a high pH is called a "caustic." The common pH scale extends from 0 (strong acid) to14 (strong caustic), with7inthe middle representing pure water (neutral):



pH is defined as follows: the lower-case letter "p" in pH stands for the negative common (base ten) logarithm, while the upper-case letter "H" stands for the element hydrogen. Thus, pH is a logarithmic measurement of the number of moles of hydrogen ions (H+) per litre of solution. Incidentally, the "p" prefix is also used with other types of chemical measurements where a logarithmic scale is desired, pCO2 (Carbon Dioxide) and pO2 (Oxygen) being two such examples.

The logarithmic pH scale works like this: a solution with 10-12 moles of H+ ions per liter has a pH of 12; a solution with 10-3 moles of H+ ions per liter has a pH of 3. While very uncommon, there is such a thing as an acid with a pH measurement below 0 and a caustic with a pH above 14. Such solutions, understandably, are quite concentrated and extremely reactive.

While pH can be measured by color changes in certain chemical powders (the "litmus strip" being a familiar example from high school chemistry classes), continuous process monitoring and control of pH requires a more sophisticated approach. The most common approach is the use of a specially-prepared electrode designed to allow hydrogen ions in the solution to migrate through a selective barrier, producing a measurable potential

(voltage) difference proportional to the solutions. The design and operational theory of pH electrodes is a very complex subject, explored only briefly here. What is important to understand is that these two electrodes generate a voltage directly proportional to the pH

of the solution. At a pH of 7 (neutral), the electrodes will produce0 volts between them. At a low pH (acid) a voltage will be developed of one polarity, and at a high pH (caustic) a voltage will be developed of the opposite polarity.

An unfortunate design constraint of pH electrodes is that one of them(called the measurement electrode) must be constructed of special glass to create the ion-selective barrier needed to screen out hydro genions from all the other ions floating around in the solution. This glass is chemically doped with lithium ions, which is what makes it react electrochemically to hydrogen ions. Of course, glass is not exactly what you would call a "conductor;" rather, it is an extremely good insulator This presents a major problem if our intent is to measure voltage between the two electrodes. The circuit path from one electrode contact, through the glass barrier, through the solution, to the other electrode, and back through the other electrode's contact, is one of extremely high resistance.

The other electrode (called the reference electrode) is made from a chemical solution of neutral (7) pH buffer solution (usually potassium chloride) allowed to exchange ions with the process solution through a porous separator, forming a relatively low resistance connection to the test liquid. At first, one might be inclined to ask: why not just dip a metal wire into the solution to get an electrical connection to the liquid? The reason this will not work is because metals tend to be highly reactive in ionic solutions and can produce a significant voltage across the interface of metal-to-liquid contact. The use of a wet chemical interface with the measured solution is necessary to avoid creating such a voltage, which of course would be falsely interpreted by any measuring device as being indicative of pH.

Here is an illustration of the measurement electrode's construction. Note the thin, lithium-doped glass membrane across which the pH voltage is generated:

Here is an illustration of the reference electrode's construction. The porous junction shown at the bottom of the electrode is where the potassium chloride buffer and process liquid

interface with each other.

The measurement electrode's purpose is to generate the voltage used to measure the solution's pH. This voltage appears across the thickness of the glass, placing the silver

wire on one side of the voltage and the liquid solution on the other. The reference electrode's purpose is to provide the stable, zero-voltage connection to the liquid solution so that a complete circuit can be made to measure the glass electrode's voltage. While the reference electrode's connection to the test liquid may only be a few kilo-ohms, the glass electrode's resistance may range from ten to nine hundred mega-ohms, depending on electrode design! Being that any current in this circuit must travel through both electrodes' resistances (and the resistance presented by the test liquid itself), these resistances are in series

An ordinary analog or even digital voltmeter has much too low of an internal resistance to measure voltage in such a high-resistance circuit. The equivalent circuit diagram of a typical pH probe circuit illustrates the problem:

Even a very small circuit current travelling through the high resistances of each component in the circuit (especially the measurement electrode's glass membrane), will produce relatively substantial voltage drops across those resistances, seriously reducing the voltage seen by the meter. Making matters worse is the fact that the voltage differential generated by the measurement electrode is very small, in the milli volt range (ideally 59.16 milli volts per pH unit at room temperature). The meter used for this task must be very sensitive and have an extremely high input resistance. The most common solution to this measurement problem is to use an amplified meter with an extremely high internal resistance to measure the electrode voltage, so as to draw as little current through the circuit as possible. With modern semiconductor components, a voltmeter with an input resistance of up to 1017 O can be built with little difficulty.

Another approach, seldom seen in contemporary use, is to use a potentiometric "null-balance" voltage measurement setup to measure this voltage without drawing any current from the circuit under test. If a technician desired to check the voltage output between a pair of pH electrodes, this would probably be the most practical means of doing so using only standard benchtop metering equipment:

As usual, the precision voltage supply would be adjusted by the technician until the null

detector registered zero, then the voltmeter connected in parallel with the supply would be viewed to obtain a voltage reading. Wiring requirements for pH electrodes tend to be even more severe than thermocouple wiring, demanding very clean connections and short distances of wire (10 yards or less, even with gold-plated contacts and shielded cable) for

accurate and reliable measurement. As with thermocouples, however, the disadvantages of electrode pH measurement are offset by the advantages: good accuracy and relative technical simplicity.

Few instrumentation technologies inspire the awe and mystique commanded by pH measurement, because it is so widely misunderstood and difficult to troubleshoot. Without elaborating on the exact chemistry of pH measurement, a few words of wisdom can given here about pH measurement systems.

TEMPERATURE SENSOR:

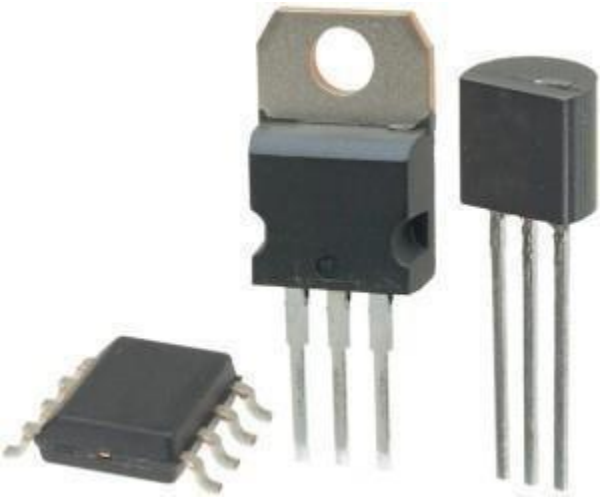


Fig 4.5

The most commonly measured physical parameter is temperature whether in process industry applications or in laboratory settings. Exact measurements are critical part of success. Exact measurements are needed for many applications such as medical applications, materials research in labs, studies of electronic or electrical components, biological research, and geological studies. Most commonly, temperature sensors are used to measure temperature in circuits which control a variety of equipment's.

There are different types of temperature sensors used in the market today including resistance temperature detectors (RTDs), thermocouples, thermistors, infrared sensor, and semiconductor sensors. Each of them has particular operating parameters. These sensors come in different varieties, but have one common thing: they all measure temperature by sensing a change in the physical characteristic.

A temperature sensor is a device, usually an RTD (resistance temperature detector) or a thermocouple, that collects the data about temperature from a particular source and

converts the data into understandable form for a device or an observer. Temperature sensors are used in many applications like HVand AC system environmental controls, food processing units, medical devices, chemical handling and automotive under the hood monitoring and controlling systems, etc.

The most common type of temperature sensor is a thermometer, which is used to measure temperature of solids, liquids and gases. It is also a common type of temperature sensor mostly used for non-scientific purposes because it is not so accurate.

Types of Temperature Sensors

There are different types of temperature sensors that have sensing capacity depending upon their range of application. Different types of temperature sensors are as follows:

- Thermocouples
- Resistor temperature detectors
- Thermistors
- Infrared sensors
- Semiconductors
- Thermometers

HUMIDITY SENSOR:

Sometimes, after rains, the air feels moist. The water seems to have suspended in the air. However, in certain AC's you click some buttons and the atmosphere brightens up. How and why does it all happen? Moisture forms up in the air, resulting in humidity. However, the humidity sensor in your AC picks it up and cleans it up for you. Isn't that wonderful? Let's take a look at how it does that.

A humidity sensor (or hygrometer) senses, measures and reports both moisture and air temperature. The ratio of moisture in the air to the highest amount of moisture at a particular air temperature is called relative humidity. Relative humidity becomes an important factor when looking for comfort. Humidity sensors work by detecting changes that alter electrical currents or temperature in the air.

- Capacitive
- Resistive

- Thermal

All three types of sensors monitor minute changes in the atmosphere in order to calculate the humidity in the air. Let us discuss these types in detail:

Accuracy

Every sensor has its own calibration curve, based on a 9 point system. It basically pitches the pros against the cons of the particular sensor.

Linearity

It indicates the voltage deviation from the BFSL value and the measured output voltage value, converted to relative humidity.

Reliability

The measurements often cause the sensor to fall out of sync. However for a sensor to be useful, it has to provide reliable measurements.

Repeatability

The measurements from a sensor, have to be so that they don’t drift apart. Repeatability is the measurement of drift among measurements of a single quantity.

Response time

Typically, the time is taken by a sensor to rise to 66% (rise time) or fall to 33% (fall time) of maximum output voltage, is known as the response time.

Applications of humidity sensors are far and wide

The applications of humidity sensor range far and wide. People with illnesses affected by humidity, monitoring and preventive measure in homes employ humidity sensors. A

humidity sensor is also found as part of home heating, ventilating and air conditioning systems (HVAC systems). These are also used in offices, cars, humidors, museums,

industrial spaces and greenhouses and are also used in meteorology stations to report and predict the weather.

We list out some projects below for reference:

Humidity Indicator

Presented here are a simple humidity indicator and controller. In industries such as textile, the change in moisture content has a direct impact on the properties of fabric, such as tensile strength, elasticity, fibre diameter and friction. Cotton and linen require high relative humidity (RH) levels of around 70-80 per cent since they are very brittle. Wool requires RH levels of around 65 per cent. While silk requires between 65 and 70 per cent. With this circuit, you can not only monitor humidity levels between 30 and 90 per cent RH but also control it.

LDR SENSOR:

Image:

FIG 4.6

An LDR (Light dependent resistor), as its name suggests, offers resistance in response to the ambient light. The resistance decreases as the intensity of incident light increases, and vice versa. In the absence of light, LDR exhibits a resistance of the order of mega-ohms which decreases to few hundred ohms in the presence of light. It can act as a sensor, since

a varying voltage drop can be obtained in accordance with the varying light. It is made up of cadmium sulphide (CdS).

An LDR has a zigzag cadmium sulphide track. It is a bilateral device, *i.e.*, conducts in both directions in same fashion.

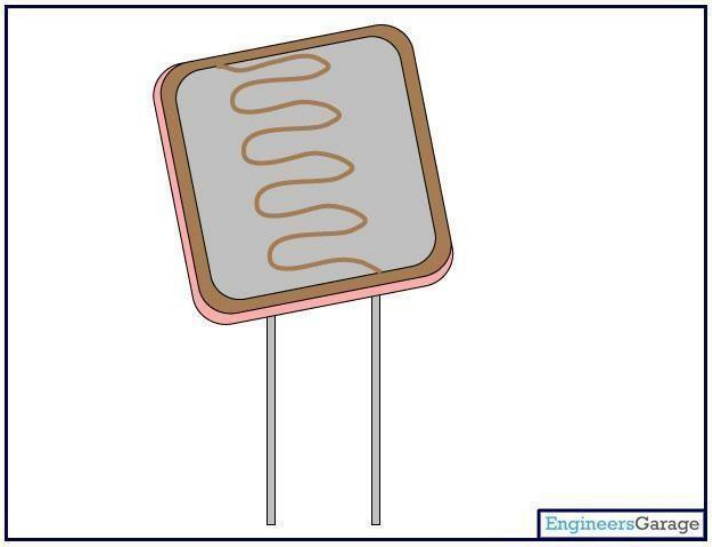


FIG 4.7

LCD 16X2:

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments),animations ansoon.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

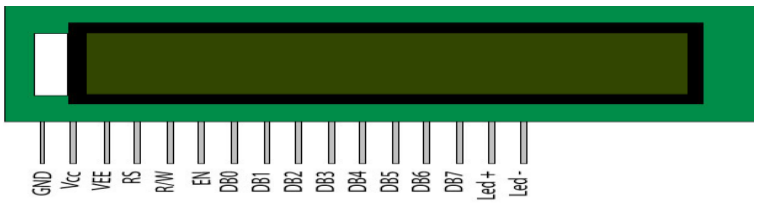


FIG 4.8

DC MOTOR:

Fig 4.3

Motors are electromechanical devices which use the interaction of magnetic fields and DC conductors to convert the electrical energy into rotary mechanical energy **Electrical DC Motors** are continuous actuators that convert electrical energy into mechanical energy. The DC motor achieves this by producing a continuous angular rotation that can be used to rotate pumps, fans, compressors, wheels, etc.

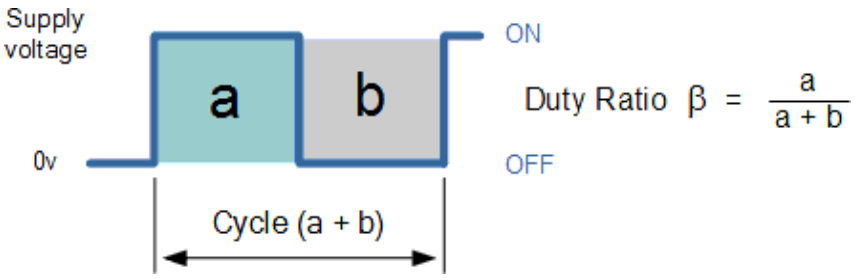
As well as conventional rotary DC motors, linear motors are also available which are capable of producing a continuous liner movement. There are basically three types of conventional electrical motor available: AC type Motors, DC type Motors and Stepper Motors.A Typical Small DC Motor

AC Motors are generally used in high power single or multi-phase industrial applications were a constant rotational torque and speed is required to control large loads such as fans or pumps.

In this tutorial on electrical motors we will look only at simple light duty **DC Motors** and **Stepper Motors** which are used in many different types of electronic, positional control, microprocessor, PIC and robotic type circuits.

Pulse Width Speed Control

We said previously that the rotational speed of a DC motor is directly proportional to the mean (average) voltage value on its terminals, and the higher this value, up to maximum allowed motor volts, the faster the motor will rotate. In other words more voltage more speed. By varying the ratio between the "ON" (t_{ON}) time and the "OFF" (t_{OFF}) time durations, called the "Duty Ratio", "Mark/Space Ratio" or "Duty Cycle", the average value of the motor voltage and hence its rotational speed can be varied. For simple unipolar drives the duty ratio β is given as:



and the mean DC output voltage fed to the motor is given as: Vmean = β x Vsupply. Then by varying the width of pulse a, the motor voltage and hence the power applied to the motor can be controlled and this type of control is called **Pulse Width Modulation** or **PWM**.

Another way of controlling the rotational speed of the motor is to vary the frequency (and hence the time period of the controlling voltage) while the "ON" and "OFF" duty ratio times are kept constant. This type of control is called **Pulse Frequency Modulation** or **PFM**.

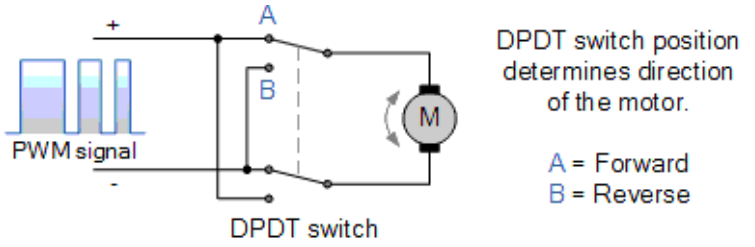
With pulse frequency modulation, the motor voltage is controlled by applying pulses of variable frequency for example, at a low frequency or with very few pulses the average voltage applied to the motor is low, and therefore the motor speed is slow. At a higher frequency or with many pulses, the average motor terminal voltage is increased and the motor speed will also increase.

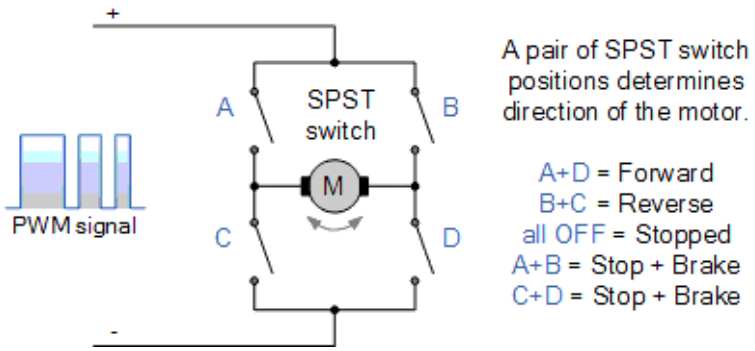
Then, Transistors can be used to control the amount of power applied to a DC motor with the mode of operation being either "Linear" (varying motor voltage), "Pulse Width Modulation" (varying the width of the pulse) or "Pulse Frequency Modulation" (varying the frequency of the pulse).

Reversing the Direction of a DC Motor

While controlling the speed of a DC motor with a single transistor has many advantages it also has one main disadvantage, the direction of rotation is always the same, its a "Uni-directional" circuit. In many applications we need to operate the motor in both directions forward and back. To control the direction of a DC motor, the polarity of the DC power applied to the motor's connections must be reversed allowing its shaft to rotate in the opposite direction. One very simple and cheap way to control the rotational direction of a DC motor is to use different switches arranged in the following manner.

DC Motor Directional Control





The first circuit uses a single double-pole, double-throw (DPDT) switch to control the polarity of the motors connections. By changing over the contacts the supply to the motors terminals is reversed and the motor reverses direction. The second circuit is slightly more complicated and uses four single-pole, single-throw (SPST) switches arranged in an "H" configuration.

The mechanical switches are arranged in switching pairs and must be operated in a specific combination to operate or stop the DC motor. For example, switch combination A + D controls the forward rotation while switches B + C control the reverse rotation as shown.

Switch combinations A + B or C + D shorts out the motor terminals causing it to brake quickly. However, using switches in this manner has its dangers as operating switches A +

C or B + D together would short out the power supply.

While the two circuits above would work very well for most small DC motor applications, do we really want to operate different combinations of mechanical switches just to reverse the direction of the motor, NO!. We could change the manual switches for set of Electromechanical Relays and have a single forward-reverse button or switch or even use a solid state CMOS 4066B quad bilateral switch. But another very good way of achieving bi-directional control of a motor (as well as its speed) is to connect the motor into a **Transistor H-bridge** type circuit arrangement as shown below.

12V BATTERY:

FIG 4.4

Lead acid batteries used in the RV and Marine Industries usually consist of two 6-volt batteries in series, or a single 12-volt battery. These batteries are constructed of several single cells connected in series each cell produces approximately 2.1 volts. A six-volt battery has three single cells, which when fully charged produce an output voltage of 6.3 volts. A twelve-volt battery has six single cells in series producing a fully charged output voltage of 12.6 volts.

A battery cell consists of two lead plates a positive plate covered with a paste of lead dioxide and a negative made of sponge lead, with an insulating material (separator) in

between. The plates are enclosed in a plastic battery case and then submersed in an electrolyte consisting of water and sulfuric acid (see figure # 1). Each cell is capable of

storing 2.1 volts.

In order for lead acid cell to produce a voltage, it must first receive a (forming) charge voltage of at least 2.1-volts/cell from a charger. Lead acid batteries do not generate voltage on their own; they only store a charge from another source. This is the reason lead acid batteries are called storage batteries, because they only store a charge. The size of the battery plates and amount of electrolyte determines the amount of charge lead acid batteries can store. The size of this storage capacity is described as the amp hour (AH) rating of a battery. A typical 12-volt battery used in a RV or marine craft has a rating 125 AH, which means it can supply 10 amps of current for 12.5 hours or 20-amps of current for a period of 6.25 hours. Lead acid batteries can be connected in parallel to increase the total AH capacity.

In figure # 2 below, six single 2.1-volt cells have been connected in series to make the typical 12-volt battery, which when fully charged will produce a total voltage of 12.6-volts.

Lead Acid Battery Recharge Cycle

The most important thing to understand about recharging lead acid batteries is that a converter/charger with a single fixed output voltage will not properly recharge or maintain your battery. Proper recharging and maintenance requires an intelligent charging system that can vary the charging voltage based on the state of charge and use of your RV or Marine battery. Progressive Dynamics has developed intelligent charging systems that solve battery problems and reduce battery maintenance.

The discharged battery shown in figure # 6 on the next page is connected to a converter/charger with its output voltage set at 13.6-volts. In order to recharge a 12-volt lead acid battery with a fully charged terminal voltage of 12.6-volts, the charger voltage must be set at a higher voltage. Most converter/chargers on the market are set at approximately 13.6-volts. During the battery recharge cycle **lead sulfate (sulfation)** begins to reconvert to lead and sulfuric acid.

During the recharging process as electricity flows through the water portion of the

electrolyte and water, (H2O) is converted into its original elements, hydrogen and oxygen. These gasses are very flammable and the reason your RV or Marine batteries must be vented outside. Gassing causes water loss and therefore lead acid batteries need to have

water added periodically. Sealed lead acid batteries contain most of these gasses allowing them to recombine into the electrolyte. If the battery is overcharged pressure from these gasses will cause relief caps to open and vent, resulting in some water loss. Most sealed batteries have extra electrolyte added during the manufacturing process to compensate for some water loss. The battery shown in figure # 7 above has been fully recharged using a fixed charging voltage of 13.6-volts. Notice that some **lead sulfate (sulfation)** still remains on the plates. This build-up will continue after each recharging cycle and gradually the battery will begin to loose capacity to store a full charge and eventually must be replaced. **Lead sulfate** build up is reduced if battery is given an **Equalizing Charge** once every 10 discharge cycles or at least once a month. An **Equalizing Charge**increases charging voltage to 14.4 volts or higher for a short period. This higher voltage causes gassing that equalizes (re-mixes) the electrolyte solution.

Since most RV and Marine craft owners seldom remember to perform this function, Progressive Dynamics has developed the microprocessor controlled Charge Wizard. The Charge Wizard will automatically provide an **Equalizing Charge** every 21 hours for a period of 15 minutes, when the battery is fully charged and not in use. Our 2000 Series of Marine Battery Chargers have the **Charge Wizard** feature built-in.

One disadvantage of recharging a lead acid battery at a fixed voltage of 13.6-volts is the recharge time is very long. A typical 125-AH RV or Marine battery will take approximately 80 hours to recharge at 13.6 volts. Increasing the charge voltage to 14.4-volts will reduce battery recharge time for a 125-AH battery to 3-4 hours. Once a battery reaches 90% of full charge, the**voltage must be reduced** from 14.4-volts to 13.6-volts to reduce gassing and water loss. The optional **Charge Wizard**automatically senses when a battery has a very low state of charge and automatically selects its **BOOST MODE** of operation. **BOOST MODE** increases the voltage of a **PD9100 Series** converter/charger to 14.4 volts. When the battery reaches the 90% charge level, the **Charge Wizard** automatically reduces the charge voltage down to 13.6 volts to complete the charge. Again, this is a standard feature on our Marine Chargers.

Another disadvantage of recharging a lead acid battery at a fixed voltage of 13.6-volts is

that once it is fully charged, 13.6 volts will cause considerable gassing and water loss. To prevent this from occurring the charging voltage must be reduced to 13.2-volts. The Charge Wizard will automatically select its STORAGE MODE of operation (13.2-volts) once the battery reaches full charge and remains unused for a period of 30 hours. This

feature is standard on all of Progressive Dynamics Marine Battery Chargers.

At a charging voltage of 13.2 volts, the converter/charger will maintain a full charge, reduce gassing and water loss. However, this lower voltage does not provide enough gassing to prevent a battery condition called **Battery Stratification. Battery Stratification** is caused by the fact that the electrolyte in the battery is a mixture of water and acid and, like all mixtures, one component, the acid, is heavier than water. Therefore, acid will begin to settle and concentrate at the bottom of the battery (see figure #8)

This higher concentration of acid at the bottom of the battery causes additional build-up of **lead sulfate (sulfation)**, which reduces battery storage capacity and battery life. In order to prevent **Battery Stratification**, an **Equalization Charge**(increasing charging voltage to 14.4-volts) must be applied periodically. The **Charge Wizard** automatically selects its **EQUALIZATION MODE** (14.4 volts) every 21 hours for a period of 15 minutes. This **Equalizing Charge** feature is standard on our Marine chargers.

As you have learned, in order to properly charge and maintain a lead acid battery you must use an intelligent charging system. Progressive Dynamics, Inteli-Power 9100 Series RV converters with a **Charge Wizard** installed, or one of our Inteli-Power Marine Battery Chargers will provide the intelligent charging system your battery needs for a long life, with low maintenance.

ZIGBEE:

Fig 4.5

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such

as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection.

Hence, Zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network.

The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or more general wireless networking such as Wi-Fi. Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that require short-range low-rate wireless data transfer.

Its low power consumption limits transmission distances to 10–100 meters line-of-sight, depending on power output and environmental characteristics.[2] Zigbee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. Zigbee is typically used in low data rate applications that require long battery life and secure networking (Zigbee networks are secured by 128 bit symmetric encryption keys.) Zigbee has a defined rate of 250 kbit/s, best suited for intermittent data transmissions from a sensor or input device. Zigbee was conceived in 1998, standardized in 2003, and revised in 2006. The name refers to the waggle dance of honey bees after their return to the beehive.

Zigbee is a low-cost, low-power, wireless mesh network standard targeted at battery-powered devices in wireless control and monitoring applications. Zigbee delivers low-latency communication. Zigbee chips are typically integrated with radios and with microcontrollers. Zigbee operates in the industrial, scientific and medical (ISM) radio bands: 2.4 GHz in most jurisdictions worldwide; though some devices also use 784 MHz in China, 868 MHz in Europe and 915 MHz in the US and Australia, however even those regions and countries still use 2.4 GHz for most commercial Zigbee devices for home use. Data rates vary from 20 kbit/s (868 MHz band) to 250 kbit/s (2.4 GHz band).

Zigbee builds on the physical layer and media access control defined in IEEE standard 802.15.4 for low-rate wireless personal area networks (WPANs). The specification includes four additional key components: network layer, application layer, *Zigbee Device Objects* (ZDOs) and manufacturer-defined application objects. ZDOs are responsible for some tasks, including keeping track of device roles, managing requests to join a network, as well as device discovery and security.

The Zigbee network layer natively supports both star and tree networks, and generic mesh networking. Every network must have one coordinator device. Within star networks, the coordinator must be the central node. Both trees and meshes allow the use of

Zigbee routers to extend communication at the network level. Another defining feature of Zigbee is facilities for carrying out secure communications, protecting establishment and transport of cryptographic keys, ciphering frames, and controlling device. It builds on the basic security framework defined in IEEE 802.15.4.

MOTOR DRIVER(L23D):

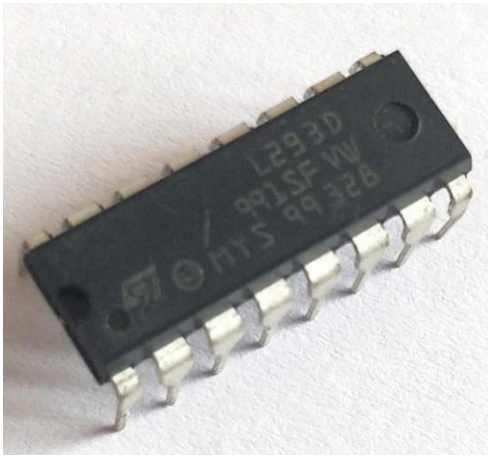
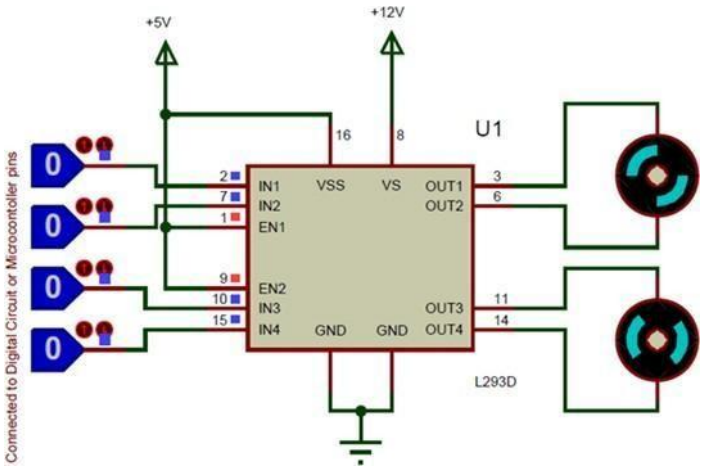


Fig 4.5

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Micron rollers like Arduino, PIC, ARM etc.. this IC will be the right choice for you.

Using this L293D motor driver IC is very simple. The IC works on the principle of Half H-Bridge, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a set up which is used to run motors both in clock wise and anti clockwise direction. As said earlier this IC is capable of running two motors at the any direction at the same time, the circuit to achieve the same is shown below.



All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs(Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

Input 1 = LOW(0v)	Output 1 = LOW	Motor 1 rotates in Anti-Clock wise Direction
Input 2 = HIGH(5v)	Output 2 = HIGH	
Input 3 = LOW(0v)	Output 1 = LOW	
Input 4 = HIGH(5v)	Output 2 = HIGH	

Input 1 = HIGH(5v)	Output 1 = HIGH	Motor 2 rotates in Anti - Clock wise Direction
Input 2 = HIGH(5v)	Output 2 = HIGH	
Input 3 = HIGH(5v)	Output 1 = LOW	
Input 4 = HIGH(5v)	Output 2 = HIGH	

The Enable pins (Enable 1,2 and Enable 3,4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1,2 are used to control the motor 1 and Input pins 3,4 are used to control the Motor 2. The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor. You can toggle the input pins based on the following table to control your motor.

Input 1 = HIGH(5v)	Output 1 = HIGH	Motor 1 rotates in Clock wise Direction
Input 2 = LOW(0v)	Output 2 = LOW	
Input 3 = HIGH(5v)	Output 1 = HIGH	Motor 2 rotates in Clock wise Direction
Input 4 = LOW(0v)	Output 2 = LOW	

Applications

- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED's can be driven
- Relay Driver module (Latching Relay is possible)

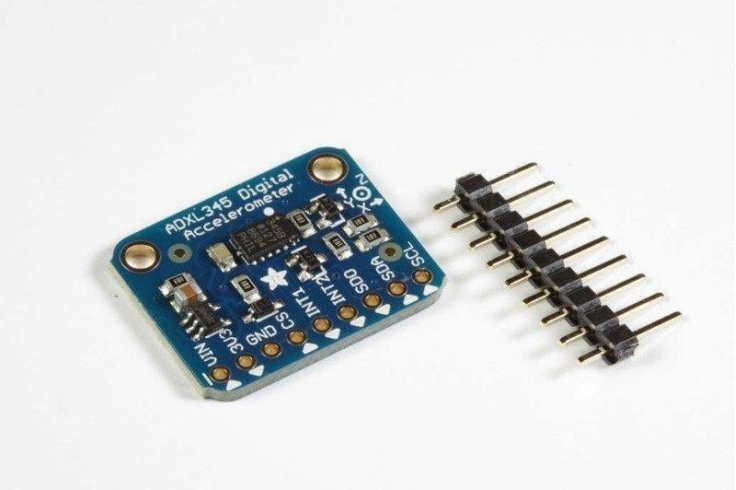
ADXL sensor:

The ADXL345 is a low-power, 3-axis MEMS accelerometer modules with both I2C and SPI interfaces. The Adafruit Breakout boards for these modules feature on-board 3.3v voltage regulation and level shifting which makes them simple to interface with 5v microcontrollersuchastheArduino.

The ADXL345 features 4 sensitivity ranges from +/- 2G to +/- 16G. And it supports output data rates ranging from 10Hz to 3200Hz.

MEMS - Micro Electro-Mechanical Systems

The sensor consists of a micro-machined structure on a silicon wafer. The structure is suspended by polysilicon springs which allow it to deflect smoothly in any direction when subject to acceleration in the X, Y and/or Z axis. Deflection causes a change in capacitance between fixed plates and plates attached to the suspended structure. This change in capacitance on each axis is converted to an output voltage proportional to the acceleration on that axis.



SOFTWARE REQUIREMENTS
ARDUINO IDE:

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. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use 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 Ivrea 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.

WRITING SKETCHES

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

FILE

1. New:

Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

2. Open:
   Allows loading a sketch file browsing through the computer drives and folders.
3. OpenRecent:
   Provides a short list of the most recent sketches, ready to be opened.
4. Sketchbook:
   shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
5. Examples:
   Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
6. Close:
   Closes the instance of the Arduino Software from which it is clicked.
7. Save:
   Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.

8. Saveas:
   Allows saving the current sketch with a different name.
9. PageSetup:
   It shows the Page Setup window for printing.
10. Print:
    Sends the current sketch to the printer according to the settings defined in Page Setup.
11. Preferences:
    Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
12. Quit:
    Closes all IDE windows. The same sketches open when Quit was chosen will be

automatically reopened the next time you start the IDE.

EDIT:

1. Undo/Redo

   Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

2. Cut

   Removes the selected text from the editor and places it into the clipboard.

3. Copy

   Duplicates the selected text in the editor and places it into the clipboard.

4. CopyforForum

   Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

5. CopyasHTML

   Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

6. Paste

   Puts the contents of the clipboard at the cursor position, in the editor.

7. SelectAll

   Selects and highlights the whole content of the editor.

8. Comment/Uncomment

   Puts or removes the // comment marker at the beginning of each selected line.

9. Increase/Decrease/Indent

   Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

10. Find

    Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

11. FindNext

    Highlights the next occurrence - if any - of the string specified as the search item in

the Find window, relative to the cursor position.

12. FindPrevious

    Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

13. SketchVerify/Compile

    Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

14. Upload

    Compiles and loads the binary file onto the configured board through the configured Port.

15. UploadUsingProgrammer

    This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.

16. ExportCompiledBinary

    Saves a .hex file that may be kept as archive or sent to the board using other tools

17. ShowSketchFolder

    Opens the current sketch folder.

18. IncludeLibrary

    Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

19. AddFile...

    Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

TOOLS

1. AutoFormat

   This formats your code nicely: i.e. indents it so that opening and closing curly

braces line up, and that the statements inside curly braces are indented more.

2. ArchiveSketch

   Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

3. FixEncoding&Reload

   Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

4. SerialMonitor

   Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

5. Board

   Select the board that you're using. See below for descriptions of the various boards.

6. Port

   This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

7. Programmer:

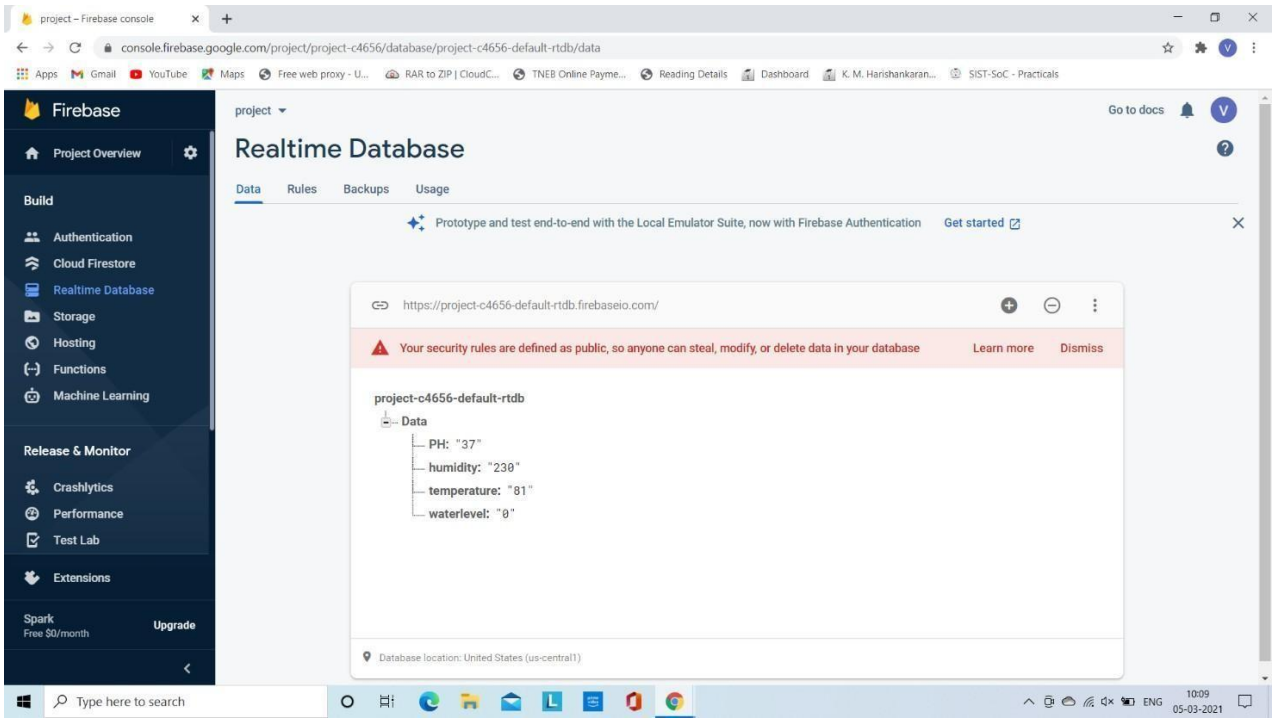
   For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a boot loader to a new microcontroller, you will use this.

8. Burn/Boot/loader:

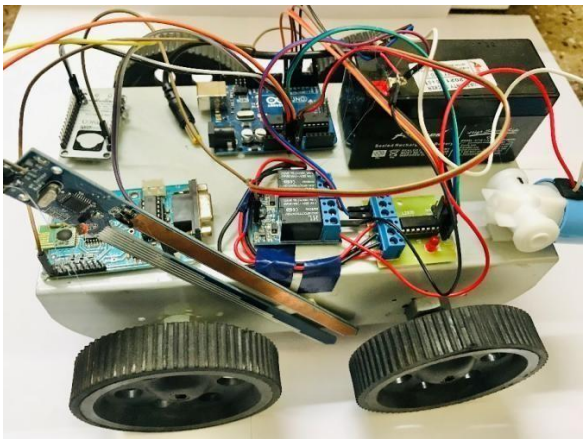
   the items in this menu allow you to burn a boot loader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuine board but is useful if you purchase a new AT mega microcontroller (which normally come without a boot loader). Ensure that you've selected the correct board from the Boards menu before burning the boot loader on the target board. This command also set the right fuses.

CHAPTER 5 OUTPUT

Firebase output:



GESTURE CONTROL



ROBOT SETUP

APP OUTPUT

CHAPTER 6

FUTURE WORK:

Our project can be improvised by using a sensor to note the soil moisture value such that usage of unnecessary Fertilizers can be reduced. A water meter can be installed to estimate the amount of water used for irrigation and thus giving cost estimation. Further, it also reduces the investment of farmers.

CONCUSION:

The sensors are successfully interfaced with raspberry pi and wireless communication is achieved. All observations and experimental tests prove that this project is a complete solution to the field activities irrigation problems. Implementation of such a system in the field can definitely help to improve the yield of the crops and aids to manage the water resources effectively reducing the wastage.

The use of automated monitoring and management systems are gaining increasing demand with technological advancement. In agricultural field loss of yield mainly occurs due to widespread disease. Mostly the detection and identification of the disease are noticed when the disease advances to the severe stage. Therefore, causing the loss in terms of yield, time and money. The proposed system is capable of detecting the disease at the earlier stage as soon as it occurs on the leaf. Hence saving the loss and reducing the dependency on the expert to a certain extent is possible. It can provide help for a person having less knowledge about the disease. Depending on these goals, we have to extract the features corresponding to the disease.

REFERENCES:

[1] V.C. Patil, K.A. Al-Gaadi, D.P. Biradar and M. Rangaswamy, "INTERNET OF THINGS (IOT) AND CLOUD COMPUTING FOR AGRICULTURE: AN OVERVIEW' Pecision Agriculture Research Chair, King Saud University, Riyadh, Saudi Arabia, Precision Agriculture Research Chair, Department of Agricultural Engineering, College of Food and Agricultural Sciences, King Saud University, Riyadh, Saudi Arabia University of Agricultural Sciences, Dharwad, Karnataka.

[2] Wu X, Zhu X, Wu G Q, Data Mining with Big Data[J]. IEEE Transactions on Knowledge & Data Engineering, 2014, 26(1):97-107.

[3] Tsai C W, Lai C F, Chiang M C, Data Mining for Internet of Things: A Survey[J]. Communications Surveys & Tutorials, IEEE 2014, 16(1):77 – 97.

[4] Nikesh Gondchwar and Prof. Dr. R.S.Kawitkar, "IOT based Smart Agriculture", International journal Of Advanced research in computer and Communication Engineering (IJARCCE), Vol. 5, Issue6, June 2016.

[5] Raspberry Pi Product Specification (Accessed 2017, December 10).

[6] T. L. Vu, T. Nguyen and Y. M. Jang, "MQTT Protocol for Connected OCC Small Cells," in Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference on, July 2017.

[7] Andrew Meola, (Accessed 2017, December 10). Why IoT, big data & smart farming are the future of agriculture.

[8] MQTT Essentials Part 3: Client, Broker and Connection Establishment (Accessed 2017, December 10).

[9] H. Saini, A. Thakur, S. Ahuja, N. Sabharwal, and N. Kumar, "Arduino based automatic wireless weather station with remote graphical application and alerts," in 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Feb 2016, pp. 605– 609.

[10] NodeMCU Features Index (Accessed 2017, December 10).