1.1 INTRODUCTION

Though comparatively inferior to some countries, weather forecasts can still prove extremely crucial in taking vital decisions in India. Average weather forecast authenticity nears 80% and incorporating their guidance is better than trusting intuition which has only a 50% probability of coming true (analogy to tossing a coin). Android application and real-time data processing has played an important role in weather forecasting. Recently different computational algorithms are being developed for processing accurate data at a faster rate. A real-time forecasting system based on microcontroller and windows operating system using a database language will be developed in this project. Global data processing and forecasting system(GDPFS) strengthens the capability of members to meets the need of users by sharing numerical weather predictions (NWP) products and android application using sensors will help in avoiding the manual errors for short range up to 72 Hours data management.

1.2 BACKGROUND

Even after a decade of being around, the weather industry in India is still nascent, facing many handicaps and deterrents. One of the major obstacles that weather forecasters confront are ill-suited expectations and methods. Weather forecasting usually has a lot of variables which fluctuate over time. First, you have to take reliable data as input. Reliable data in weather prediction generally means a huge density of data. It's not sufficient, for example, to just have an instrument measuring a certain value at a certain area. You often have to get an idea of the gradient of those i.e. how fast things are changing and sometimes, even a gradient of gradients. The second facet is predicting future weather based on present data. This requires huge amount of computing power, along with really good algorithms. India still uses, for the most part, old algorithms and highly outdated computing systems. Hence, their processing power is extremely low compared to what is available now. Hence, the inaccurate predictions. Forecast systems, particularly those dealing with short-term forecasts, generally use numerical weather prediction methods based on sets of mathematical equations on the behavior of the atmosphere. There are four types of weather forecasts. Now-casting is about predicting weather conditions up to 24 hours from the current time. Short range forecasts are valid up to 72 hours ahead while medium range forecasts predict conditions for a period of 4 to 10 days. Monthly and seasonal forecasts like predictions about the arrival of monsoons, come under long range forecasting, that is the prediction of average weather conditions for a minimum of 30 days and up to an entire season. The equations dealing with various atmospheric phenomena like land/ sea temperature, rain, wind etc. are combined into complex mathematical models. Although recent advances in satellite and computer technology have helped in significantly improving weather forecasting.

The cloud computing system will make the process less complex, as it will also provide the previous weather details helping computational algorithms to access it anytime at any place, also it will increase the range of period predicting weather conditions as large data can be stored on cloud making data management easier.System designed here will overcome the problem of taking manual readings for short range forecasts up to 72 hours, the manual system will be automated using the sensors and an android application which will automatically display the readings for every 72 hours and hence manual reading errors will be avoided.

In spite of everything, the experience of Indian users with respect to weather forecasts is on a correction path, riding on evolutionary changes ignited by private enterprise.

1.3 RELEVANCE

The Indian Meteorology department collecting data with help of analog sensors had been a long term activity, displaying the results to the citizens in that particular location is done through a long process where one has to wait for the conditions to be forecasted, so now a days an easy way to get the weather condition clear is to display them onto their personal devices which is done with the help of an android application ,run by an online data system which is keep updated with the help of data from the sensors such as hygroclip and wind vane sensors with the help of the microcontroller updating the data to the online database every single minute.

Real time data logging also has many facets involved which are stated below, in order to keep them as low as possible many changes are to be done in the system such as to get noiseless sensor output we need to condition the output signal from the sensor properly, to reduce the time error the controller being used should have a fast processing speed etc.

1.4 PROJECT UNDERTAKEN

Android application is designed for weather monitoring using mainly 2 sensors *viz.* hygroclip sensor which determines the temperature and humidity, second is wind vane sensor which gives the direction of wind.

Following steps were followed in order to design the required system: -

1)Interfacing of the sensors- hygroclip and wind vane sensors were interfaced with the controller with the help of signal conditioning circuit.

2)Signal Conditioning- a) For the hygroclip sensor which determines temperature and humidity has inbuilt signal conditioning, so external signal conditioning for sensor’s output signal isn’t required.

b) op-amp lm324 was used for conditioning the output signal from the circuit containing wind vane in it.

3)microcontroller- Arduino was used as the microcontroller to interface the sensors with.

4)Internet – ESP8266 Node MCU was used to connect the system till Arduino with the internet, it is first interfaced with the controller here Arduino, to receive data so that it can send it over the server.

5)Database- Google database *viz*, Firebase is used to store the data from the sensors online and then transfer it to the android application.

6)Android application- Eclipse software is used to make an android application.

1.5 ORGANIZATION OF PROJECT REPORT

Chapter 1: Introduction

In first chapter introduction about the concepts and technologies associated with weather forecasting are stated also the problems faced by the weather forecasting department are mentioned. The contribution of the existing method in the project which will be designed is stated along with the challenges faced by the present technology and the block diagram explaining the all over working of the system is drawn.

Chapter 2: Literature Survey

Literature Survey is the chapter which covers all the information gathered together. It consists of overview of what is to be done in project and any kind of traces to pave off the exact job to be completed. Papers and journals referred are mentioned along with the comparison table for microcontrollers is drawn. Detailed block diagram of system is also drawn.

Chapter 3: Design Aspect

All the sensors used and there detailed working is mentioned along with the software which will be used to complete the process is described. The specifications like voltage, accuracy and precision of all the sensors is written.

Chapter 4: Manufacturing

This chapter discusses about the steps required for hardware manufacturing in detail.

Chapter 5: Experimentations

This chapter discusses about the changes made in components used project in detail.

Chapter 6: Result and Conclusion

All the papers and websites referred to work on the project are mentioned.

1.6 SUMMARY

Looking at the major challenges faced by the present weather forecasting system in India due to the manual handling of short range up to 72 Hours data management, complex computational algorithms, outdated software used and also due to the non-linearity in the inaccurate weather predictions due to India’s geographical location, it has become necessary for the weather sectors in the country to enhance automation and make use of updated and sophisticated software for weather prediction. The major challenge in this sector includes the inefficiency of the system due to lack of automation which introduces manual error giving rise to bigger problems in predicting weather conditions, the system designed will be completely automated based on the given weather sensors, microcontroller, a cloud computing system for data recollection and an android application for easier access of data rather than taking readings manually every 72 hours. It overcomes the problem of meteorological department by giving them a complete automated data reading system in their premises, making it easier for them to take readings time efficiently and error free readings with the help of system designed.

2.1 INTRODUCTION

Literature Survey is the chapter which covers all the information gathered together. It consists of overview of what is to be done in project and any kind of traces to pave off the exact job to be completed. With the help of Literature Survey, we can come to final conclusion about what is the exact problem arising to the current scenario we are going to work for. This traces different sources such as Conference papers, Internet, Magazines and other available resources too. In this chapter, we are covering what we have done as a research to settle to certain conclusion of how further stepping is going to happen regarding the topic of Android application development for weather forecasting using cloud computing.

2.2 PAPERS/JOURNALS REFERRED

“Orientation correction of wind direction measurements by means of staring lidar” by Michael Schmidt, Juan Jose Trujillo and Martin Kuhn, Michael Schmidt [1]

In spite of the efforts made at the time of installation of wind vanes or ultrasonic anemometers (Sonic), there is always a remaining uncertainty of several degrees in the absolute north of such sensors. In this research a method is presented to reduce the azimuthal orientation error of wind direction sensors by means of Doppler Lidar measurements. The method is based on the comparison between the conventional sensor and a distant long range lidar pointing to it in staring mode. By comparing their line-of-sight wind speeds any misalignment between both systems can be estimated more accurately. This method was applied in a measurement campaign in the offshore wind farm alpha ventus next to the meteorological mast FINO1. The maximum alignment error of a Sonic was reduced to below +-1 degree. This accurate alignment has asserted, that no bias exists between Lidar and Sonic wind speed measurements.

“Design & Implementation of an Efficient Windmill Anemometer for Wind Speed Measurement Using Microcontroller” by Sree Bash Chandra Debnath, Biki Barua, Rezaul Karim and Muslim Uddin faisal [2]

Direction & speed, these are the important parameters to characterize the wind condition of the environment. In our paper, we tried to build a microcontroller based efficient anemometer. Though, in the market, there are several models is existed for anemometer, in our design we have used microcontroller PIC 18F2550 to control the circuit that’s making the design more efficient & cost efficient along with the model is a windmill type. A wind sensor is used to sense the wind speed and it is fixed to the shaft with a segmented wheel. When the wind blows, the wind sensor sense the wind and the shaft start to rotate continuously relative with wind speed. In the practical environment, our design shows the improved & efficient result.

“A Comprehensive Survey on Security in Cloud Computing” by Gururaj Ramachandra,
Mohsin Iftikhar and Farrukh Aslam Khan [3]

According to a Forbes’ report published in 2015, cloud-based security spending is expected to increase by 42%. According to another research, the IT security expenditure had increased to 79.1% by 2015, showing an increase of more than 10% each year. International Data Corporation (IDC) in 2011 showed that 74.6% of enterprise customers ranked security as a major challenge. This paper summarizes a number of peer-reviewed articles on security threats in cloud computing and the preventive methods. The objective of our research is to understand the cloud components, security issues, and risks, along with emerging solutions that may potentially mitigate the vulnerabilities in the cloud. It is a commonly accepted fact that since 2008, cloud is a viable hosting platform; however, the perception with respect to security in the cloud is that it needs significant improvements to realize higher rates of adaption in the enterprise scale. As identified by another research, many of the issues confronting the cloud computing need to be resolved urgently. The industry has made significant advances in combatting threats to cloud computing, but there is more to be done to achieve a level of maturity that currently exists with traditional/on-premise hosting.

“Continuous evaluation for application development on cloud computing environments” by Vaibhav Gupta, Rachit Puri and Monir Verma [4]

As the technologies of cloud computing grow and mature, software developers, software companies, and academia are more likely to develop, deploy, and operate their applications or services on cloud computing environments. By leveraging cloud computing environments, the complexity and the cost of physical hardware, operation, and related management tasks can be reduced. The capacities brought by cloud computing environments can also improve particular quality characteristics of the deployed applications or services. However, separate or late consideration and evaluation of both application and cloud computing environment could cause negative impacts on the overall functionalities and qualities. In this paper, the concept and the design of continuous evaluation for application development on cloud computing environments are introduced. Through the continuous evaluation framework, it facilitates the joint consideration and evaluation of application and cloud computing environment as the development is performed and streamlines the development, deployment, and evaluation tasks for better quality and efficiency.

“A survey on location-based application development for Android platform” by Gabriel V. Iana and Cristian Monea[5]

Android is currently the fastest growing mobile platform. One of the fastest growing areas in Android applications is Location Based Service (LBS). LBS provides information services based on the current or a known location and is supported by the Mobile positioning system. Presently, MOSDAC (Meteorological and Oceanographic Satellite Data Archival Centre) disseminates the weather forecast information through web. Android is one of the most widely used mobile OS these days and that is the reason why it is the best practice to develop application on Android platform. The application for disseminating location-based weather forecast is a client-server application on Android platform. It provides weather forecast information as per user's location or location of interest. While developing a client-server application, the communication between client and database server becomes imperative. This paper discusses detailed analysis for choosing appropriate type of web service, data exchange protocols, data exchange format, and Mobile positioning technologies for client-server application. It also highlights issues like memory capacity, security, poor response time, and battery consumption in mobile devices. This paper is about exploring effective options to establish the dissemination service over smart phones with Android OS.

“Distributed storage and processing of real-time data in smart grid dispatching and control systems” by Velu C M, P. Vivekanadan and Kashwan K R [6]

Existing real-time data processing will come to a performance limitation with the rapid development of power grid. This paper proposes a method of distributed storing and processing for real-time data in smart grid dispatching and control systems. The method applies the multi-machine-multi-thread and parallel computing technology to improve data processing efficiency. Real-time data is partitioned into fragments according to region information. By experiments, we prove that distributed storing and processing technology has obvious advantages on performance compared with the original solution, and it can help to relieve the stress of large-scale real-time data processing in smart grid dispatching and control systems.

2.3 MICROCONTROLLER

A microcontroller is a self-contained system with peripherals, memory and a processor that can be used as an embedded system. Microcontrollers are basically employed in devices that need a degree of control to be applied by the user of the device. Programmable microcontrollers contain general purpose input/output pins. The number of these pins varies depending on the microcontroller. They can be configured to an input or an output state by software. When configured to an input state, these pins can be used to read external signals or sensors. When they are configured to the output state, they can drive external devices like LED displays and motors.

	8051	**PIC**	**Arduino(AVR)**	**ARM**
Bus width	8-bit for standard core	8/16/32-bit	8/32-bit	32-bit mostly also available in 64-bit
Communication Protocols	UART, USART,SPI,I2C	PIC, UART, USART, LIN, CAN, Ethernet, SPI, I2S	UART, USART, SPI, I2C, (special purpose AVR support CAN, USB, Ethernet)	UART, USART, LIN, I2C, SPI, CAN, USB, Ethernet, I2S, DSP, SAI (serial audio interface), IrDA
Speed	12 Clock/instruction cycle	4 Clock/instruction cycle	1 clock/ instruction cycle	1 clock/ instruction cycle
Memory	ROM, SRAM, FLASH	SRAM, FLASH	Flash, SRAM, EEPROM	Flash, SDRAM, EEPROM
ISA	CLSC	Some feature of RISC	RISC	RISC
Memory Architecture	Von Neumann architecture	Harvard architecture	Modified	Modified Harvard architecture
Power Consumption	Average	Low	Low	Low
Families	8051 variants	PIC16, PIC17, PIC18,	Tiny, Atmega, Xmega, special purpose AVR	ARMv4,5,6,7 and series

		PIC24, PIC32		
Community	Vast	Very Good	Very Good	Vast
Manufacturer	NXP, Atmel, Silicon Labs, Dallas, Cyprus, Infineon, etc.	Microchip Average	Atmel	Apple, Nvidia, Qualcomm, Samsung Electronics, and TI etc.
Cost (as compared to features provide)	Very Low	Average	Average	Low
Popular Microcontrollers	AT89C51, P89v51, etc.	PIC18fXX8, PIC16f88X, PIC32MXX	Atmega8, 16, 32, Arduino Community	LPC2148, ARM Cortex-M0 to ARM Cortex-M7, etc.

Table 2.1: Comparison of different microcontrollers

2.4 SENSORS USED

2.4.1 Potentiometric wind vane sensor: The sensor used for measurement of wind direction is an IMD-make potentiometric wind vane. The potentiometer in the wind vane is a servo-micro torque potentiometer and has a maximum resistance of 10 kilo-ohms over an end gap of about 4 degrees. The potentiometer is coupled to the wind vane shaft so as to give a resistance output increasing linearly with the increasing of wind direction.

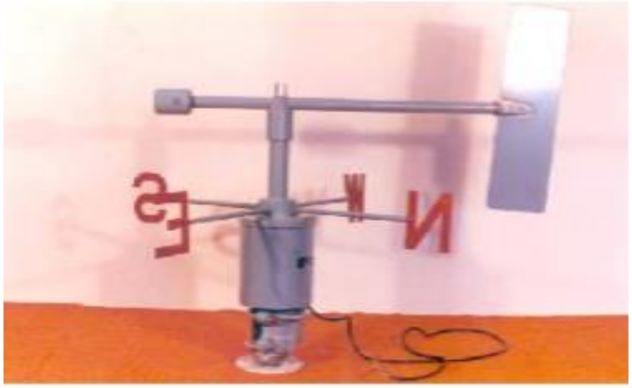


Fig 2.1: Potentiometric wind vane sensor

Thus 0 Kohms corresponding to the north, 2.5 KOhm for east, 5KOhm for south, 7.5KOhm for west and the variation of 0-360 degree corresponds to 0 to 10 kilo ohms.

Calibration procedure for Potentiometric Wind vane:

1. Mark geometric North using magnetic compass.

2. Measure resistance output of pot using multimeter. Move the vane till the resistance is exactly zero ohms.

3. Now arrest vane movement. Rotate North Direction Rod and align to the wind vane position. Fix the north direction rod. (tighten the screws)

4. Fix wind direction sensor without disturbing the position of direction rods. Now rotate whole wind direction sensor over the mast using screw mechanism.

PARAMETERS	**VALUES**
Linearity	**28 ohms per degree**
Max. resistance	**10 kilo-ohms**
Accuracy	**2%**
i/p voltage	**5V or 12V**
Contactless	**yes**

Table no. 2.2 Potentiometric wind vane parameters

2.4.2 Optical anemometer sensor

The sensor used for wind speed measurement is the IMD make three-cup anemometer. The sensor is basically chops IR beam using chopper tooth. System uses an opto coupler for this purpose. No. of chopped electric pulses per min is proportional to the Wind speed. The square pulses are maintained at 0 to 5Volts using Schmitt trigger IC. Later sent to Micro controller which directly gives value of Wind Speed in RS232 format (Data bits:8, Parity: None, Stop Bits:1, Baud rate:4800). Calibration of sensor: Chopper anemometer is automatically calibrated in Wind Tunnel by using standard alcohol manometer. At least 10 readings are taken on wind tunnel by generating different wind speeds, difference in chopper anemometer reading and standard wind speed computed from Manometer is entered in the memory unit of Chopper anemometer thorough same RS232 port, so that anemometer gives wind speed with an accuracy of ±2% of true wind speed up to 100Knots after completion of calibration. A Digital to Analog convertor IC used in the Anemometer to give directly 0 to 4Volts DC for 0 to 100Knots Wind Speed.

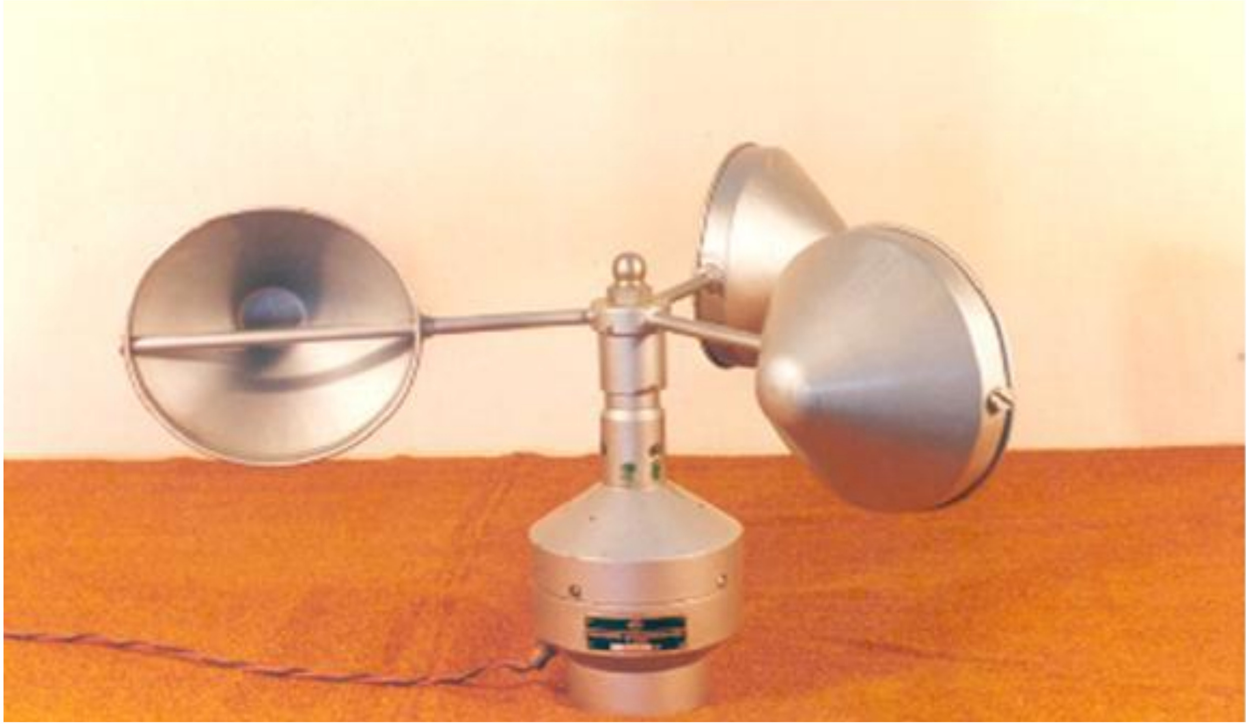


Fig 2.2: Optical anemometer sensor

The requirements from IMD were:

- Minimum of 5 degree of resolution.
- The data should be updated on to the server every 5 minutes.
- The sensor instalment should not affect the current working of the instrument in any means.

PARAMETERS	**VALUES**
Resolution	**5 degrees**
Max. resistance	**10 knots**
Accuracy	**2%**
i/p voltage	**5V**
Contactless	**yes**
Linearity	**0 to 4 volts (0 to 100Knots Wind Speed)**

Table no. 2.3 Cup anemometer parameters

2.4.3 Hygroclip sensor

Hygroclip is a combined sensor for both temperature and relative humidity. The basic sensor for relative humidity is a thin polymer, which is having the property to absorb moisture from the air, and changes its electrical permittivity in proportion to the relative humidity. The polymer is placed between the parallel plate capacitor as a dielectric. The basic sensor for the temperature is PT-100 whose resistance is 100 ohms at 0□C. and the resistance increases linearly with the increase in temperature. Hygroclip requires +12V dc power at field. It has a measuring range of 0-100% for relative humidity and -40 □C to 60 □C for temperature. Its output is 0 1000mV.

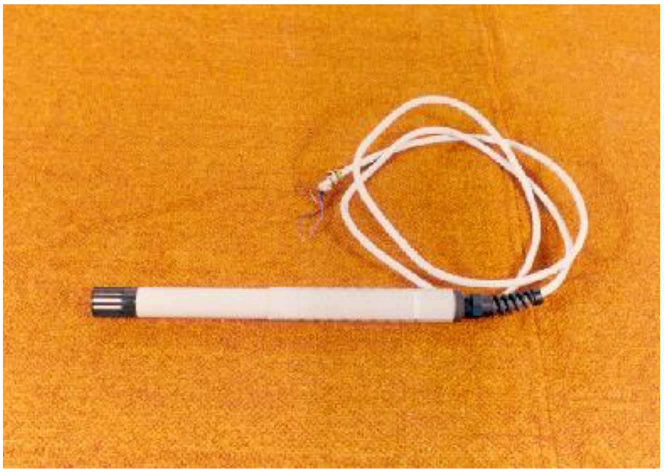


Fig 2.3: Hygroclip Sensor

PARAMETERS	**VALUES**
Linearity	**100 ohms at 0 degrees**
temperature range	**-40 to 60 degrees Celsius**
Relative humidity	**0 to 100%**
Accuracy	**2%**
I/p voltage	**12V**
Contactless	**yes**

Table no. 2.4 Hygroclip sensor parameters

2.5 ARDUINO BOARD

Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. 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.

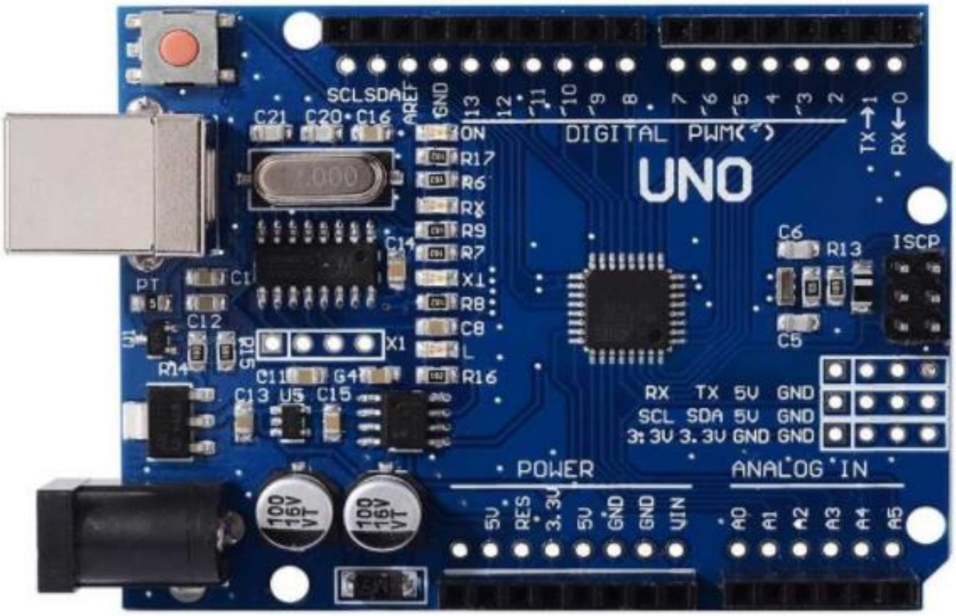


Fig 2.4: Arduino development board

Input and Output

Each of the 54 digital pins on the Arduino 2560 Mega 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 ATmega16U2 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 using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, 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.

TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the Duemilanove or Diecimila.
The Mega2560 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.

2.6 SUMMARY

Various hardware components which we are going to use are described in these chapter. The need for the Softwares is also outlined. Their applications, advantages for the concern are also elaborated. Different papers and journals which we have referred found their way out in this chapter.

3.1 INTRODUCTION

This chapter gives an overview of design and sensors we have used in the project. It also tells us about the software used to complete the tasks. Different types of sensors are used and interfaced with the controller to calculate the weather conditions and that data is accessible using the application.

3.2 DETAILED BLOCK DIAGRAM

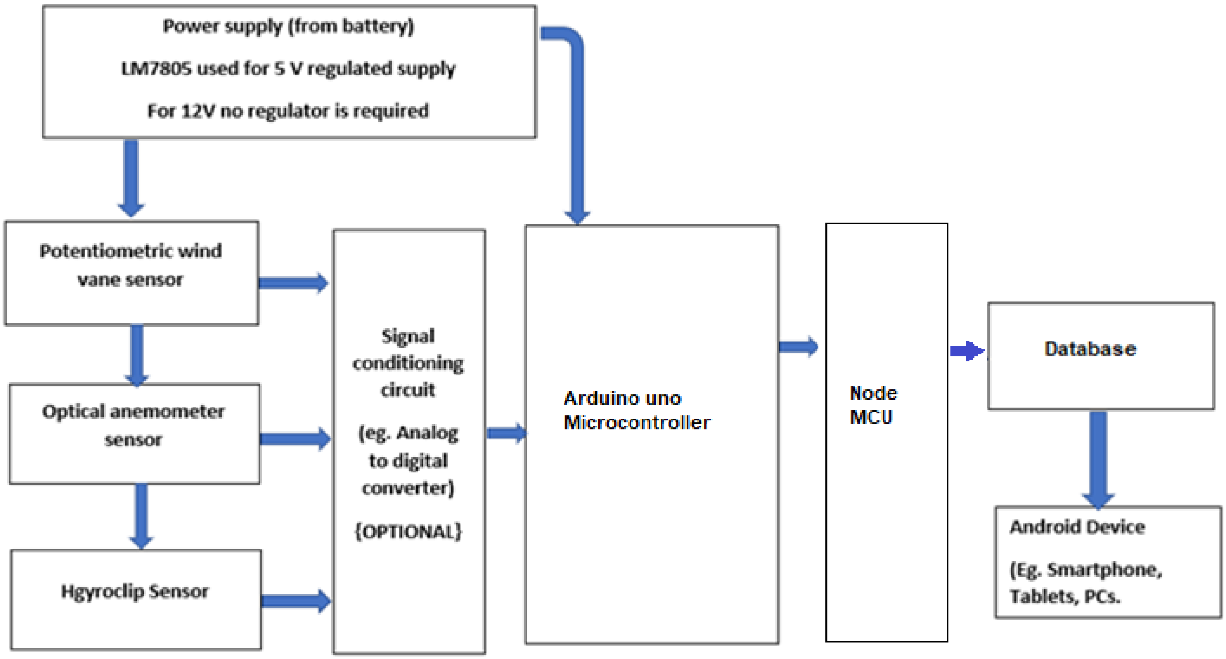
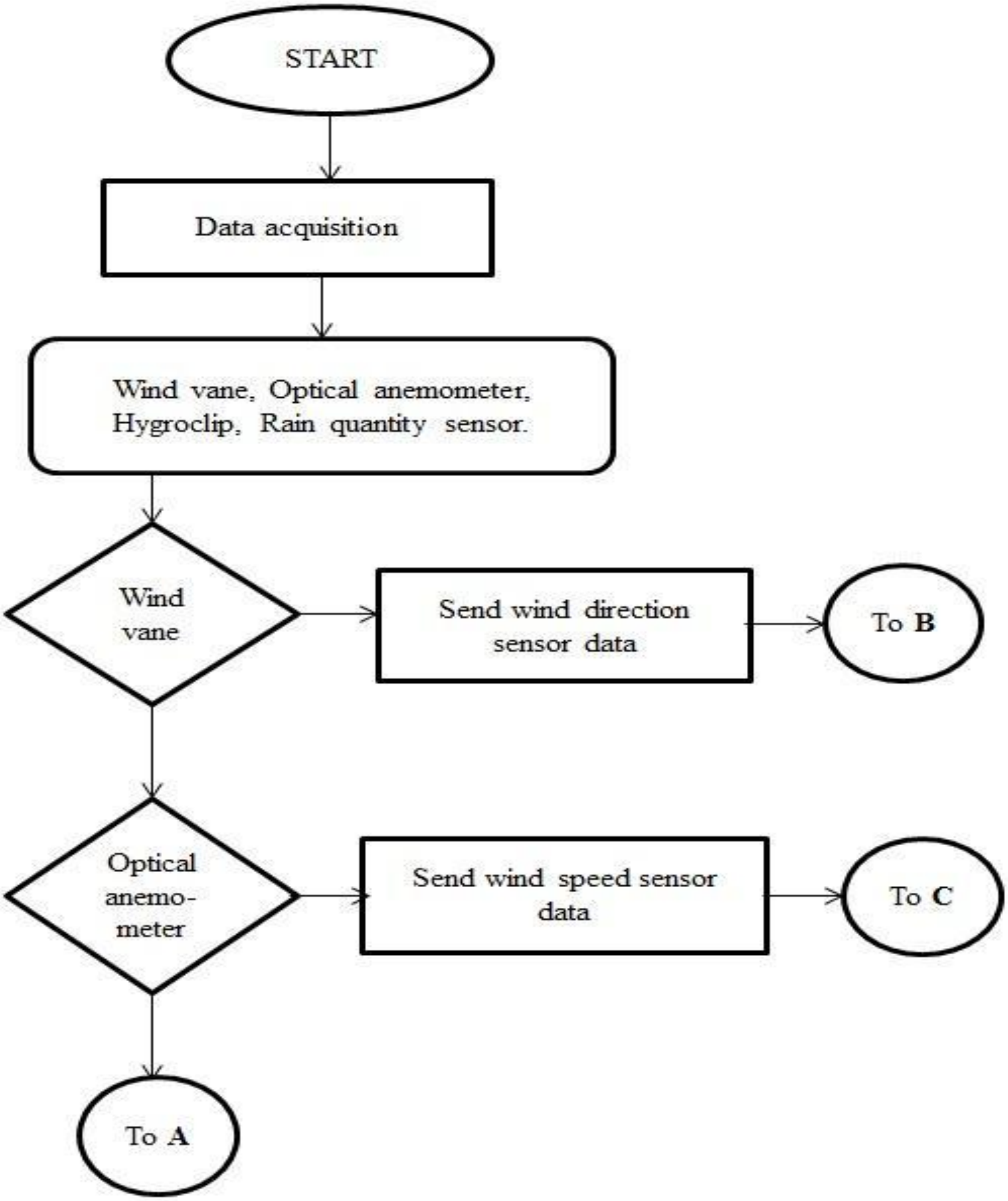

Power supply (from battery)
LM7805 used for 5 V regulated supply
For 12V no regulator is required
Potentiometric wind vane sensor
Optical anemometer sensor
Hgyroclip Sensor
Signal conditioning circuit
(eg. Analog to digital converter)
{OPTIONAL}
Arduino uno Microcontroller
Node MCU
Database
Android Device
(Eg. Smartphone, Tablets, PCs.

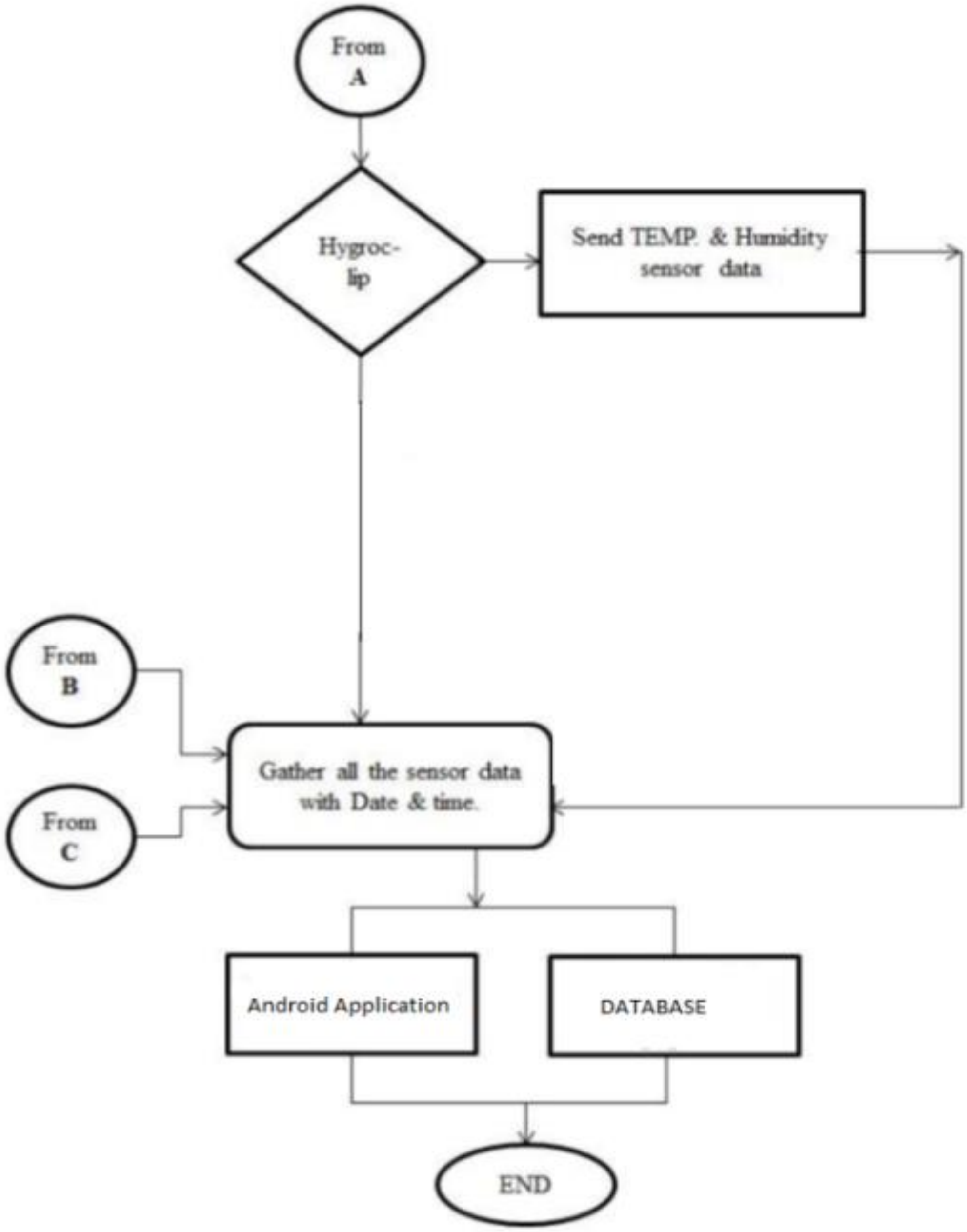
Fig 3.1: Detailed block diagram of the system

Explanation:

The figure shows the detailed block diagram of the system. Power Supply is required by every component so, the regulator is used for sensors to have required supply. The sensors used are: Potentiometric wind vane sensor, Optical anemometer sensor and Hgyroclip sensor. These sensors give the direction of wind, speed of wind, temperature and pressure. These parameters are then fed to the signal processing unit, (like analog to digital converter) but this circuit is optional in our case, as we are using Arduino as the controller. Arduino has an inbuilt conditioning system. The controller will process the data and give it to the cloud storage unit and ultimately the data can be accessed by any Android device (E.g. Smartphone, Tablets, PCs etc.)

3.2.1 Flow Chart:


START
Data acquisition
Wind vane, Optical anemometer, Hygroclip, Rain quantity sensor.
Wind vane
Send wind direction sensor data
To B
Optical anemo-meter
Send wind speed sensor data
To C
To A


From
A
Hygroc-
lip
Send TEMP. & Humidity
sensor data
From
B
From
C
Gather all the sensor data
with Date & time.
Android Application
DATABASE
END

3.3 HARDWARE:

3.3.1 ESP8266 Node MCU:

Introduction:

ESP8266EX is among the most integrated Wi-Fi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area. ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).

Espressif Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or I2C/UART interface). ESP8266EX is among the most integrated Wi-Fi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area. ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).

3.3.2 Internal block diagram:

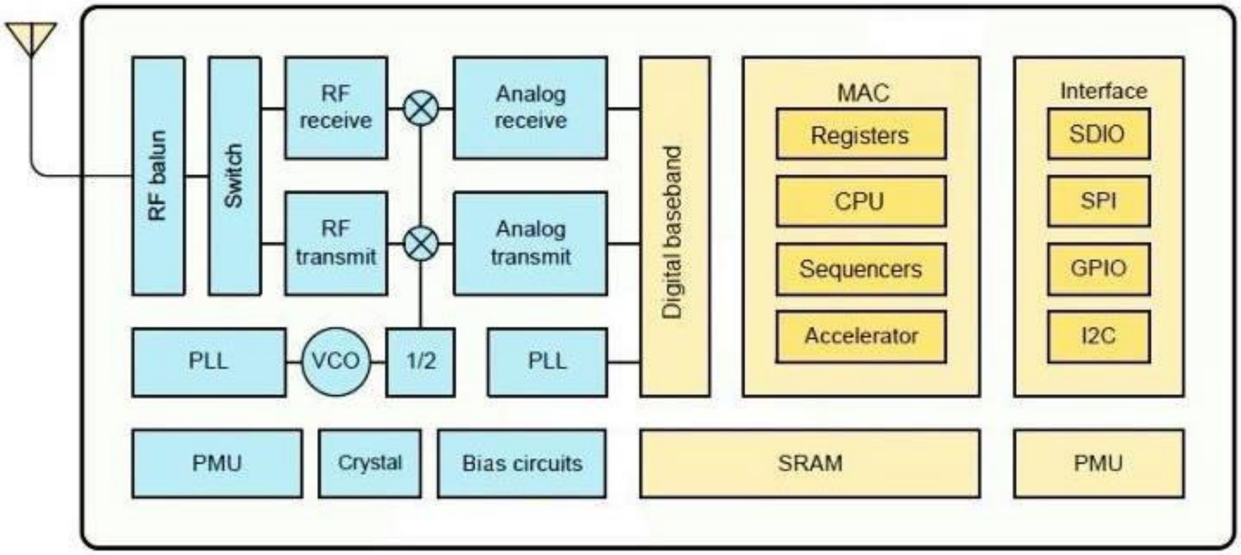


FIG 3.2: Internal block diagram of Node MCU

3.3.3 Features:

802.11 b/g/n

- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes

- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C
- FCC, CE, TELEC, Wi-Fi Alliance, and SRRC certified

3.3.4 Pin Configuration:

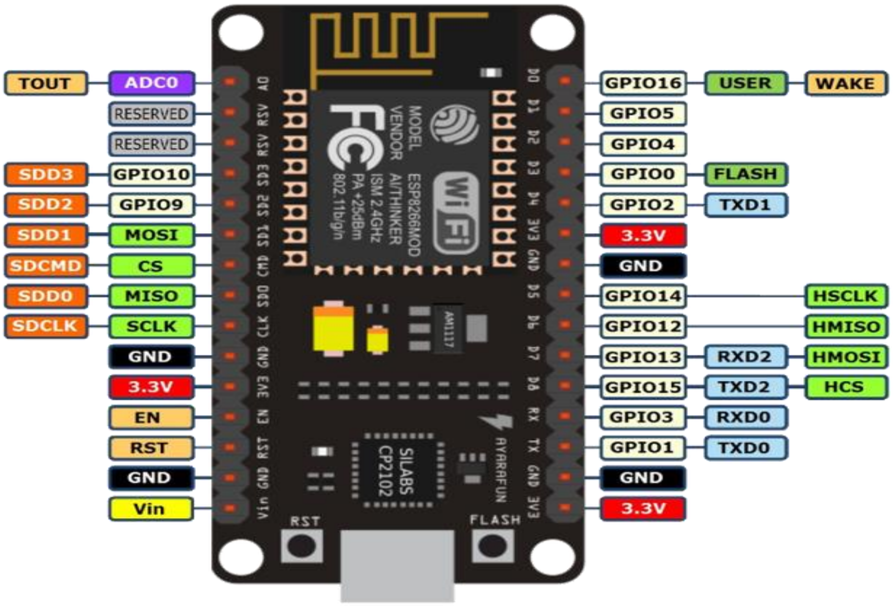


FIG 3.3: ESP8266 (Node MCU)

Name	**Type**	**Function**
VCC	P	Power 3.0 ~ 3.6V
GND	P	Ground
RESET	I	External reset signal (Low voltage level: Active)
ADC(TOUT)	I	ADC Pin Analog Input 0 ~ 1V
CH_PD	I	Chip Enable. High: On, chip works properly; Low: Off, small current
GPIO0(FLASH)	I/O	General purpose IO, If low while reset/power on takes chip into serial programming mode
GPIO1(TX)	I/O	General purpose IO and Serial TXd
GPIO3(RX)	I/O	General purpose IO and Serial RXd
GPIO4	I/O	General purpose IO
GPIO5	I/O	General purpose IO

GPIO12	I/O	General purpose IO
GPIO13	I/O	General purpose IO
GPIO14	I/O	General purpose IO
GPIO15(HSPI_CS)	I/O	General purpose IO, connect this pin to ground through 1KOhm resistor to boot from internal flash.

Table 3.1: Pins of node MCU and its uses

Electrical Characteristics:

Working Voltage: 3.3V

Maximum IO Driving Power IMAX: 12 mA

Maximum IO Voltage Level VMAX: 3.6V

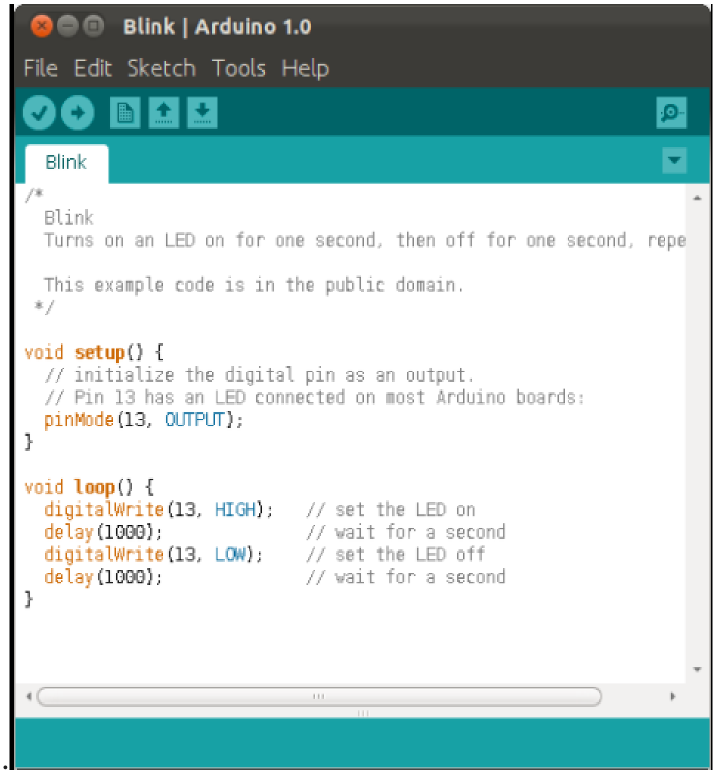
Current Consumption: 100mAmp

3.4 SOFTWARE

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.

3.4.1 Arduino IDE

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.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repe

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Fig 3.4: Screenshot of Arduino IDE, showing a simple program of LED blinking.

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.

Due to the company's open source nature, there exist many free public libraries for developers to use to augment their projects.

3.4.1.1 Libraries in Arduino

Standard Libraries

- **EEPROM** - reading and writing to "permanent" storage
- **Ethernet / Ethernet 2** - for connecting to the internet using the Arduino Ethernet Shield, Arduino Ethernet Shield 2 and Arduino Leonardo ETH
- **Firmata** - for communicating with applications on the computer using a standard serial protocol.
- **GSM** - for connecting to a GSM/GRPS network with the GSM shield.
- **LiquidCrystal** - for controlling liquid crystal displays (LCDs)
- **SD** - for reading and writing SD cards
- **Servo** - for controlling servo motors
- **SPI** - for communicating with devices using the Serial Peripheral Interface (SPI) Bus
- **SoftwareSerial** - for serial communication on any digital pins. Version 1.0 and later of Arduino incorporate Mikal Hart's **NewSoftSerial** library as SoftwareSerial.
- **Stepper** - for controlling stepper motors
- **TFT** - for drawing text, images, and shapes on the Arduino TFT screen
- **Wi-Fi** - for connecting to the internet using the Arduino Wi-Fi shield
- **Wire** - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.

3.4.2 DATABASE:

Firebase:

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data. Store and sync data with our NoSQL cloud database. Data is synced across all clients in real-time and remains available when your app goes offline.

How does it work?

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real time events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great real-time experience that can serve millions of users without compromising on responsiveness.

IMPLEMENTATION PATH

1) Integrate the Firebase Realtime Database SDKs

2) Create Realtime Database References

3) Set Data and Listen for Changes

4) Enable Offline Persistence

5) Secure your data

Key capabilities

1) REALTIME-Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.

2) OFFLINE-Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.

3) ACCESSIBLT FROM THE CLIENT’S DEVICE-The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there’s no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or write.

4) SCALE ACROSS MULTIPLE DATABASES -With Firebase Real time Database on the Blaze pricing plan, you can support your app's data needs at scale by splitting your data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on your project and authenticate users across your database instances. Control access to the data in each database with custom Firebase Realtime Database Rules for each database instance.

3.5 SOFTWARE USED

3.5.1 Eagle

Eagle is a software used for designing PCB, as the components are not available in Proteus. Eagle is abbreviated as Easily Applicable Graphical Layout Editor.

3.5.1.1 EAGLE Circuit Diagram:

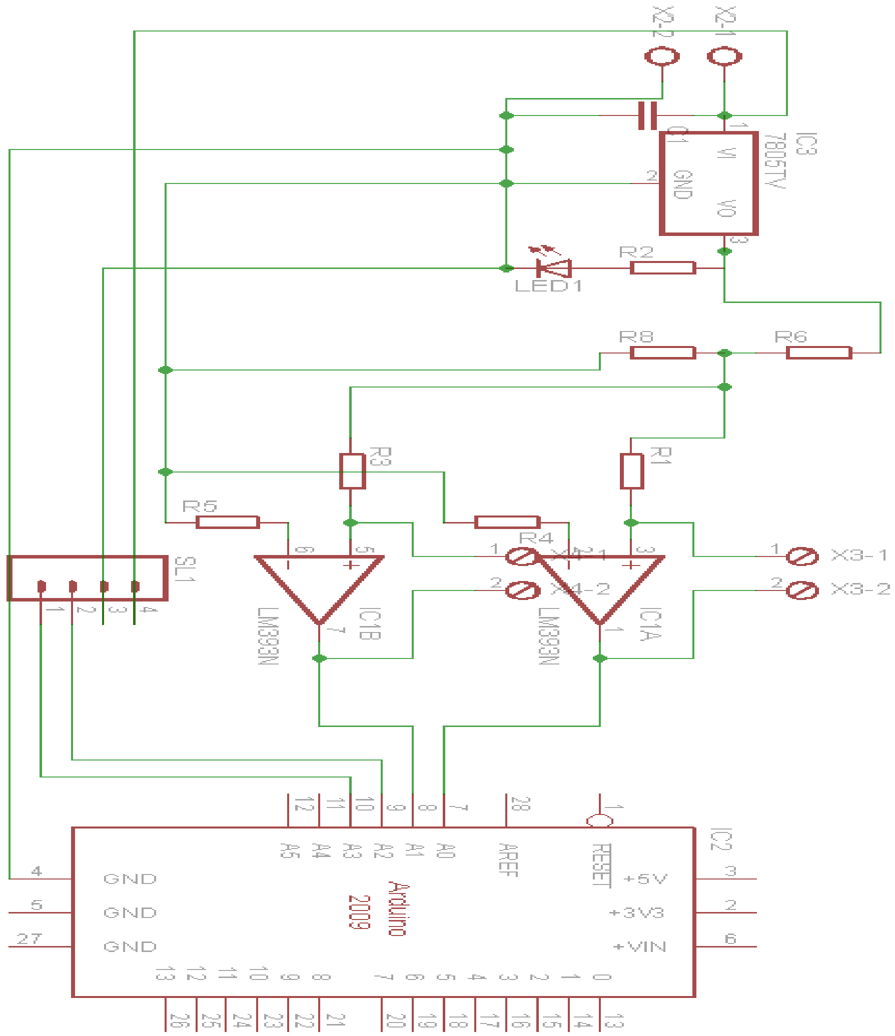


FIG 3.5: Eagle Schematic (circuit diagram)

3.5.2 Arduino

The Arduino Integrated Development Environment - or Arduino Software (IDE) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

3.6 SUMMARY

Thus, this chapter includes the block diagram and the description of the proposed project. The theory of each block is explained.

4.1 INTRODUCTION

Manufacturing includes the production of different parts of assemblies required for the project. In this section, we will study manufacturing of different components of system and corresponding knowledge.

4.2 PCB

- **Base Materials used for PCB**

The base materials used for PCB’s are glass epoxy, epoxy paper, polyester etc. Copper foil used for copper clad is manufactured by the process of electronic deposition.

- **Preparation of Single Sided PCB**

In single sided PCB conductor tracks are present on one side of copper clad board. So crossing of conductors is not allowed. It is mechanically & chemically cleansed. The photo resist is an organic solution which when exposed to light of proper wavelength, changes their solubility in developer but after exposure to light is not soluble.

Laminate coating of photo resist is done by: (i) Spray coating(ii) Dip coating(iii) Roller coating.

The coated copper clad and laminated film negative is kept in intimate contact with each other. The assembly is exposed to UV light and is rinsed in the developer tank. Proper developer has to be used for a particular photo resist and then the PCB is dyed in a tray. The dye reveals the flux to be used for a particular photo resist. Then the PCB is dyed in a tray.

- **FABRICATION**

The required circuit is designed and the layout of the circuit is done on the component side as well as the copper clad side. Spaces are provided for holes to insert the respective components. Etch resistant ink coatings are given on the interconnecting marks.

- **ETCHING**

The copper clad PCB is etched with ferrous chloride solution containing a small amount of Hydro Choric Acid for increasing activeness of Ferric Chloride in etching. Wherever the varnish coating is there the copper remains. Then it is washed with water and Oxalic Acid.

The vast majority of printed circuit boards are made by bonding a layer of copper over the entire substrate, sometimes on both sides, (creating a "blank PCB") then removing unwanted copper after applying a temporary mask (e.g. by etching), leaving only the desired copper traces. A few PCBs are made by adding traces to the bare substrate (or a substrate with a very thin layer of copper) usually by a complex process of multiple electroplating steps.

There are three common "subtractive" methods (methods that remove copper) used for the production of printed circuit boards:

1.Silk screen printing uses etch-resistant inks to protect the copper foil. Subsequent etching removes the unwanted copper. Alternatively, the ink may be conductive, printed on a blank (non-conductive) board. The latter technique is also used in the manufacture of hybrid circuits.

2.Photoengraving uses a photo mask and chemical etching to remove the copper foil from the substrate. The photo mask is usually prepared with a photo plotter from data produced by a technician using CAM, or computer-aided manufacturing software. Laser-printed transparencies are typically employed for photo tools; however, direct laser imaging techniques are being employed to replace photo tools for high-resolution requirements.

3.PCB milling uses a two or three-axis mechanical milling system to mill away the copper foil from the substrate. A PCB milling machine (referred to as a 'PCB Prototype') operates in a similar way to a plotter, receiving commands from the host software that control the position of the milling head in the x, y, and (if relevant) z axis. Data to drive the Prototype is extracted from files generated in PCB design software and stored in HPGL or Gerber file format.

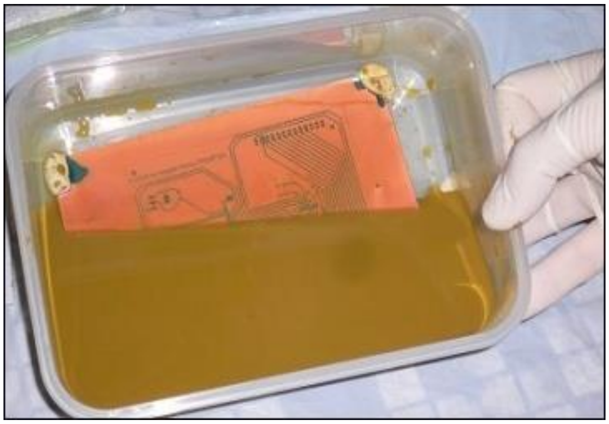


FIG 4.1: PCB Etching

- **DRILLING**

Holes through a PCB are typically drilled with tiny drill bits made of solid tungsten carbide. The drilling is performed by automated drilling machines with placement controlled by a drill tape or drill file. These computer-generated files are also called numerically controlled drill (NCD) files or "Excellon files".

The drill file describes the location and size of each drilled hole. These holes are often filled with annular rings (hollow rivets) to create vias. Vias allow the electrical and thermal connection of conductors on opposite sides of the PCB.

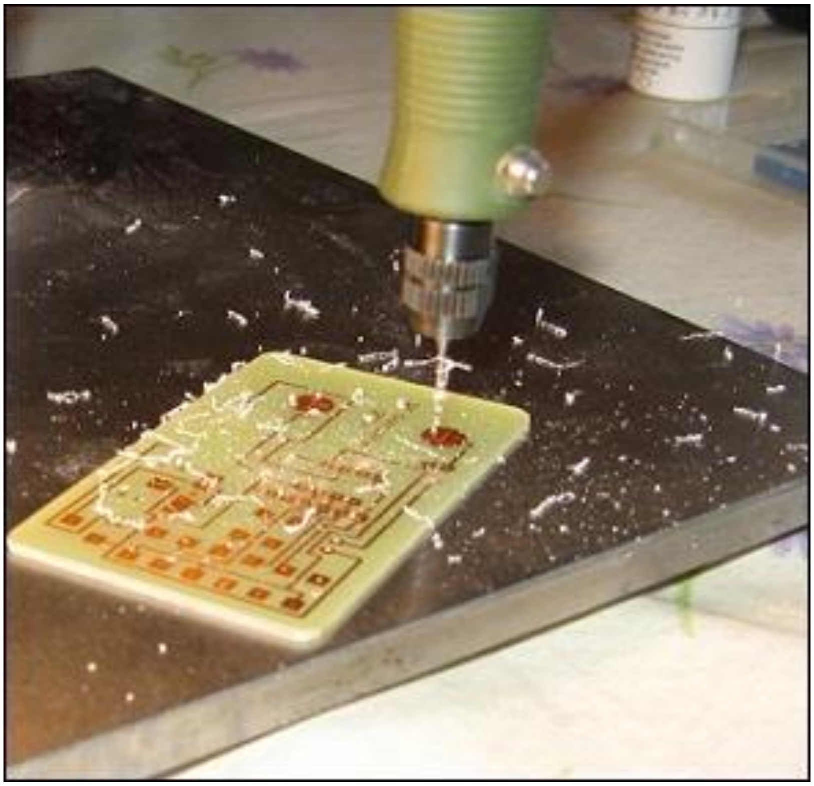


FIG 4.2: PCB drilling

- **Plating and Coating**

PCBs are plated with Solder, Tin, or Gold over Nickel as a resist for etching (removal) away the (unneeded after plating) underlying copper. Matte solder is usually fused to provide a better bonding surface or stripped to bare copper. Treatments, such as benzimidazole thiol, prevent surface oxidation of bare copper. The places to which components will be mounted are typically plated, because untreated bare copper oxidizes quickly, and therefore is not readily solder able.

- **SOLDERING**

Soldering is the process of joining of two metals using an alloy solder consisting of Tin and Lead (Sn-Pb). Tin determines the melting whereas the Lead is used to reduce the cost. After the PCB fabrication is done, the various components are arranged at proper locations on the PCB and then the soldering is done. All liquids consist of particles which attract each other. The surface is always trying to shrink and this is because of surface tension. The principle behind soldering is that when liquid particles are brought in contact with the walls of the solid surface, it may happen that the solid attracts the liquid surface. This property is called adhesive property. Care must be taken that the melting point of solder is below that of the metal so that its surface is melted without melting without the metal.

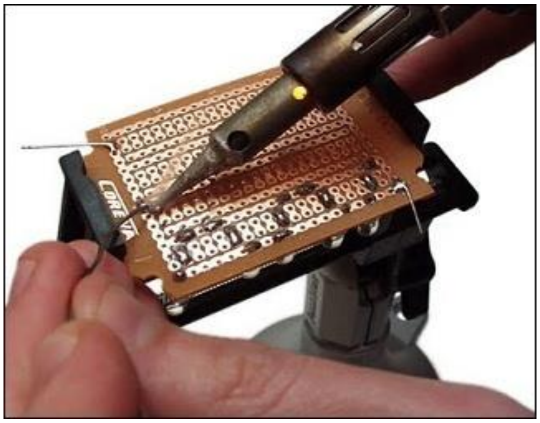


FIG 4.3: Soldering

- **NEED FOR FLUX**

During the soldering process the flux acts as a medium for improving the degree of melting. The basic functions of flux are mentioned below:

1. Removes oxide from the surface.
2. It transfers heat from source to the joining & provide liquid cover including air gap.
3. Removal of residue after the completion of the soldering operation.

4.3 EAGLE PCB Layout:

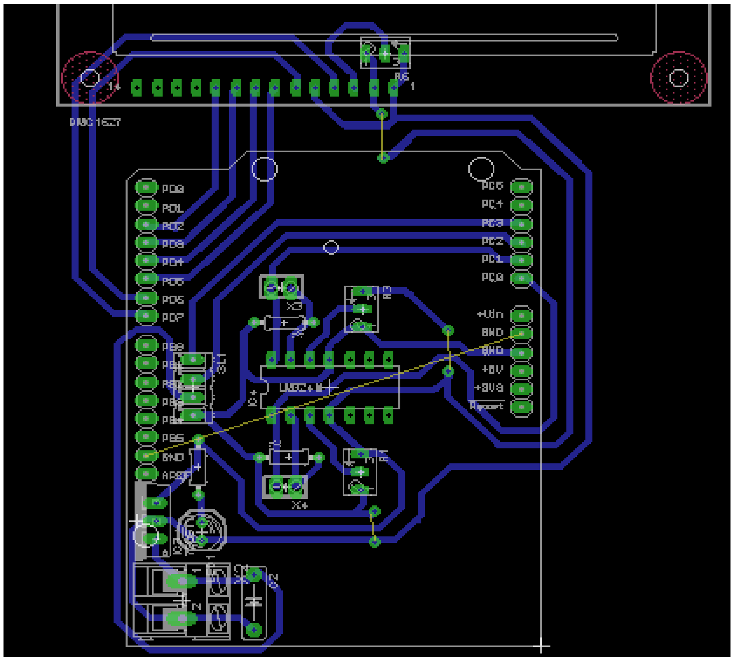


FIG 4.4: Eagle board Layout

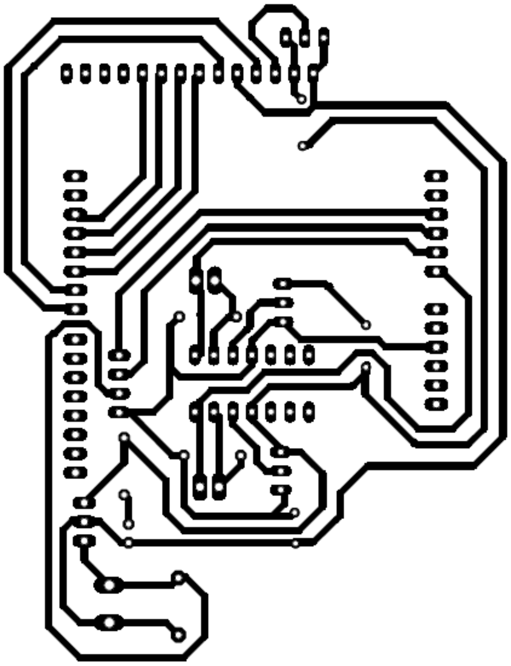


FIG 4.5: Pdf file of the board

4.3.1 PROJECT PCB:

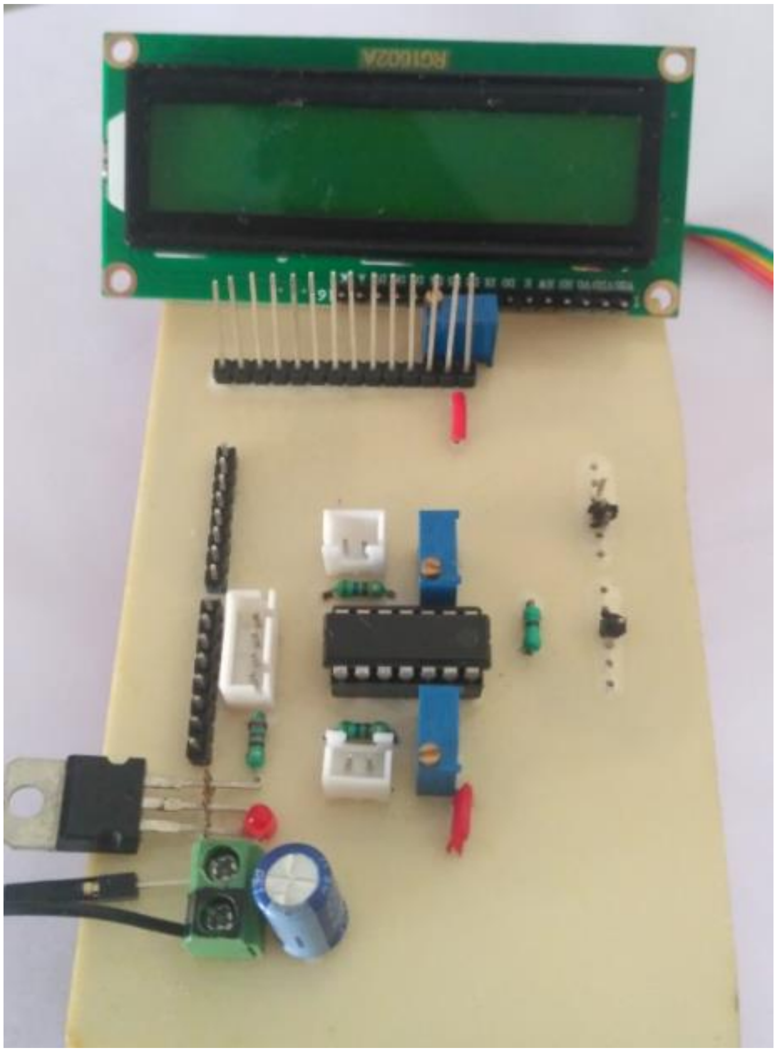


FIG 4.6: Final PCB

5.1 INTRODUCTION

Experimentation includes the change in selection of components depending upon the requirement of the project so that the efficiency, and heat acquired by different components can be controlled

5.2 EXPERIMENTATION

5.3 Experimentation of developing a system of Weather forecasting was carried mainly in 3-stages:

1. Hardware Implementation

2.Android App making & Firebase.

3.Integration of hardware to firebase and APP

1.Hardware Implementation:

Hardware implementation was started with brainstorming and implementing various testing modules on ‘Proteus’ software for simulation, routing of the circuit was done on ‘Autodesk Eagle’, further prototyping the PCB’s by manually transferring the negative ink on copper clad and then selectively etching the unnecessary part of cladding by $FeCl_{3.}$

Prototypes were tested using standard debugging techniques like:

- Continuity check using Digital Multimeter.
- Removing the shorts track by track breakers (desecrating by paper cutters).
- Checking of IC’s (No manufacturing defects).
- Checking for dry solders after soldering of components.

Prototyping was started by interfacing of various forecasting transducers like Temperature sensor, Humidity sensor, and Wind direction sensor.These sensors were implemented by creating a data acquisition system using Arduino IDE & Arduino UNO development board(**ATmega328**).

Data acquired from the transducers was further sent to a GSM module using Serial Peripheral *Interface* (*SPI*) communication type. So that data could be further sent to a WEB SERVER using the secure connection through the internet.

GSM module was eliminated due to two major problem faced:

- Server required needed a permanent domain name purchase which was expensive.
- Only one of either thing of data receiving or creating a secure connection was possible on GSM module

Elimination of problems:

Problems were eliminated by minor hardware changes like GSM module was replaced by NodeMCU(Develop/breakout board of ESP8266 Wi-Fi module with 32-bit controller) and web server by a firebase linked with the App.

Essential Replacement helped in increasing the flexibility of for a vast number sensors (up to 6 in analog transducers & 6-10 digital transducers) interfacing for data acquisition system. Node MCU was used as a client configuration for taking data from UNO and communicating with firebase.

The idea of multiplexing can be used for data acquisition system of fewer transducers, and with good use of interrupt service routine, UNO can be eliminated by providing the output of mux to the analog pin and parallel connection of select inputs to digital pins of NodeMCU.

2.Andriod App making & Firebase :

Android App was made using a web-based freeware ‘MIT(Massachusetts Institute of Technology) App Inventor’Freeware allows an easy access and construction of the highly complex app blocks with a user-friendly UI.

App mainly has a system of accepting the data from firebase through channel then compare it with the variables and continuously update the variable values and putting it in numeric of float format in graphical user interface format of Sensor and data blocks.

Firebase Cloud Messaging Formerly known as Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) is a cross-platform solution for messages and notifications for Android, iOS, and web applications, which currently can be used at no cost.

3.Integration of hardware to firebase and App

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

Data acquired from Arduino Uno to NodeMCU was Properly sorted and was offloaded to the website by defining the web-address of site where the data was compared and stored (updated)to corresponding variables. Which further reflected changes in the Andriod app by sending values through the secure portal connection.

5.3 SIMULATION:

Simulation was carried out on the software Proteus 7.0, all the three sensors potentiometric wind vane, anemometer and Hygroclip sensor were interfaced with the microcontroller and the values were displayed on LCD 16x2

. 5.3.1 Interfacing sensors:

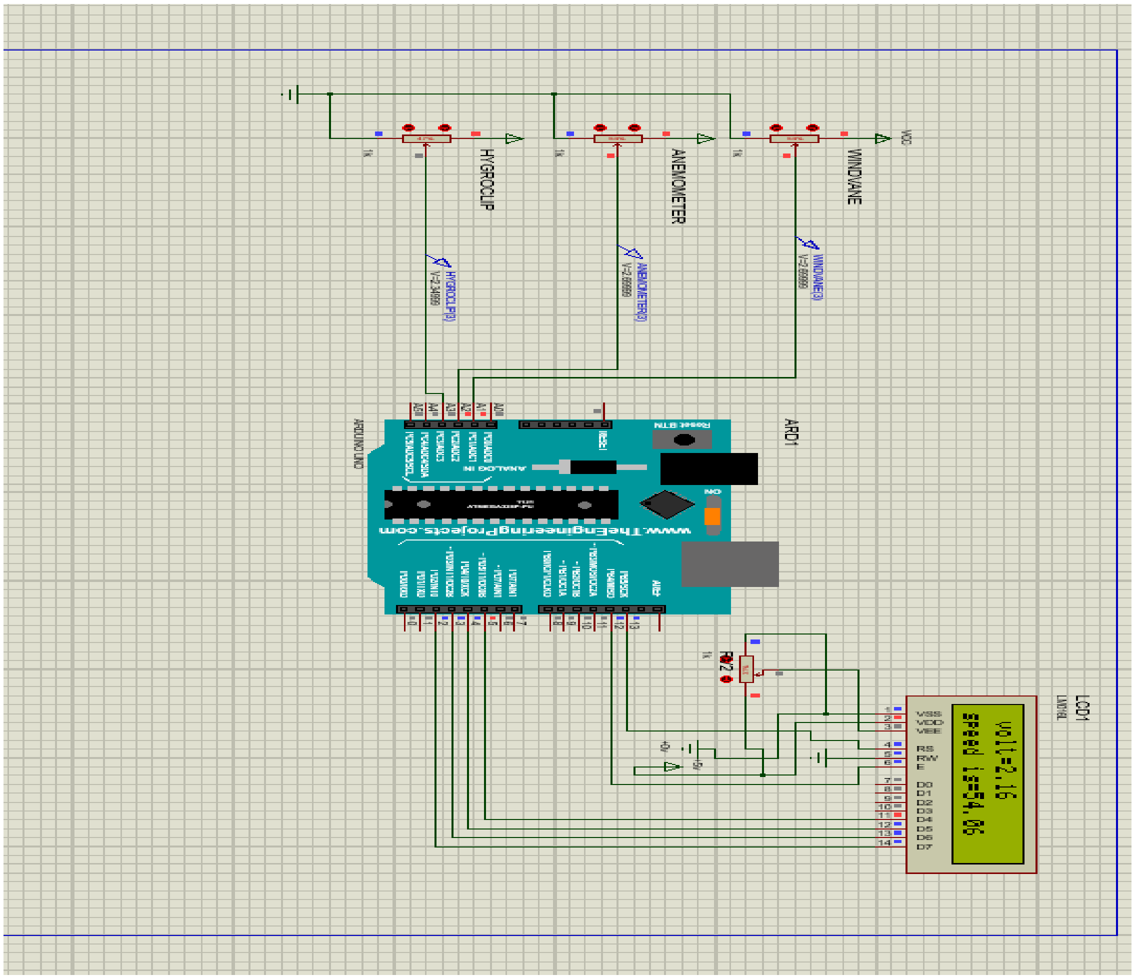
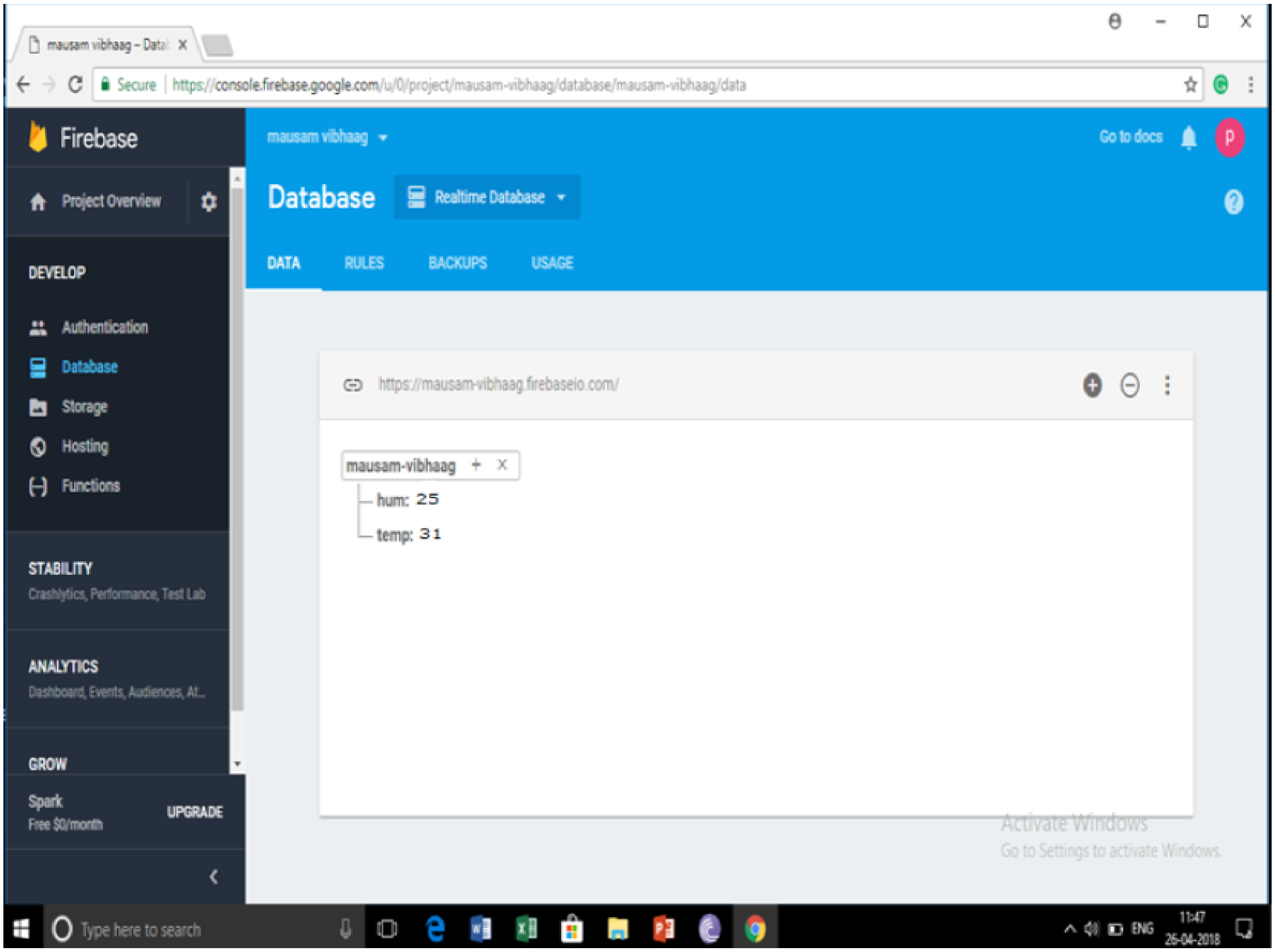


FIG 5.1 Simulation of interfacing of sensors

6.1 INTODUCTION

This chapter includes the results obtained at various stages of the project as well as the final results.

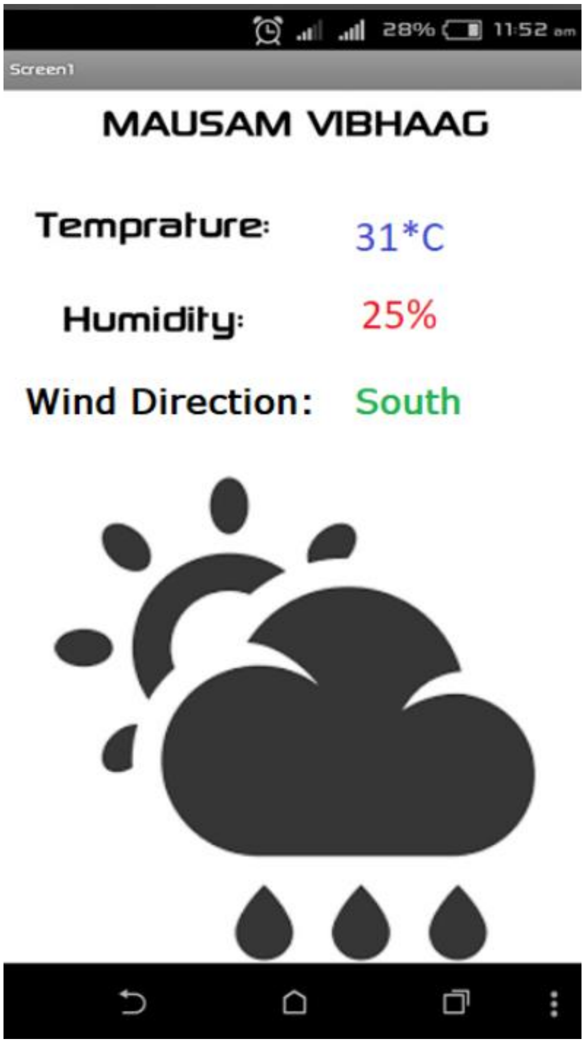
6.2 FIREBASE DATABASE



```
mausam-vibhaag
├ hum: 25
└ temp: 31
```

FIG 6.1: Firebase Database

6.3 APPLICATION



MAUSAM VIBHAAG

Temprature: 31*C

Humidity: 25%

Wind Direction: South

FIG 6.2: Application displaying weather parameters

6.5 CONCLUSION

- The values from all two sensors *viz.* potentiometric wind vane, Hygroclip which were interfaced with microcontroller, were noted down and displayed with the help of LCD 16x2.
- Data is successfully transferred from hardware to firebase database and the database was linked with the android application.
- Data was successfully accessed by the Android application from firebase database.

6.6 ADVANTAGES

- Any time accessibility of weather parameters.
- Wind direction sensing is automated and auto-updating the IMD servers at regular interval of time.
- Self-sustained unit, not entirely dependent on the external electrical connections.
- Reduction in the tiring and tedious work of human forces.

6.7 DISADVANTAGES

- The system becomes liable to the electronic failures for only the digital reading.
- The digital readings are to some extent dependent on the external power supply.

6.8 SUMMARY

Simulation of the interfacing of wind speed and direction sensors is shown and also the flowchart of the working of data processing system is drawn.

REFERENCES

[1] Michael Schmidt, Juan Jose Trujillo and Martin Kuhn, Michael Schmidt "Orientation correction of wind direction measurements by means of staring lidar"*et al* 2016 *Physics Conf. Ser.* **749** 012005

[2] Sree Bash Chandra Debnath, Biki Barua, Rezaul Karim and Muslim Uddin Faisal "Design & Implementation of an Efficient Windmill Anemometer for Wind Speed Measurement Using Microcontroller" International Journal of Engineering Research (978-1-4799-7075-9/15/$31.00©2015 IEEE)

[3] Gururaj Ramachandra, Mohsin Iftikhar and Farrukh Aslam Khan "A Comprehensive Survey on Security in Cloud Computing" Procedia Computer Science 110 (2017) 465–472

[4] Vaibhav Gupta, Rachit Puri and Monir Verma "Continuous evaluation for application development on cloud computing environments" Applied System Innovation (ICASI), 2017 International Conference

[5] Gabriel V. Iana and Cristian Monea "A survey on location-based application development for Android platform" Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in 19-20 March 2015 IEEE.

[6] Velu C M, P. Vivekanadan and Kashwan K R, Advanced Information Technology, "Distributed storage and processing of real-time data in smart grid dispatching and control systems" Electronic and Automation Control Conference (IAEAC), 2017 IEEE 2nd