WEATHER STATION

**UDITH V 1MS12CS124**

Student, Computer Science Engineering Department, M.S Ramaih Institute of technology, Bangalore, India

**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.

Keywords: .SOC- System on chip, IOT – Internet of things

1. **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 fulfilment of the requirements that constitute the project.

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 analyse the data over a period of time which is difficult if done manually. This IOT project serves as a solution.

**Topics to be included**

Introduction

Design

Implementation

Testing and Comparison

Results

Scope and Future work

Conclusion

Reference

1. **Design**

An 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.

1. Modules And Sensor Specifications

The technical specifications of the Arduino Uno are:

Microcontroller ATmega328P

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limit) 6-20V

Digital I/O Pins 14

PWM Digital I/O Pins 6

Analog Input Pins 6

DC Current per I/O Pin 20 mA

DC Current for 3.3V Pin50 mA

Flash Memory 32 KB (ATmega328P)

of which 0.5 KB used by boot loader

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

• Good for 20-80% humidity readings with 5%

• Good for 0-50°C temperature readings ±2°C

• No more than 1 Hz sampling rate

• 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

• Up to 0.03hPa / 0.25m resolution

• -40 to +85° Coperational range,+-2°C 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

Features:

• Bluetooth protocol: Bluetooth Specificationv2.0+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) / 160kbps,

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

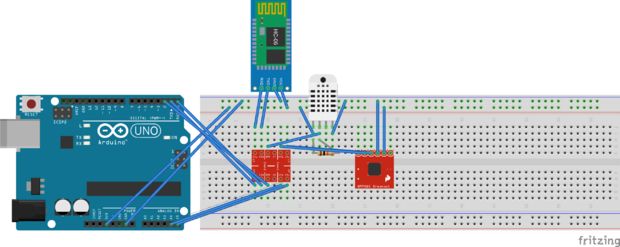


Fig. 1. The circuit diagram

ESP8266 Wi-Fi 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).

1. **implementations**

The project employs c programming with Arduino Uno sensor compatible packages to implement the project .The project employs native interfacing technique to connect components.

1. User interfaces

1. Thing speak 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.

1. 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 Pin Out:

• KEY: If brought HIGH before power is applied, forces AT Command Setup Mode. LED blinks slowly (2 seconds)

• VCC: +5V

• 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

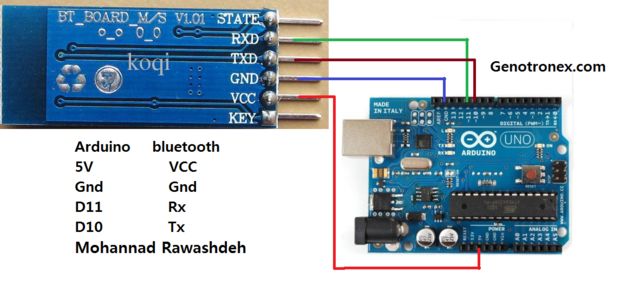


Fig. 1. Arduino interfacing with Bluetooth module

2. Internet interface

ESP8266 Wi-Fi 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!

Arduino Ethernet Shield

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiz net 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)

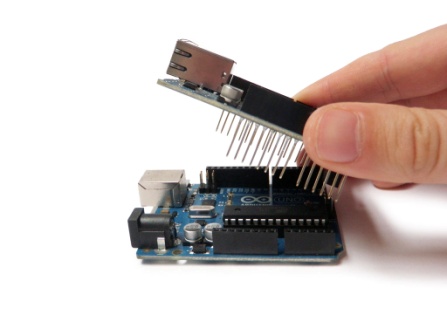


Fig. . Arduino with Ethernet shield

1. 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

Example:

void startEthernet()

{

client. stop();

Serial.println ("Connecting Arduino to network...");

Serial.println ();

delay (1000);

// connect to network and obtain an IP address using DHCP

if (Ethernet.begin(mac) == 0)

{

Serial.println ("DHCP Failed, reset Arduino to try

Again");

Serial.println ();

}

else

{

Serial.println ("Arduino connected to network using

DHCP");

Serial.println ();

}

delay(1000);

}

.

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

1. **Testing And Comparison**

Testing the project involved experimenting with various ways of connecting devices. This involved trying out various modules that would enable accessing the internet.

The primary choice was to access it through the Wi-Fi.

This involved interfacing with a Esp8266 Wi-Fi module.

This involved communicating with the module through AT command instruction set.

Another way of connecting with the internet was through an Ethernet shield that employed the use of GET and POST in the Http protocol to send data to the Thing speak Cloud.

There was also use of graphs as means of visualization

1. **Results**

The project worked better with the Ethernet shield. There were minimal issues with interfacing the different sensors. However there was considerable difficulty with the Esp8266 Wi-Fi module as it involved a lot of troubleshooting. Furthermore, it was found that the latest line of production of the former is Hardware incompatible with the Arduino Uno.

This is because the module runs at Baud rate much higher than that compatible with Software Serial Communication.

The alternative Ethernet shield is used to communicate over the internet.

The Thingspeak API seem to be highly efficient with successful visualization .

The communication with the android app also was successful in rendering the sensor data in text format.

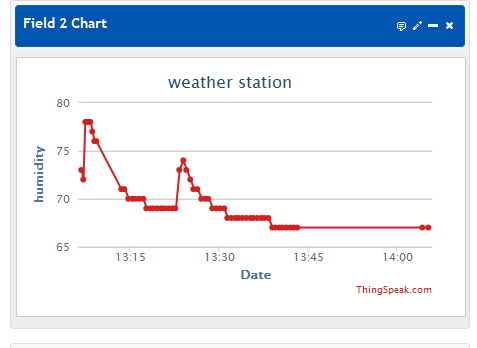
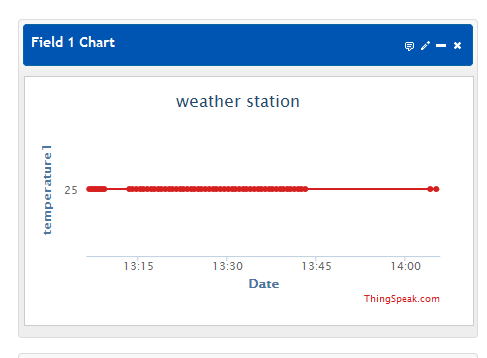
****

Fig. 1.Graphical Visualization on Thingspeak

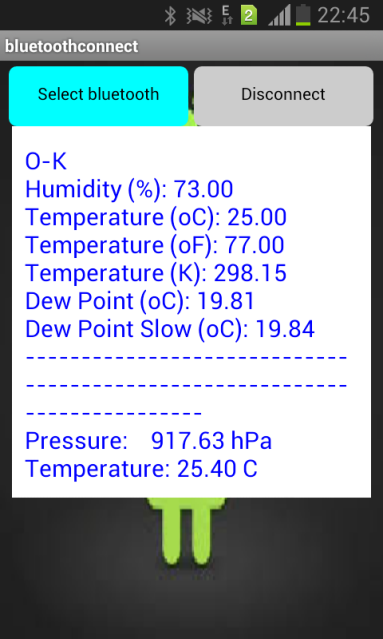
****

Fig. . Android app screenshot

1. **Scope and Future Work**

This Internet of things Project is 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.

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

1. **Conclusion**

This project serves as an efficient way to demonstrate the Internet of Things. How the new electronic world is dealing with the wealth of information available. This project is also successful in demonstrating the transmission of data among different devices and platforms and the conversion of raw data into useful information with increased accessibility.

**ACKNOWLEDGMENT**

I wish to acknowledge Computer Science Department of MSRIT and the faculty for the wealth of advice and guidance. YouTube and its users for providing various tutorials and to Wikipedia for the in-depth knowledge of the technical aspects of this project. Instructables for its various schematics and sample code.

**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