**M S Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR nagar, MSRIT post, Bangalore-54

A Dissertation Report on

WEATHER STATION

Submitted by

UDITH V 1MS12CS124

*in partial fulfillment for the award of the degree of*

# *Bachelor of Engineering in Computer Science & Engineering*

>



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.S.RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

[www.msrit.edu](http://www.msrit.edu), **May 2015**

# Abstract

# The problem of monitoring the work environment and at the same time increasing connectivity is one of the core problems facing today. There is always a need to monitor the work environment like finding out the temperature and the pressure .This can be particularly helpful to maintaining lab conditions .It might also be required to analyze the data over a period of time which is difficult if done manually.

# This Iot project serves as a solution.

# The project proposes to use a open source microcontroller with the aim of providing a cheap electronic alternative in measuring the temperature by means of an bmp80 pressure sensor and a standard DHT11 temperature and humidity module.

# It then uses a Bluetooth Transceiver Module with TTL Outputs-HC05 to transmit the sensor data to the android device.

# The application running on the android device pairs with the Bluetooth module and receives the sensor data and displays it accordingly.

# This project also uses Thingspeak a cloud based platform and service to create channels and facilitate the monitoring and logging of the temperature over a period of time and also charts them which communicate the changes in the environment to a layman through graphical representation.

#### Contents

***Declaration i***

***Acknowledgements ii***

***Abstract iii***

***List of Figures***

***List of Tables***

1 **INTRODUCTION**

* 1. General Introduction………………. **<Page No>**
  2. Statement of the Problem…………..
  3. Objectives of the project……………
  4. Project deliverables……………
  5. Current Scope………………………
  6. Future Scope……………………….

1. **PROJECT ORGANIZATION**
   1. Software Process Models
   2. Roles and Responsibilities
2. **LITERATURE SURVEY**

3.1….Introduction

3.2…Main Body

3.3 Conclusion of Survey

1. **SOFTWARE REQUIREMENT SPECIFICATIONS**

**4.1** Product Overview

**4.2** External Interface Requirements

**4.2.1** User Interfaces

**4.2.2** Hardware Interfaces

**4.2.3** Software Interfaces

**4.2.4** Communication Interfaces

**4.3** Functional Requirements

**4.3.1** Functional Requirement 1.1

**4.3.2** Functional Requirement 1.n

**4.4** Software System Attributes

**4.4.1** Reliability

**4.4.2** Availability

**4.4.3** Security

**4.4.4** Portability

**4.4.5** Maintainability

**4.4.6** Performance

**4.5** Performance Requirements

**4.6** Database Requirement

**4.7** Design Constraints

**4.8** Other Requirements

1. **DESIGN**
   1. Introduction
   2. Architecture Design
   3. Graphical User Interface
   4. Class Diagram and Classes (represent Inheritance, Aggregation and Association)
   5. Sequence Diagram
   6. Data flow diagram
   7. Metric calculation
2. **IMPLEMENTATION**
   1. Tools Introduction
   2. Technology Introduction
   3. Overall view of the project in terms of implementation
   4. Explanation of Algorithm and how it is been implemented
   5. Information about the implementation of Modules
3. **TESTING**

**7.1** Results and Snapshots

1. **CONCLUSION & SCOPE FOR FUTURE WORK**
2. **REFERENCES**

**INTRODUCTION**

# General Introduction

The purpose of this document is to present a detailed description of the Weather

Station Monitoring system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli such as change in temperature and pressure. This document is intended for both the assessment by the faculty in charge and the developer of the system in order to fairly ascertain the fulfillment of the requirements that constitute the project.

## Statement of the Problem

The problem of monitoring the work environment and at the same time increasing connectivity is one of the core problems facing today. There is always a need to monitor the work environment like finding out the temperature and the pressure .This can be particularly helpful to maintaining lab conditions .It might also be required to analyze the data over a period of time which is difficult if done manually.

This IOT project serves as a solution

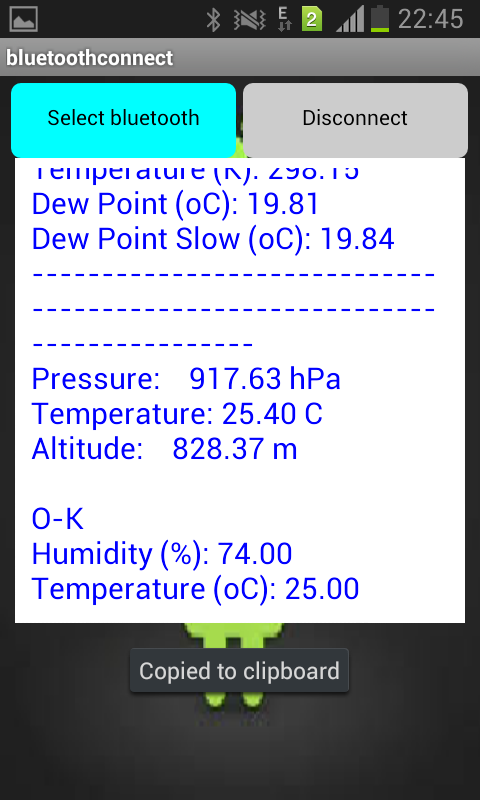
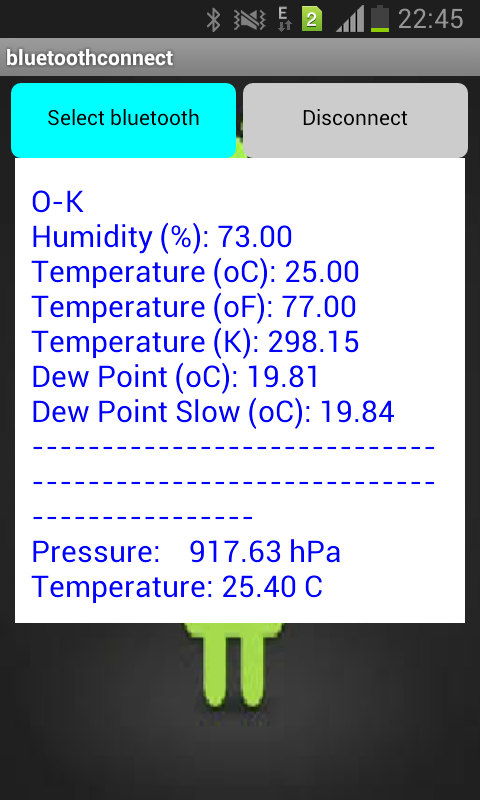
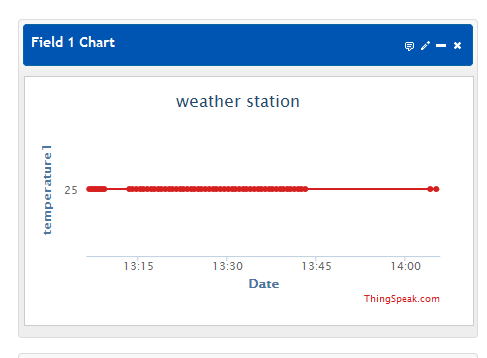
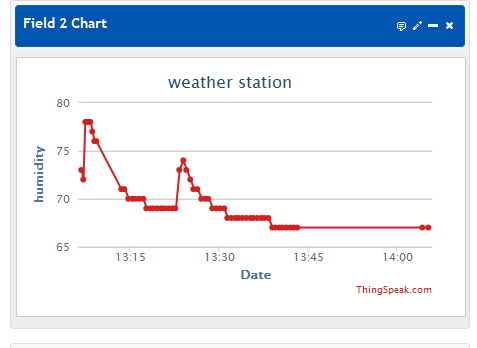
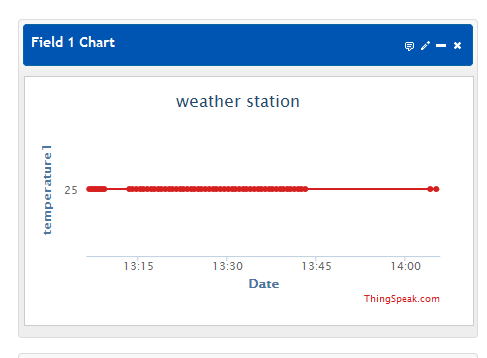
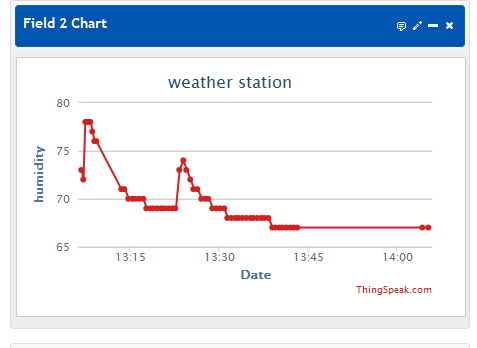
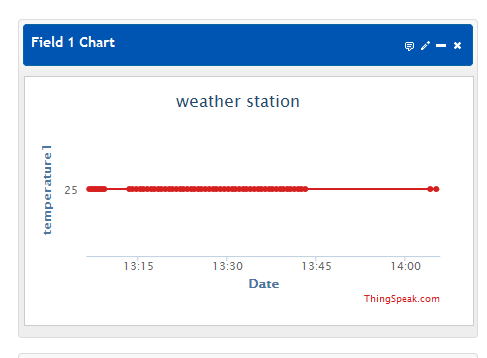
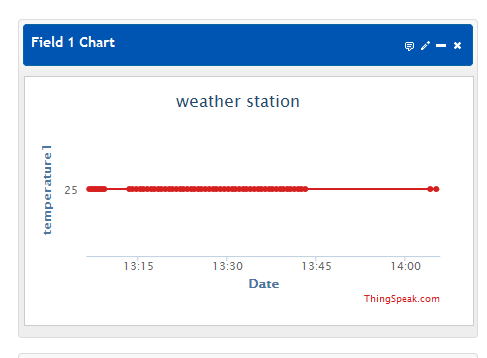
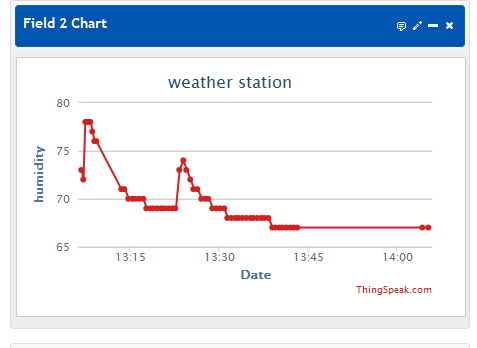
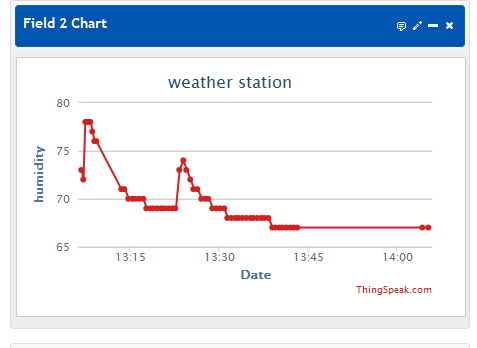
**1.3 Objectives of the Project**

This system is designed to allow an user to manage and communicate with the station through his Smartphone while also being able to track the changes in temperature from any remote device with an internet connection The software will facilitate communication between user, the weather station and Thingspeak, a cloud based service that offers API and charting capabilities via Bluetooth and Ethernet or WiFi. API’s that employ channels as a means of communication are used in logging in of the temperature and humidity through the system to provide instant access to the information; Thingspeak also enables us to perform calculations to find dew point among other things through app making features available. This project also employs the use of Appinventor an MIT based imitative that allows for making simple app’s on a cloud platform through various tools in a purely graphical manner.

## Project deliverables

The project deliverables of this project is an Android app that allows you to access sensor data over Bluetooth and API that displays the graphical representation of the data.

The screen shots of the Android app and the Thingspeak API



## 

**1.5 Current Scope**

This Internet of things Project will be a Weather Station Monitoring system for a chosen environment of the user. This system will be designed to keep track of changes in the weather of the users work environment and use the latest open source hardware and System On Chip technology with low price modules in order to increase connectivity and enable sharing and statistical charting of sensor data .By accessing the clients work environment’s data we are able to increase connectivity and fairly demonstrate the concept of internet of things. The project will meet the user’s needs while remaining easy to understand and use.

More specifically, this system is designed to allow an user to manage and communicate with the station through his Smartphone while also being able to track the changes in temperature from any remote device with an internet connection The software will facilitate communication between user, the weather station and Thingspeak, a cloud based service that offers API and charting capabilities via Bluetooth and Ethernet or Wi-Fi. API’s that employ channels as a means of communication are used in logging in of the temperature and humidity through the system to provide instant access to the information; Thingspeak also enables us to perform calculations to find dew point among other things through app making features available. This project also employs the use of Appinventor an MIT based imitative that allows for making simple app’s on a cloud platform through various tools in a purely graphical manner.

**1.6 Future Scope**

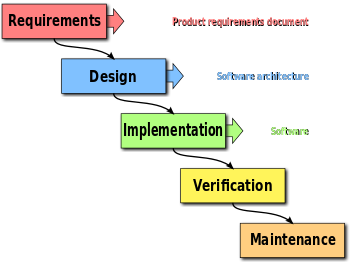
The Future of this project is to be successfully transmit the data over the internet through Wi-Fi. The future can also contain a more pictorial representation of sensor data like gauges.

The device can also be extended to perform operations all of which can be automated and further exploration can be made in the field of device to device communication.

**PROJECT ORGANIZATION**

**2.1 Software Process Models**

Then software process model used for this project is the waterfall model and in some cases extreme programming.



The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

The waterfall development model originates in the manufacturing and construction industries: highly structured physical environments in which after-the-fact changes are prohibitively costly, if not impossible. Since no formal software development methodologies existed at the time, this hardware-oriented model was simply adapted for software development.

In Royce's original waterfall model, the following phases are followed in order:

System and software requirements: captured in a product requirements document

Analysis: resulting in models, schema, and business rules

Design: resulting in the software architecture

Coding: the development, proving, and integration of software

Testing: the systematic discovery and debugging of defects

Operations: the installation, migration, support, and maintenance of complete systems

Thus the waterfall model maintains that one should move to a phase only when its preceding phase is reviewed and verified.

Various modified waterfall models (including Royce's final model), however, can include slight or major variations on this process.[3] These variations included returning to the previous cycle after flaws were found downstream, or returning all the way to the design phase if downstream phases deemed insufficient.

**2.2 Roles and Responsibilities**

As this project is done by a single candidate the all the roles are performed by him.

This involves various roles like hardware integration, coding and integration and software testing.

The majority of the tasks are done with trial and error method with some help from internet tutorials and online courses.

**LITERATURE SURVEY**

**3.1 Introduction**

The literature surveyed for this project mainly comes from three sources. These three sources comprise of mainly sample projects on the topic of sensor SOC interaction, manipulation of data and the advent of the latest technologies concerning the cloud and Internet of things

The sources are as follows:

* 30 Arduino projects for the evil genius
* Arduino and android projects for the evil genius
* Some sample projects from instructables and YouTube

**3.2 Main Body**

ARDUINO INTERFACE BOARDS provide the Evil Genius with a low-cost, easy-to-use technology to create their evil projects. A whole new breed of projects can now be built that can be controlled from a computer. Before long, the computer-controlled, servo-driven laser will be complete and the world will be at the mercy of the Evil Genius! This book will show the Evil Genius how to attach an Arduino board to their computer, to program it, and to connect all manner of electronics to it to create projects, including the computer-controlled, servo-driven laser mentionedearlier, a USB-controlled fan, a light harp, a USB temperature logger, a sound oscilloscope, and many more.

**3.3 Conclusion**

The books and the YouTube videos are helpful with the project conception and implementation .Although there is a lot of trouble shooting that is left undone. This is done by visiting forums on the various components there is a vibrant community that employs the use of the various this open source platform.

**SOFTWARE REQUIREMENT SPECIFICATIONS**

## 

## 4.1 External Interface Requirements

### 4.1.1 User Interfaces

**1. Thingspeak API**

The user can log into the account on thingspeak he can review his channel directly if he makes his channel public for this the user requires any device such as a pc, laptop etc with an internet connection. The user can see the charting of the temperature, humidity and pressure as well as the dew point computed by the thingspeak app.

**2. Android app**

A first-time user of the mobile application should see a page with two buttons one enables him to list the Bluetooth device in the area of which he can pair with the weather station by selecting it and then he can connect with it using the connect button .when he/she connects the application starts to receive and displays it in the textbox. If the user is satisfied with the reading he/she should be able to disconnect from the device through the disconnect button which now occupies the position previously occupied by connect button .

### 4.1.2 Hardware Interfaces

***1. Bluetooth interface***

The Arduino board communicates with the Smartphone through Bluetooth module in this case we use a HC-05 Bluetooth module.

These modules are based on the Cambridge Silicon Radio BC417 2.4 GHz Bluetooth Radio chip. This is a complex chip which uses an external 8 Mbit flash memory

HC-05 is a more capable module that can be set to be either Master or Slave (Available HERE)

HC-06 is a Slave only device. (It looks physically just like the HC-05).(Note: Now HC-06 not cheaper)

These small ( 3 cm long) modules run on 3.3V power with 3.3V signal levels, They have no pins and usually solder to a larger board.

The module has two modes of operation, Command Mode where we can send AT commands to it and Data Mode where it transmits and receives data to another Bluetooth module.

"Breakout" Boards that make these easy to use are available and recommended. These mount the sub-module like that shown on the right on a slightly larger board.

* The Green HC-05 sub-module is soldered on top of the Blue BT Board
* The HC-05 module includes the Radio and Memory chips, 26 MHz crystal, antenna and RF matching network.
* The right section of the BT Board has connection pins for power and signals as well as a 5V to 3.3V Regulator, LED, and level shifting.

HC-05 PinOut :

* KEY: If brought HIGH before power is applied, forces AT Command Setup Mode. LED blinks slowly (2 seconds)
* VCC: +5 Power
* GND: System / Arduino Ground
* TXD: Transmit Serial Data from HC-05 to Arduino Serial Receive. NOTE: 3.3V HIGH level: OK for Arduino
* RXD: Receive Serial Data from Arduino Serial Transmit
* STATE: Tells if connected or not

### For connecting with the internet we use one of the below listed solutions

**1. Internet interface**

# ESP8266 WiFi Module

### This module is very low cost, but very simple. It doesn't support SSL, or communcation over SPI - just UART! It doesn't have 5V to 3V logic level shifting so you'll probably want to pick up a logic level shifter, it also doesn't have a regulator on board and it can use big spikes of 300mA or more current at 3.3V so if using with an Arduino, an external 3V regulator is essential!

Arduino Ethernet Shield

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The most recent revision of the board exposes the 1.0 pinout on rev 3 of the Arduino UNO board.

The Ethernet Shield has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Uno and Mega (using the Ethernet library). The onboard microSD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.

The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.

The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

* IEEE802.3af compliant
* Low output ripple and noise (100mVpp)
* Input voltage range 36V to 57V
* Overload and short-circuit protection
* 9V Output
* High efficiency DC/DC converter: type75% @ 50% load
* 1500V isolation (input to output)

### 4.1.3 Software Interfaces

The software interfaces between Arduino and the modules are mainly through sketch.

The programming for the android app is done in a cloud based app maker. Such as appinventor

### 4.1.4 Communications Interfaces

The communication between the modules and arduino usually take place through an in-built library known as Software serial library.

The Arduino hardware has built-in support for serial communication on pins 0 and 1 (which also goes to the computer via the USB connection). The native serial support happens via a piece of hardware (built into the chip) called a UART. This hardware allows the Atmega chip to receive serial communication even while working on other tasks, as long as there room in the 64 byte serial buffer.

The SoftwareSerial library has been developed to allow serial communication on other digital pins of the Arduino, using software to replicate the functionality (hence the name "SoftwareSerial"). It is possible to have multiple software serial ports with speeds up to 115200 bps. A parameter enables inverted signaling for devices which require that protocol.

The version of SoftwareSerial included in 1.0 and later is based on the NewSoftSerial library by Mikal Hart.

Limitations

The library has the following known limitations:

If using multiple software serial ports, only one can receive data at a time.

.

## 

## 4.2 Functional Requirements

### 4.2.1 Pair with Smartphone

The project must be able to transmit its Bluetooth id . It must be able to pair with the android device once it is initiated by the android app

### 4.2.2 Pair with device

The app must be able to detect and select the weather station and and initiate pairing

### 4.2.3 Read data from Sensors

The weather station must be able to read data from the Dht11 sensor and BMP pressure sensor

### 4.2.4 Transmit data to android app

The arduino must send sensor data in readable form to the android device

### 4.2.5 Display sensor data to user in real time

The android app must display to the user the sensor data as it is made available through Bluetooth connection in real time

### 4.2.6 Connect to the Internet through either Wifi or Ethernet

The device must connect to the internet to transmit data through the use of either the Esp8266 module or Ethernet shield.

### 4.2.7 Communicate to Thingspeak

The device should place the sensor data through the internet connection on the data channels that are to be processed

### 4.2.8 Data channel reading and processing

The Thingsspeak API must read the data and display it in a graphical manner like charts etc.

### 4.2.9 Calculate dew point

The Thing speak app must calculate the dew point and display it.

## 4.3 System Attributes

### 4.3.1 Reliability

The device is highly reliable once constructed as the project because it consistently performs according to its specifications. Since this is a closed project the functionality remains the same .The use of standard variation of C language ensures that it reliable to updates.

### 4.3.2Availability

### The device is always available as it uses portable as well as traditional power supply

### 4.3.3 Security

The device is secure in its connections a s it uses the Wi-Fi that is already under encryption also the pins are so configured that they become a one way line .The Thingspeak channels are also very secure as they use a unique API key for every channel.

### 4.3.4 Portability

The software and hardware can run on any platform .The interface is very standard and is compatible with every port configuration

The App however only runs on an android device

### 4.3.5 Maintainability

The software and the modules are open source or cheaply manufactured and can be replaced easily.

### 4.3.4 Performance

The code is written in a variation of C making it faster. The internet connection also uses TCP/IP connection for fast communication .The communication is serial between the modules and the Soc so the performance is very high.

## 4.4 Design Constraints

.

The project uses the following components:

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control the physical world.

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller it is connected to a computer with a USB cable or powered it with a AC-to-DC adapter or battery.

The technical specifications are:

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

It uses the following sensors:

DHT11 temperature and humidity sensor

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

Technical specifications are:

* Low cost
* 3 to 5V power and I/O
* 2.5mA max current use during conversion (while requesting data)
* Good for 20-80% humidity readings with 5% accuracy
* Good for 0-50°C temperature readings ±2°C accuracy
* No more than 1 Hz sampling rate (once every second)
* Body size 15.5mm x 12mm x 5.5mm
* 4 pins with 0.1" spacing

BMP180 Barometric Pressure/Temperature/Altitude Sensor- 5V ready

This precision sensor from Bosch is the best low-cost sensing solution for measuring barometric pressure and temperature. Because pressure changes with altitude you can also use it as an altimeter! The sensor is soldered onto a PCB with a 3.3V regulator, I2C level shifter and pull-up resistors on the I2C pins.

Technical Details:

* Vin: 3 to 5VDC
* Logic: 3 to 5V compliant
* Pressure sensing range: 300-1100 hPa (9000m to -500m above sea level)
* Up to 0.03hPa / 0.25m resolution
* -40 to +85°C operational range, +-2°C temperature accuracy
* This board/chip uses I2C 7-bit address 0x77.

HC-05 - Bluetooth To Serial Port Module

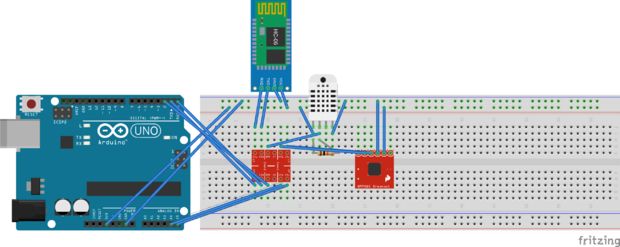
Serial port bluetooth, Drop-in replacement for wired serial connections, transparent usage. You can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project and etc.

And now, we provide HC-05 and HC-06. HC-05 could be setting to Master or Slave by user. HC-06 has be designed Master or Slave when the factory, user couldn't change the role

Feature :

* Bluetooth protocal: Bluetooth Specification v2.0+EDR
* Frequency: 2.4GHz ISM band
* Modulation: GFSK(Gaussian Frequency Shift Keying)
* Emission power: ≤4dBm, Class 2
* Sensitivity: ≤-84dBm at 0.1% BER
* Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
* Security: Authentication and encryption
* Profiles: Bluetooth serial port
* Power supply: +3.3VDC 50mA
* Working temperature: -20 ~ +75 Centigrade
* Dimension: 26.9mm x 13mm x 2.2 mm

The circuit is as follows absent of the Wi-Fi module or Ethernet shield one of which will be added later. This additional module or shield allows us to place our data on the data channels through the internet



ESP8266 WiFi Module

This module is very low cost, but very simple. It doesn't support SSL, or communication over SPI - just UART! It doesn't have 5V to 3V logic level shifting so you'll probably want to pick up a logic level shifter, it also doesn't have a regulator on board and it can use big spikes of 300mA or more current at 3.3V so if using with an Arduino, an external 3V regulator is essential

Technical Details

* Dimensions: 25mm x 15mm x 1mm / 1.0" x 0.6" x 0.04"
* Weight: 1.5

Arduino Ethernet Shield

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The most recent revision of the board exposes the 1.0 pinout on rev 3 of the Arduino UNO board.

The Ethernet Shield has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Uno and Mega (using the Ethernet library). The onboard microSD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.

The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.

The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

* IEEE802.3af compliant
* Low output ripple and noise (100mVpp)
* Input voltage range 36V to 57V
* Overload and short-circuit protection
* 9V Output
* High efficiency DC/DC converter: type75% @ 50% load
* 1500V isolation (input to output)

## 4.5 Logical Database Requirements

No database is used other the one available in the internal working of Thingspeak

**DESIGN**

**5.1 Introduction**

The approach to this project design is one of exploration and experiment. The nature of performing design on open source software and hardware is the issue of troubleshooting. This project presented a unique situation to test one’s integration skills. The is an amalgamation of various projects that are fine tuned to be compatible with the everyday updates that floods this development environment.

**5.2Architectural design**

The project uses the following components:

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control the physical world

Arduino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller it is connected to a computer with a USB cable or powered it with a AC-to-DC adapter or battery.

The technical specifications are:

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

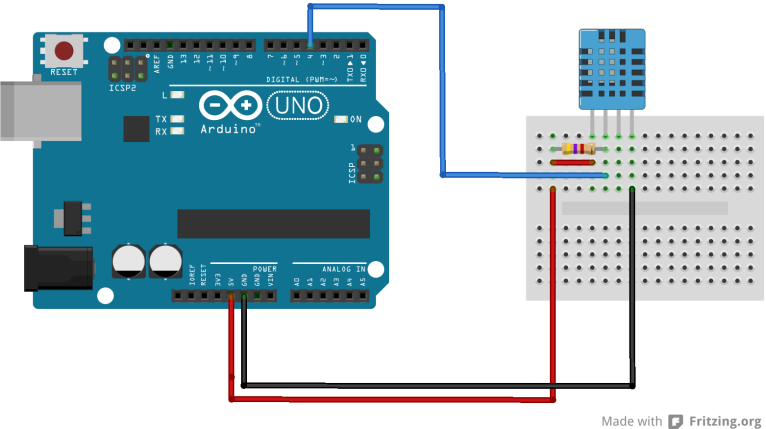
It uses the following sensors:

DHT11 temperature and humidity sensor

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

Technical specifications are:

* Low cost
* 3 to 5V power and I/O
* 2.5mA max current use during conversion (while requesting data)
* Good for 20-80% humidity readings with 5% accuracy
* Good for 0-50°C temperature readings ±2°C accuracy
* No more than 1 Hz sampling rate (once every second)
* Body size 15.5mm x 12mm x 5.5mm
* 4 pins with 0.1" spacing

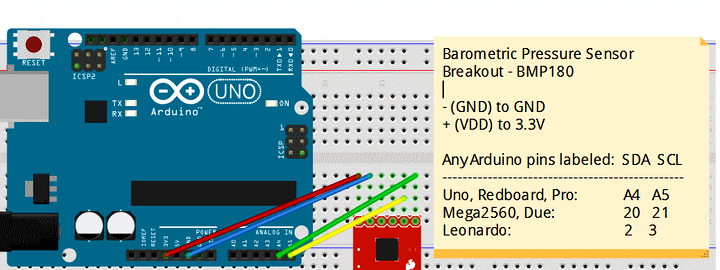


BMP180 Barometric Pressure/Temperature/Altitude Sensor- 5V ready

This precision sensor from Bosch is the best low-cost sensing solution for measuring barometric pressure and temperature. Because pressure changes with altitude you can also use it as an altimeter! The sensor is soldered onto a PCB with a 3.3V regulator, I2C level shifter and pull-up resistors on the I2C pins.

Technical Details:

* Vin: 3 to 5VDC
* Logic: 3 to 5V compliant
* Pressure sensing range: 300-1100 hPa (9000m to -500m above sea level)
* Up to 0.03hPa / 0.25m resolution
* -40 to +85°C operational range, +-2°C temperature accuracy
* This board/chip uses I2C 7-bit address 0x77.



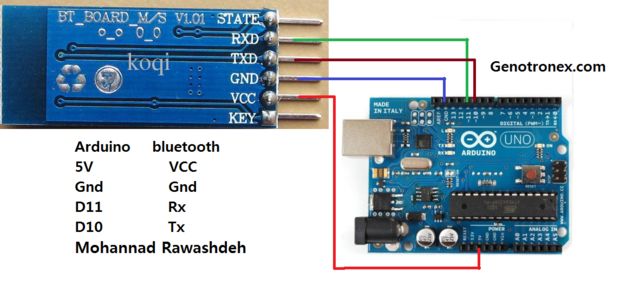
HC-05 - Bluetooth To Serial Port Module

Serial port bluetooth, Drop-in replacement for wired serial connections, transparent usage. You can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project and etc.

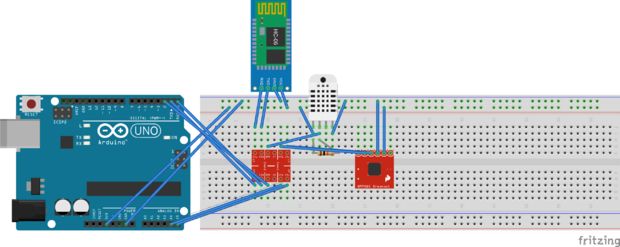
And now, we provide HC-05 and HC-06. HC-05 could be setting to Master or Slave by user. HC-06 has be designed Master or Slave when the factory, user couldn't change the role

Feature :

* Bluetooth protocal: Bluetooth Specification v2.0+EDR
* Frequency: 2.4GHz ISM band
* Modulation: GFSK(Gaussian Frequency Shift Keying)
* Emission power: ≤4dBm, Class 2
* Sensitivity: ≤-84dBm at 0.1% BER
* Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
* Security: Authentication and encryption
* Profiles: Bluetooth serial port
* Power supply: +3.3VDC 50mA
* Working temperature: -20 ~ +75 Centigrade
* Dimension: 26.9mm x 13mm x 2.2 mm



The circuit is as follows absent of the Wi-Fi module or Ethernet shield one of which will be added later. This additional module or shield allows us to place our data on the data channels through the internet

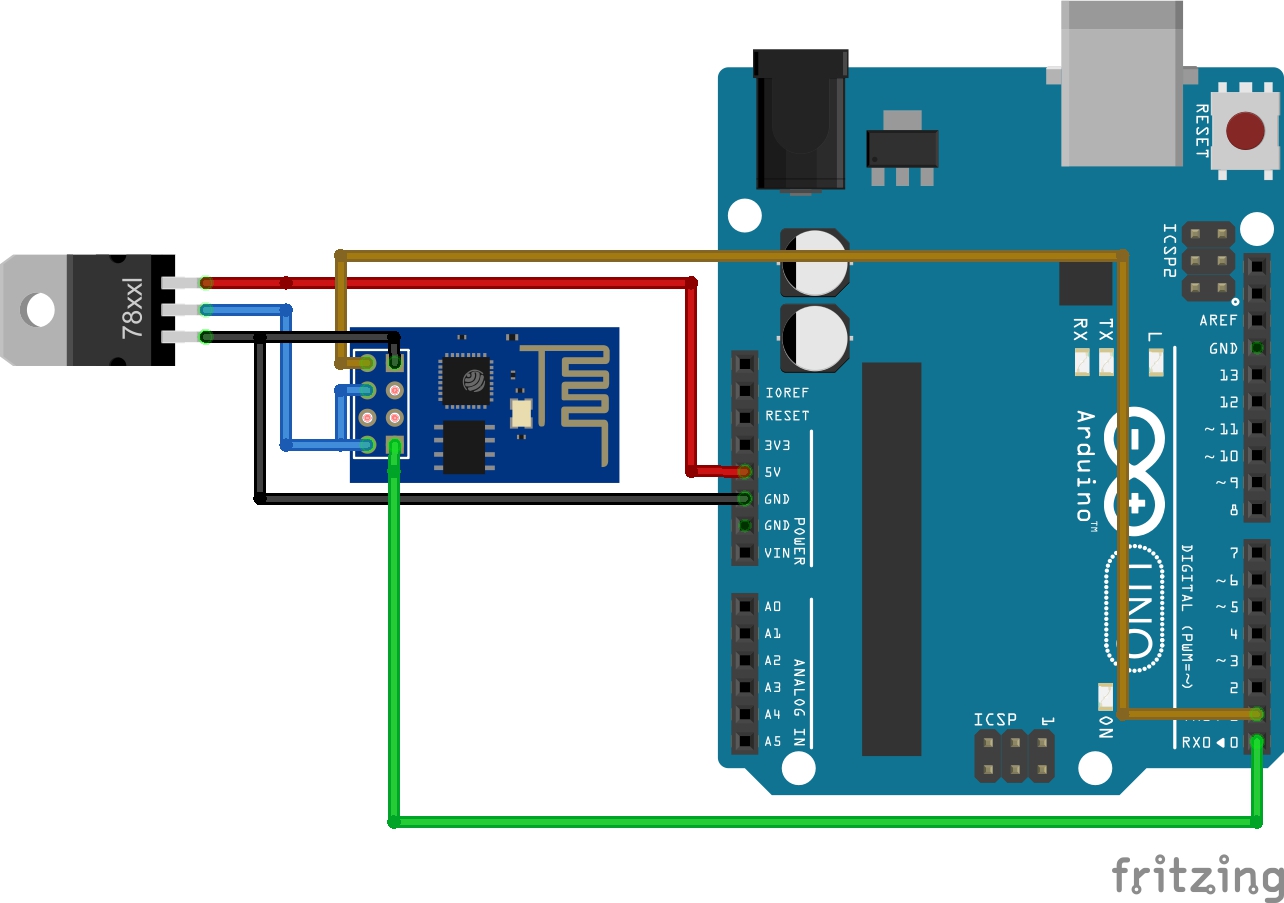


ESP8266 WiFi Module

This module is very low cost, but very simple. It doesn't support SSL, or communication over SPI - just UART! It doesn't have 5V to 3V logic level shifting so you'll probably want to pick up a logic level shifter, it also doesn't have a regulator on board and it can use big spikes of 300mA or more current at 3.3V so if using with an Arduino, an external 3V regulator is essential

Technical Details

* Dimensions: 25mm x 15mm x 1mm / 1.0" x 0.6" x 0.04"
* Weight: 1.5g



Arduino Ethernet Shield

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The most recent revision of the board exposes the 1.0 pinout on rev 3 of the Arduino UNO board.

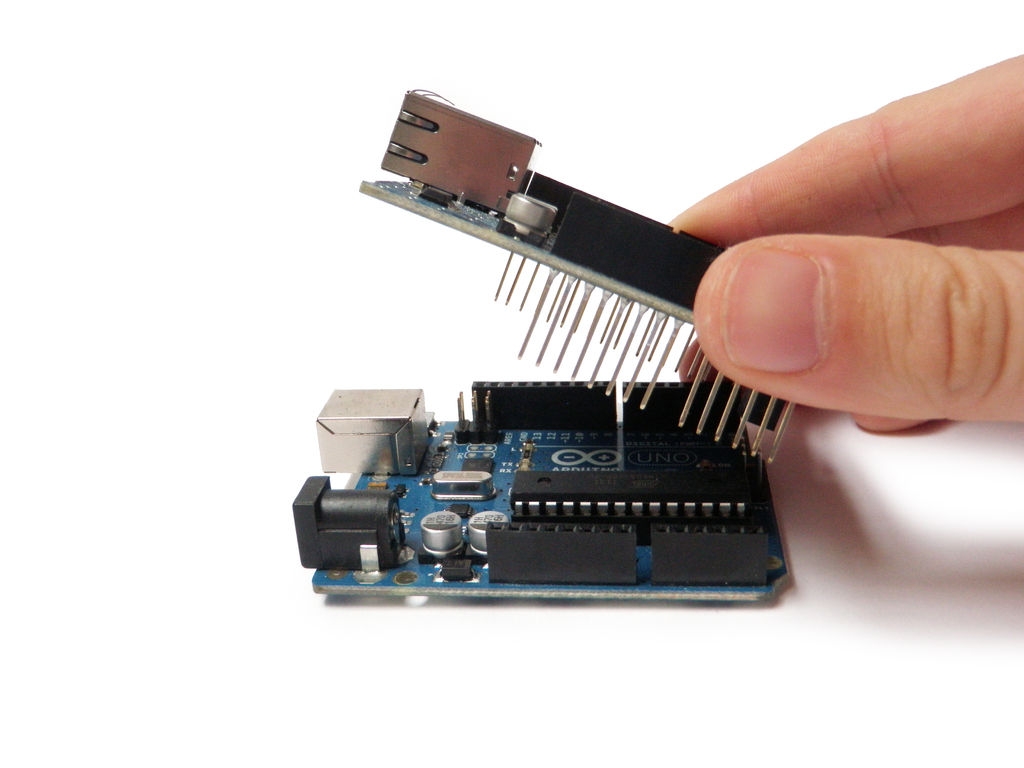
The Ethernet Shield has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Uno and Mega (using the Ethernet library). The onboard microSD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.

The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.

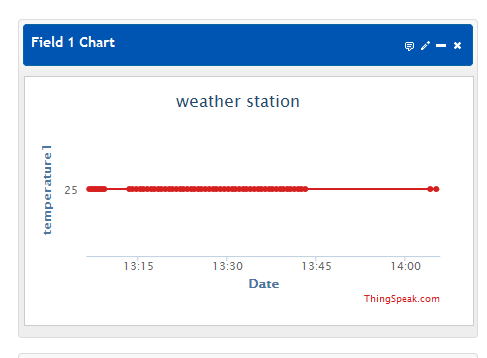
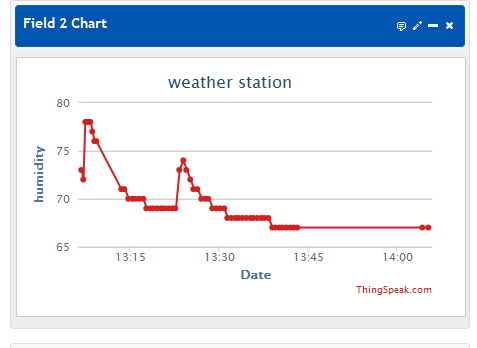
The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

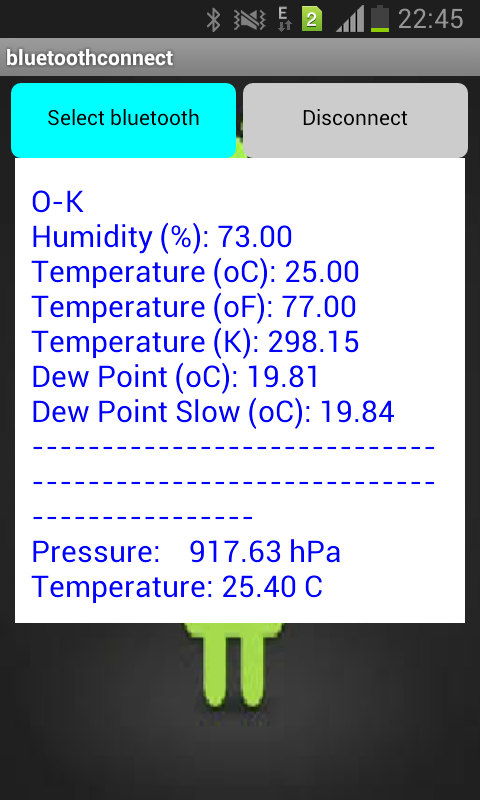
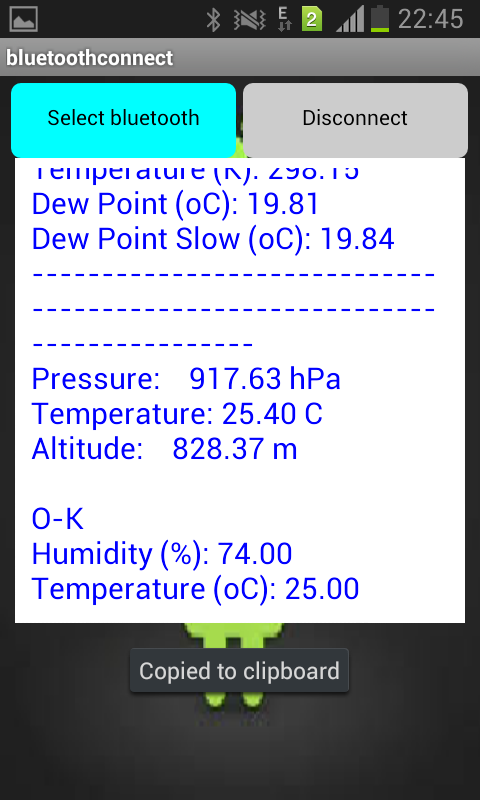
* IEEE802.3af compliant
* Low output ripple and noise (100mVpp)
* Input voltage range 36V to 57V
* Overload and short-circuit protection
* 9V Output
* High efficiency DC/DC converter: type75% @ 50% load
* 1500V isolation (input to output)



**5.3 Graphical User Interface**

The graphical UI consists of a thingspeak API and an Android app





**5.4 data flow diagram**

|  |  |
| --- | --- |
| GET READINGS FROM TEMPERATURE SENSOR  START ARDUINO  START  GET READING FROM BMP180 SENSOR  RECIEVE SENSOR DATA  LAUNCH ANDROID APPLICATION  TRANSMIT DATA THROUGH BLUETOOTH MODULE  DISPLAY TEMPERATURE AND DISTANCE ON APP  TRANSMIT DATA THROUGH WIFI MODULE/ETHERNET SHEILD  LOG IN THINGSPEAK AND OPEN THE CHANNEL  RECIEVE SENSOR DATA    DISPLAY TEMPERATURE AND DISTANCE THROUGH GRAPHS AND CALCULATE DEW POINT  STOP |  |

**8.0 CONCLUSION AND SCOPE FOR FUTURE WORK**

**8.1 CONCLUSION**

India is an agriculture–oriented country. For the quality and Productivity improvement of greenhouse and open field crops, it is necessary to measure and control several interacting physical variables. These tasks can only be accomplished by ‘control systems with built in software’.

Our system highlights about the approach to control the environment in a Greenhouse. The greenhouse controller senses the changes in the temperatures, humidity, the gases present inside the greenhouse through input sensors and processes to take control action. Real time monitoring provides reliable, timely information of crop and soil status, which is important in taking decisions for crop production improvement.

**8.2 SCOPE FOR FUTURE WORK**

Future work will involve finding ways to connect with wireless networks and the more efficient use of API’s

**9.0 REFERENCES**

[1] Wikipedia

[2] Internet of things, Donald Norris.

[3] 30 Arduino projects for evil genius, Simon Monk

[4] Arduino+Android Project for the Evil Projects for the Evil Genius, Simon Monk

**5.5 Sequence diagram**

