Design & Implementation of Automatic weather station with wireless Sensor network system

[1]Dr K Rameshbabu, [2]Mr Misay.Mangesthu, [3]Mr.Melkapu, Belete,Goshuion
[1]Lecturer, [2]Lecturer&head, [3]students.
[1,2,3]EETM, FTVTI, AA, Ethiopia,

***Abstract:** Climate change is the leading causes of natural disaster in the world. Air temperature, humidity, wind speed and direction, and air pressure readings are by far the best parameters that could promote early detection of climate changes. Recently, wireless network is playing a vital role in automatic weather station and other effort relating to crating smart city [1]. Automatic weather station system is one of the major improvements due to advancements in technology in the field of electronics communication. the number of weather station own country, Ethiopia, is Manned weather stations and ragged to operate because of the labor power, this bring tremendous challenges to the data reader man/weather station/ to provide quick service to the concerned metrology agency. Its proposed Design and Implementing Automatic weather station with Wireless network system. So, this project is specifically developed to test and measure temperature, Humidity, and air pressure by using appropriate sensors.*

First the sensors are installing on mast, connect with Data logger and then Data logger is programmed. Programming was done to read and display real time data from WSN systems. The raw data read from the sensors are converted to engineering units and stored in the Data logger. Real-time output of data can be accomplished with a communication solution UHF/VHF radio transmission. The transmitted Real-time data is received by gateway. Sensors data is logged into SQL server. A wireless sensor network system is configured and integrated into weather station. Again, a LabVIEW program was made to retrieve and manipulate data.

Depend on our design it has three nodes placed on targeted location these three nodes are controlled under host controller. Finally, install test sensor is used to analyze and check validity of AWS and WSN sensors.

Index Terms – climate, datalogger, weather sensor, wireless sensor etc.

I. INTRODUCTION

Ethiopia is one of esAfrica's in weather information richest. Its records established country before Decades. Weather stations developed from the ancient manual stations to automatic station however communicating between stations and the Central weather station still is done manually through telephone and depends on a human worker and that affects sometime in accuracy of readings (time it gets readings). Add on to this methodology poor in flexibility in scheduling readings times and refreshing the readings at any time. This project is about designing weather station that automatically connected to the center through a wireless network. Beside it gives a deep control of reading scheduling. The weather station is designed to measure air temperature, air pressure and humidity.

An automatic weather station (AWS) is an automated version of the traditional weather station, either to save human labor or to enable measurements from remote areas. [1] An AWS will typically consist of a weather-proof enclosure containing the data logger, rechargeable battery, telemetry (optional) and the meteorological sensors with an attached solar panel or wind turbine and mounted upon a mast. The specific configuration may vary due to the purpose of the system. [1] The system may report in near real time via the Argos System and the Global Telecommunications System, [2] or save the data for later recovery. [3]

In the past, automatic weather stations were often placed where electricity and communication lines were available. Nowadays, the solar panel, wind turbine and mobile phone technology have made it possible to have wireless stations that are not connected to the electrical grid or hardline telecommunications network. [4]

A sensor network is a network of microcontroller integrated smart devices, called nodes, which are spatially distributed and sensors. A sensor is a primary component of network essential for monitoring real world physical condition or variables such as temperature, humidity, presence (absence), sound, intensity, vibration, pressure and motion. The important design and implementation requirements of a typical sensor network are energy efficiency, memory capacity, computational speed and bandwidth. The smart device has a microcontroller, a radio transmitter, and an energy source. Sometimes a central computer is integrated into the network in order to manage the entire networked system. A sensor network essentially performs three basic functions: sensing, communicating and computation by using the three fundamental components: hardware, software and algorithms, respectively. A wireless sensor network (WSN) consists of nodes, sensors and a gateway. The smart device or node has some degree of intelligence for signal processing and management of network data. (Mahalik, 2007)

This device gathers data and transmits it via wireless links to user through a gateway. The nodes can be stationary or moving. They can be aware of their locations or not. They can be homogeneous or not. Monitoring and communication are performed cooperatively by the nodes. The main features of WSNs are: scalability with respect to the number of nodes in the network, self-organization, self-healing, energy efficiency, of connectivity among nodes, low complexity, low cost, and small size of nodes. (Buratti, Martalò, Verdone, & Ferrari, 2011) Automatic weather station AWS is an application with sensors, data logger and equipment for observing weather conditions to provide information to study weather and weather forecasting. The measurements taken include temperature, humidity, and pressure.

Problem Statement of the Project: Currently, there are a number of weather stations systems available for the metrology which can be used only when manually near to controller. These systems are manned everywhere and difficult to address update information to the concerned body on time. The variable measuring device is monitor and data transferred to the PC is in manually [2]. Such systems have problems in terms of flexibility, portability, distance and time. Within hour regular monitoring of measuring device is not possible because of tedious work, lost labor time and energy. These systems cannot cover the entire large geographical area /weather climate/ data and cannot transmit continuously and wirelessly [1]. So, to overcome these limitations of the systems we have proposed a new system called "design and implementing AWS with wireless network System". This system is able to transmit the weather climate parameters of provided area continuously and over long distance wirelessly. Therefore, using the proposed system, the man/controller in metrology agency/ can attend weather climate /nodes/ being anywhere and they can provide quick service and save a lot of time and energy

II.OBJECTIVES

II.1 General objective: The general objective of this project is to Design and Implementation of automatic weather station with wireless sensor network system for Ethiopia (in case of metrology Agency).

II.2 Specific objectives of the project: The specific objectives of our project are:

- To develop remote weather climate temperature reading system
- To develop remote weather climate humidity reading system
- To develop remote weather climate air pressure reading system
- To display the measurements in the station. & To indicate of the system status in the station.

II.3 Significance of the project: Quick service/automated real information/ or human energy and time in manned weather station are among well-known issues both the metrology agency and the people. This project provides the Ethiopian metrology agency and other concerned body to maximize the quality, and breadth of service by controlling every time and everywhere. As the geographical area increases the sensor node for services increases. Thus, the ability to maintain the quality and update information of metrology while effectively managing financial and human resources is achieved by this project. The use of modern communication technology in this context is also the sole decisive factor that makes such weather station and sensor node wireless communication system successful.

II.4 Scope &Limitations of the Project: As broad and remote geographical area gradually becomes a trend in wireless communication networking, the integration of wireless sensors with global connectivity seems to be the next step in providing automatic weather station services [9, 11]. The end-product of the project is process and transmit up to three channels of weather climate measured signals via 2.4 GHz, IEEE 802.15.4 radio based on ZigBee technology.

It allows the transmission of data for any or all of the three most common parameters used in metrology, which are the air temperature, air pressure and humidity. To achieve this aim, it requires an embedded RF interface between the data logger and a gateway so that the signals can be transmitted transparently from a sensor node to a weather station/host controller/ via gateway. The data logger unit has directly-wired inputs from sensors on the weather climate and a wireless connection to speed a gateway. Our project is limited to move dynamically a sensor node one location to the other location.

Project Organization: This project is entitled as “design and implementing automatic weather station with wireless network” and it consists of the following five chapters with detail description: Chapter one introduces the project. Chapter two describes the Sensor network overview while chapter three describes system model and development. Chapters four and five explain the results and discussion, and conclusion and recommendation of the project respectively.

III SYSTEM HARDWARE AND SOFTWARE DEVELOPMENT

System Hardware Development: In this section, each sensor is interfaced with the microcontroller (Arduino Uno), SD module and LCD separately. The sensor is placed at weather outside the door bedside to measure the data. Then, the data measured is analyzed by microcontroller. Each sensor has three pins with the input pin connected to +5V, ground pin connected to ground and output pin connected to the microcontroller (Arduino Uno) input pin. The data pin of the transmitter modulo is connected with the Arduino Uno Tx pins and The data pin of the receiver modulo is connected to the virtual display instead of gateway and computer . The LCD is powered up with +5V. Data bus lines of LCD are connected to Arduino Uno output pins.
The following figures show the hardware interfacing and connection set-up.

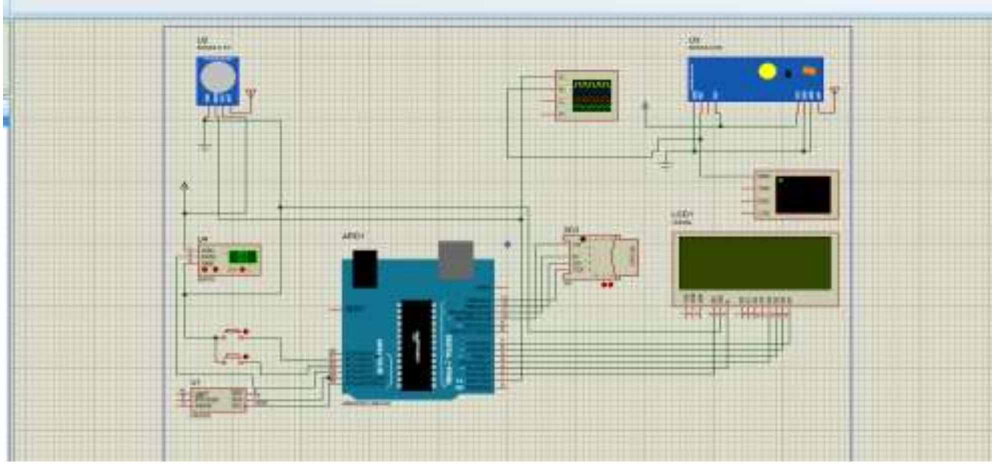


Fig 3.1 DHT22 sensor interfaced with Arduino Uno and SD card

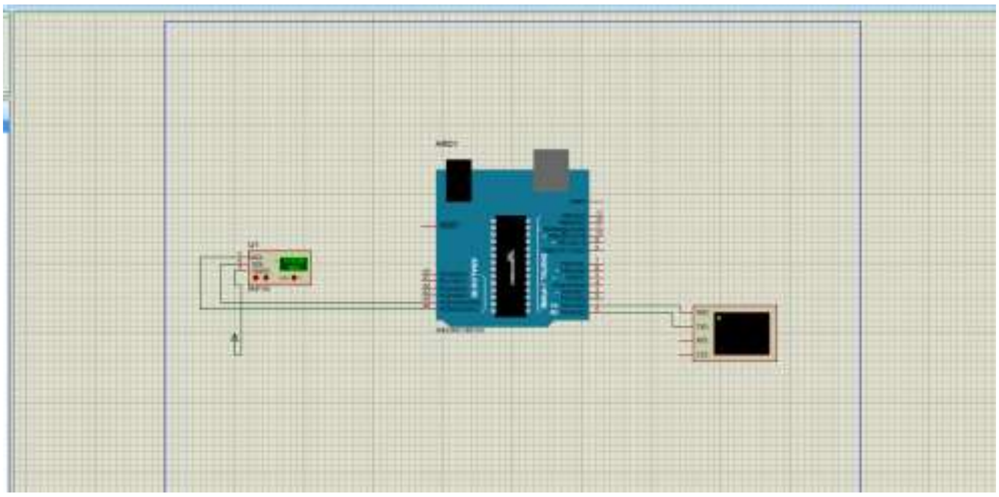


Fig 3.2 BMP180 sensor interfaced with Arduino Uno

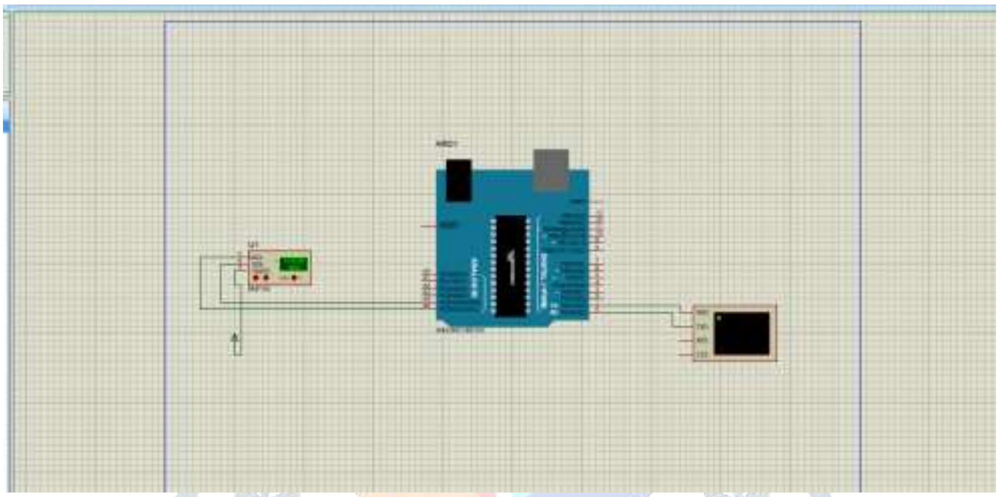


Fig 3.3 Arduino Uno Integrated Development Environment (IDE)

The open-source Arduino Uno environment makes it easy to write code and upload it to the I/O board. It runs on Windows, Mac OSX and Linux. The environment can be written in Java and based on Processing, avr-gcc, and other open source soft wares such as C and C++ programming language. The usage of the software is very easy once plugged, the correct port where the arduino is plugged must be selected and the type of the arduino as well, then, upload the code on the board and read the serial output. The Arduino micro-controller boot-loader calls the setup () function when it is started and it iterates forever over the loop () function.

Basic structure of Arduino Uno IDE code

```
Void setup ()
{
    // Put your setup code here to run once!!
}
Void loop ()
{
   //Put your main code here to run repeatedly!!
}
```

IV FLOW CHARTS:

```
// Temperature and Humidity sensor IDE code
#include <SPI.h>
#include <SD.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include <DHT.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
#define B1     A1
#define B2     A2
#define DHTPIN A3
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
File dataLog;
boolean sd_ok = 0;
```

```
char temperature[] = " 00.0";
char humidity[]    = " 00.0 %";
char Time[]   = "  :  :  ";
char Calendar[] = "  /  /20  ";
byte i, second, minute, hour, date, month, year, previous_second;
int Temp, RH;
void setup() {
 Serial.begin(9600);
 Serial.print("Initializing SD card...");
 if (!SD.begin())
  Serial.println("initialization failed!");
 else {
 Serial.println("initialization done.");
 sd_ok = 1;
}
pinMode(B1, INPUT_PULLUP);
pinMode(B2, INPUT_PULLUP);
lcd.begin(20, 4);
Wire.begin();
dht.begin();
lcd.setCursor(0, 0); lcd.print("TIME:");
lcd.setCursor(0, 1); lcd.print("DATE:");
lcd.setCursor(0, 2); lcd.print("Temp =");
lcd.setCursor(11, 2); lcd.write(223);
lcd.setCursor(12, 2); lcd.write('C');
lcd.setCursor(0, 3); lcd.print("RH  =");
Serial.println("   DATE     |   TIME     | TEMPERATURE | HUMIDITY");
if(sd_ok) {
 dataLog = SD.open("Logger.txt", FILE_WRITE);
 if(dataLog) {
   dataLog.println("   DATE     |   TIME     | TEMPERATURE | HUMIDITY");
   dataLog.close();
 }
 }
}
void DS3231_display(){
 second = (second >> 4) * 10 + (second & 0x0F);
 minute = (minute >> 4) * 10 + (minute & 0x0F);
 hour = (hour >> 4)     * 10 + (hour & 0x0F);
 date = (date >> 4)     * 10 + (date & 0x0F);
 month = (month >> 4)   * 10 + (month & 0x0F);
 year = (year >> 4)     * 10 + (year & 0x0F);
 Time[7]      = second % 10 + 48;
 Time[6]      = second / 10 + 48;
 Time[4]      = minute % 10 + 48;
 Time[3]      = minute / 10 + 48;
 Time[1]      = hour % 10 + 48;
 Time[0]      = hour / 10 + 48;
 Calendar[9] = year  % 10 + 48;
 Calendar[8] = year  / 10 + 48;
 Calendar[4] = month  % 10 + 48;
 Calendar[3] = month  / 10 + 48;
 Calendar[1] = date  % 10 + 48;
 Calendar[0] = date  / 10 + 48;
 lcd.setCursor(5, 0);
 lcd.print(Time);
 lcd.setCursor(5, 1);
 lcd.print(Calendar);
}
void blink_parameter(){
 byte j = 0;
 while(j < 10 && digitalRead(B1) && digitalRead(B2)){
   j++;
   delay(25);
 }
}
byte edit(byte x, byte y, byte parameter){
 char text[3];
 while(!digitalRead(B1));
 while(true){
  while(!digitalRead(B2)){
   parameter++;
   if(i == 0 && parameter > 23)
   parameter = 0;
   if(i == 1 && parameter > 59)
   parameter = 0;
   if(i == 2 && parameter > 31)
   parameter = 1;
   if(i == 3 && parameter > 12)
   parameter = 1;
   if(i == 4 && parameter > 99)
   parameter = 0;
   sprintf(text,"%02u", parameter);
   lcd.setCursor(x, y);
   lcd.print(text);
   delay(200);
 }
 lcd.setCursor(x, y);
 lcd.print("  ");
 blink_parameter();
 sprintf(text,"%02u", parameter);
 lcd.setCursor(x, y);
 lcd.print(text);
 blink_parameter();
 if (!digitalRead(B1)){
   i++;
   return parameter;
 }
 }
}
void loop() {
 if (!digitalRead(B1)){
   i = 0;
   hour     =edit(5, 0, hour);
   minute   =edit(8, 0, minute);
   date     =edit(5, 1, date);
   month    =edit(8, 1, month);
   year     =edit(13, 1, year);
   minute  = ((minute / 10) << 4) + (minute % 10);
   hour  = ((hour / 10) << 4) + (hour % 10);
   date  = ((date / 10) << 4) + (date % 10);
   month  = ((month / 10) << 4) + (month % 10);
   year  = ((year / 10) << 4) + (year % 10);
   Wire.beginTransmission(0x68);
   Wire.write(0);
   Wire.write(0);
   Wire.write(minute);
   Wire.write(hour);
   Wire.write(1);
   Wire.write(date);
   Wire.write(month);
   Wire.write(year);
   Wire.endTransmission();
   delay(200);
  }
 Wire.beginTransmission(0x68);
 Wire.write(0);
 Wire.endTransmission(false);
 Wire.requestFrom(0x68, 7);
 second = Wire.read();
 minute = Wire.read();
 hour   = Wire.read();
 Wire.read();
 date  = Wire.read();
 month = Wire.read();
 year  = Wire.read();
 DS3231_display();
 if(previous_second != second){
   previous_second = second;
   RH = dht.readHumidity() * 10;
   Temp = dht.readTemperature() * 10;
   if(Temp < 0){
     temperature[0] = '-';
     Temp = abs(Temp);
  }
  else
   temperature[0] = ' ';
   temperature[1] = (Temp / 100) % 10 + 48;
   temperature[2] = (Temp / 10) % 10 + 48;
   temperature[4] = Temp % 10 + 48;
   if(RH >= 1000)
   humidity[0]     = '1';
   else
   humidity[0] = ' ';
   humidity[1] = (RH / 100) % 10 + 48;
   humidity[2] = (RH / 10) % 10 + 48;
   humidity[4] = RH % 10 + 48;
   lcd.setCursor(6, 2);
   lcd.print(temperature);
   lcd.setCursor(6, 3);
   lcd.print(humidity);
   Serial.print(Calendar);
   Serial.print("   |   " );
  Serial.print(Time);
  Serial.print(" |   ");
  Serial.print(temperature);
 Serial.print("°C | ");
 Serial.print(humidity);
 if(sd_ok){
dataLog = SD.open("Logger.txt",FILE_WRITE);
 if(dataLog) {
 dataLog.print(Calendar);
       dataLog.print("   |   " );
       dataLog.print(Time );
       dataLog.print(" |    " );
       dataLog.print(temperature);
       dataLog.print("°C   |  ");
       dataLog.println(humidity);
       dataLog.close();
     }
   }
 }
 delay(50);
}
```

Barometric pressure sensor IDE code

```
        #include <Wire.h>
#include <Adafruit_BMP085.h>
Adafruit_BMP085 bmp;
  void setup() {
  Serial.begin(9600);
  if (!bmp.begin()) {
         Serial.println("Could not find a valid BMP085 sensor, check wiring!");
         while (1) {}
  }
}
  void loop() {
    Serial.print("Temperature = ");
    Serial.print(bmp.readTemperature());
    Serial.println(" *C");

    Serial.print("Pressure = ");
    Serial.print(bmp.readPressure());
    Serial.println(" Pa");

    // Calculate altitude assuming 'standard' barometric
    // pressure of 1013.25 millibar = 101325 Pascal
    Serial.print("Altitude = ");
    Serial.print(bmp.readAltitude());
    Serial.println(" meters");

    Serial.print("Pressure at sealevel (calculated) = ");
    Serial.print(bmp.readSealevelPressure());
    Serial.println(" Pa");

  // you can get a more precise measurement of altitude
  // if you know the current sea level pressure which will
  // vary with weather and such. If it is 1015 millibars
  // that is equal to 101500 Pascals.
    Serial.print("Real altitude = ");
    Serial.print(bmp.readAltitude(101500));
    Serial.println(" meters");
    Serial.println();
    delay(500);
}
```

LabVIEW programming:The following describes the data acquisition and presentation from sensors and integration of WSN system into sensors. The sensors data is logged in Datalogger that must be collected in a computer and manipulated. LabVIEW has VIs which supports serial communication.A LabVIEW program was made to display real time data of WSN sensors. The Figure 4-1 shows flowchart of program which was built in LabVIEW.

Flow chart:

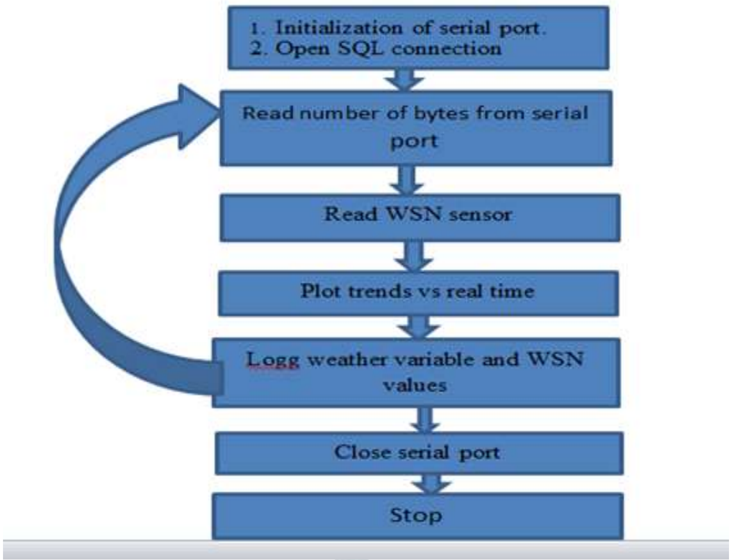

1. Initialization of serial port.
2. Open SQL connection
Read number of bytes from serial port
Read WSN sensor
Plot trends vs real time
Logg weather variable and WSN values
Close serial port
Stop

Figure 4.1 Flowchart diagram of LabVIEW program

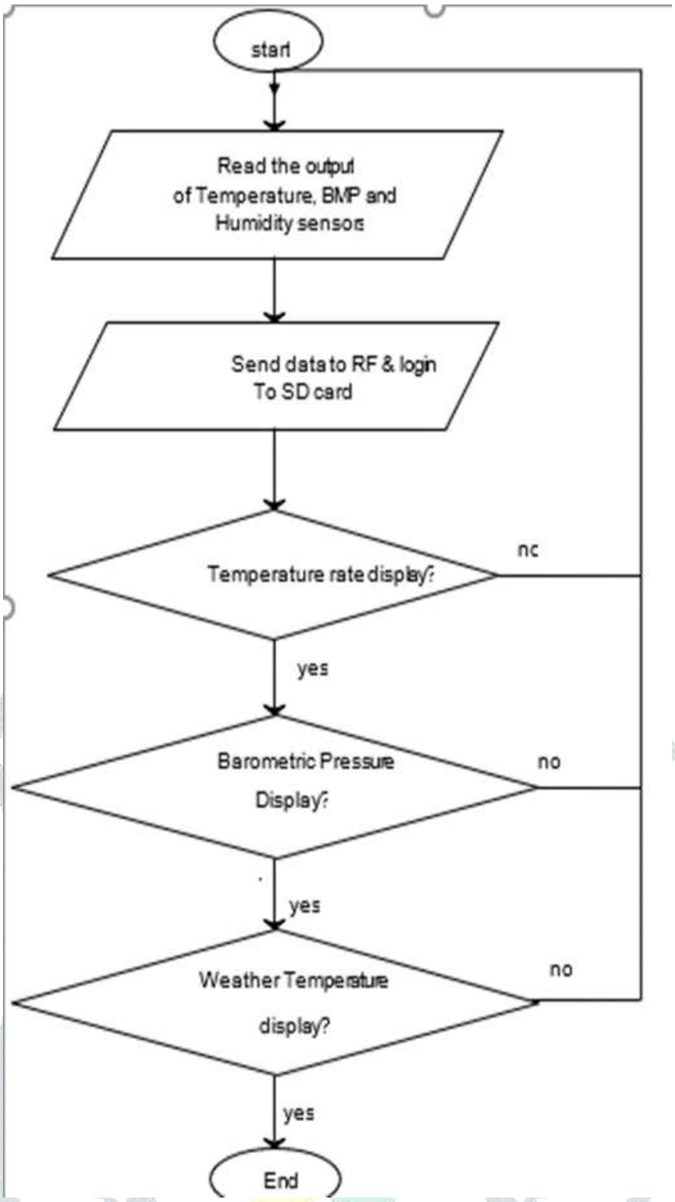

start
Read the output of Temperature, BMP and Humidity sensors
Send data to RF & login To SD card
Temperature rate display?
nc
yes
Barometric Pressure Display?
no
yes
Weather Temperature display?
no
yes
End

Fig 4.2 Flow char of work flow

The block diagram was made as state machine. A state machine consists a case structure inside a while loop. Case structure consists of many cases which contain a program code or a part of program code. In the block diagram first case 'Init' inside while loop contains the code which initializes the serial communication. Second case 'Visa Read' contains the code which reads the specified number of bytes from com port connected to computer. Also, the output string is manipulated and sensor values are extracted from string and indicated on front panel. The third case 'Read WSN sensor' displays plots real sensors values vs real time and d sensor are logged into a server called SQL server. In the next case 'communication is closed by a VI called VISA close. Each case in case structure specifies the different states in stare machine. Shift register carries values from one state to another. The following figure shows that when the program is run, to the next state specified in case structure.

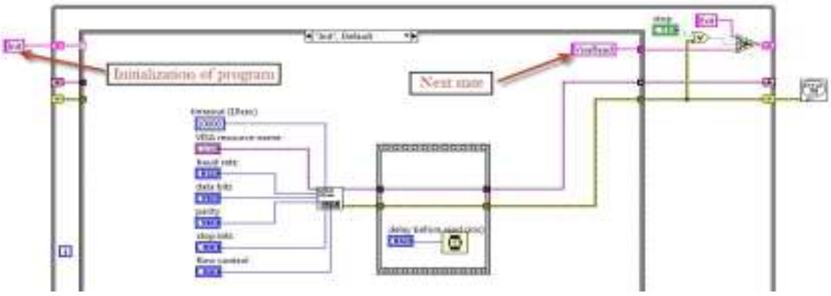
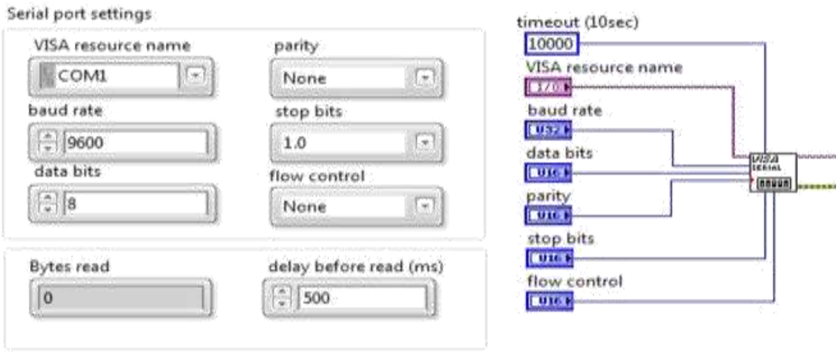


Figure 4.2 State machine

The following Figure 4.3 shows VISA Configure 'Serial Port VI' in the right configuration settings in the left. This VI initializes serial port communication. 'By indicator shows bytes read by the VISA read VI. 'VISA resource name's which will be opened which is serial communication port RS-232. Baud rate is the rate at which data transmission is occurred. 'Data bits' shows the number of bits in incoming data.



Serial port settings

VISA resource name: COM1

parity: None

baud rate: 9600

stop bits: 1.0

data bits: 8

flow control: None

Bytes read: 0

delay before read (ms): 500

Figure 4.3 VISA configure serial port VI on right and settings on left

The following VI plots the data vs date and time at which data is being displayed.

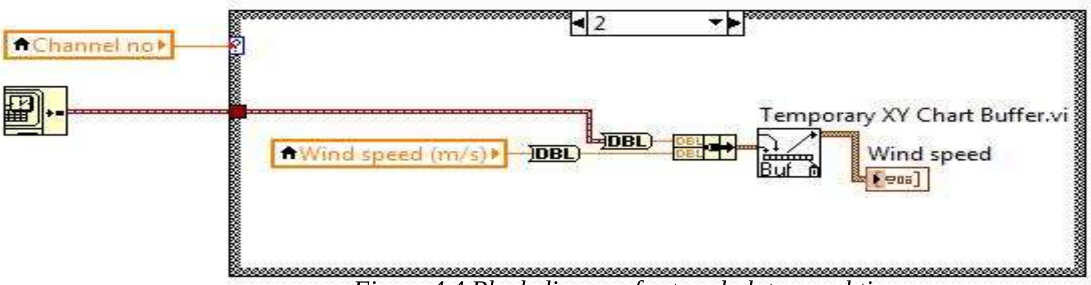


Figure 4.4 Block diagram for trend plot vs real time

The WSN system was integrated into weather station using following code.

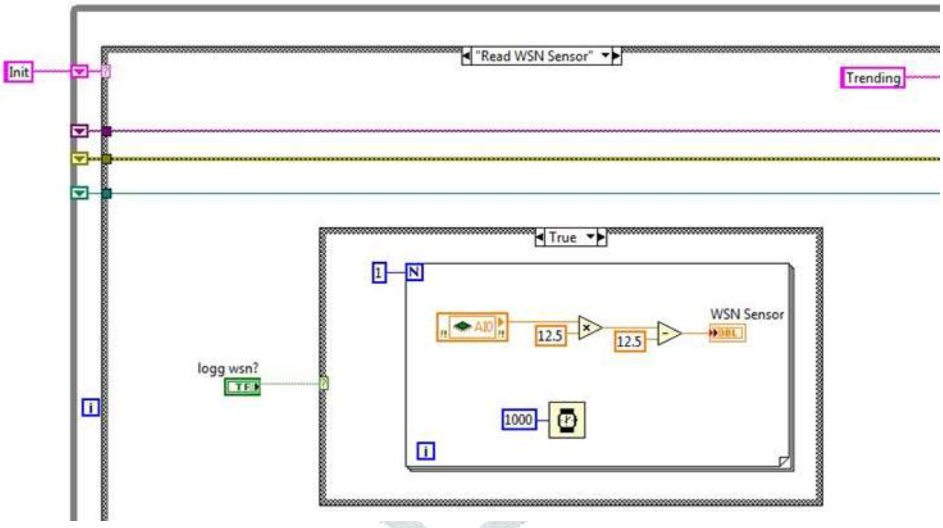
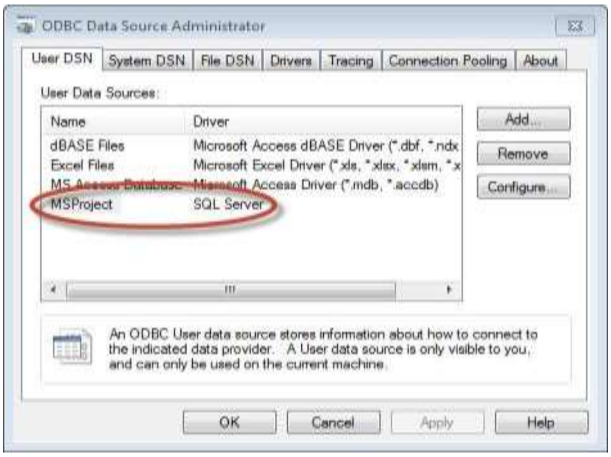


Figure 4.5 Block diagram for WSN

Data logging:Data logging is the process where information from sensing devices is stored in a device called Datalogger. This information is used to study the systems which are monitored. Data logging is used in many systems for example a black box is used to collect flight information which may be used latter. Data logging in weather station is important in the sense that weather condition is predicted on the basis of collected data.

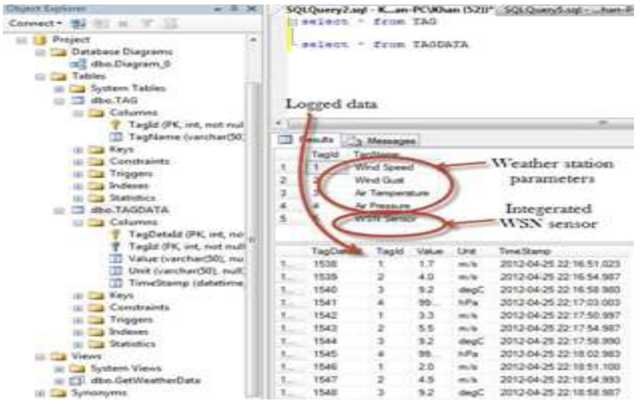
Datalogger was used to store information from weather sensors. Now the data must be downloaded and analyzed in order to know about the information which was captured by sensors. The data can be downloaded in text files or in other format and stored in computer. These files can be opened in spreadsheet. One solution is that data can be saved in software like SQL server. SQL is a relational database management system (RDBMS) from Microsoft. LabVIEW is a powerful programming tool that provides among others database communication. One must establish a connection to a database before accessing data in a table or executing a SQL query. There are different methods in order to connect to Database. In computing, Open Database Connectivity (ODBC) provides a standard software API method for using database management systems (DBMS). (Halvorsan, 2011) Here open database connectivity (ODBC) Database Source Name (DSN) was used. ODBC connection named 'MS Project' was Source created Administrator' 'ODBC Figure Data.T



User Data Sources:

Name	Driver
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.ndx
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.x
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)
MSProject	SQL Server

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.



```
select * from TAG

select * from TAGDATA
```

	TagId	TagName
1	1	Wind Speed
2	2	Wind Gust
3	3	Air Temperature
4	4	Air Pressure
5	5	WSN Sensor

	TagDataId	TagId	Value	Unit	TimeStamp
1...	1538	1	1.7	m/s	2012-04-25 22:16:51.023
1...	1539	2	4.0	m/s	2012-04-25 22:16:54.987
1...	1540	3	9.2	degC	2012-04-25 22:16:58.980
1...	1541	4	99...	hPa	2012-04-25 22:17:03.003
1...	1542	1	3.3	m/s	2012-04-25 22:17:50.997
1...	1543	2	5.5	m/s	2012-04-25 22:17:54.987
1...	1544	3	9.2	degC	2012-04-25 22:17:58.990
1...	1545	4	99...	hPa	2012-04-25 22:18:02.983
1...	1546	1	2.0	m/s	2012-04-25 22:18:51.100
1...	1547	2	4.5	m/s	2012-04-25 22:18:54.993
1...	1548	3	9.2	degC	2012-04-25 22:18:58.987

5.1 shows the created database connection in ODBC Data Source Administrator.

Figure 5.2 Database created in SQL server

Figure 5.1 ODBC data source administrator with created database connection, Two tables were created in SQL contains server to parameters while other table 'TAG DATA' the overview contain of complete database where two tables with their columns, database diagram and views are shown. 'View 'in SQL server is a table which is formed by selecting columns from other tables in the database named 'Get Weather Data'. A 'View data shown in Figure 5.3.

VI RESULTS AND DISCUSSION

VI.1. **Results**: The following figures show the result of this project. Before, after simulation works in LabVIEW software.

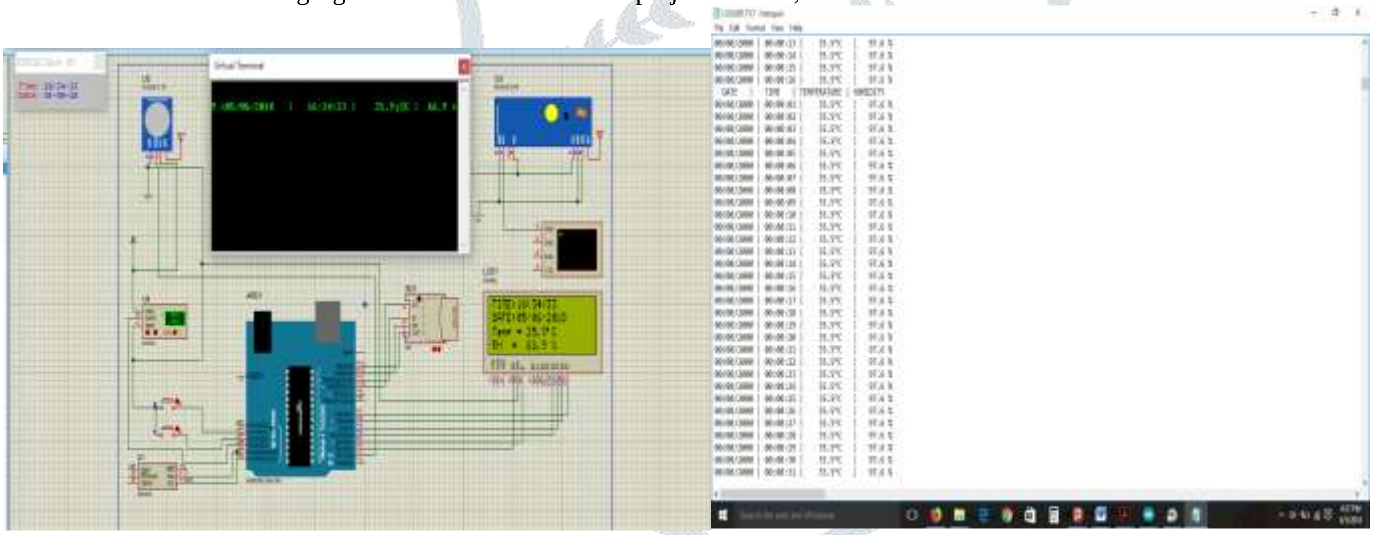


Fig 6.1 before simulation of circuit

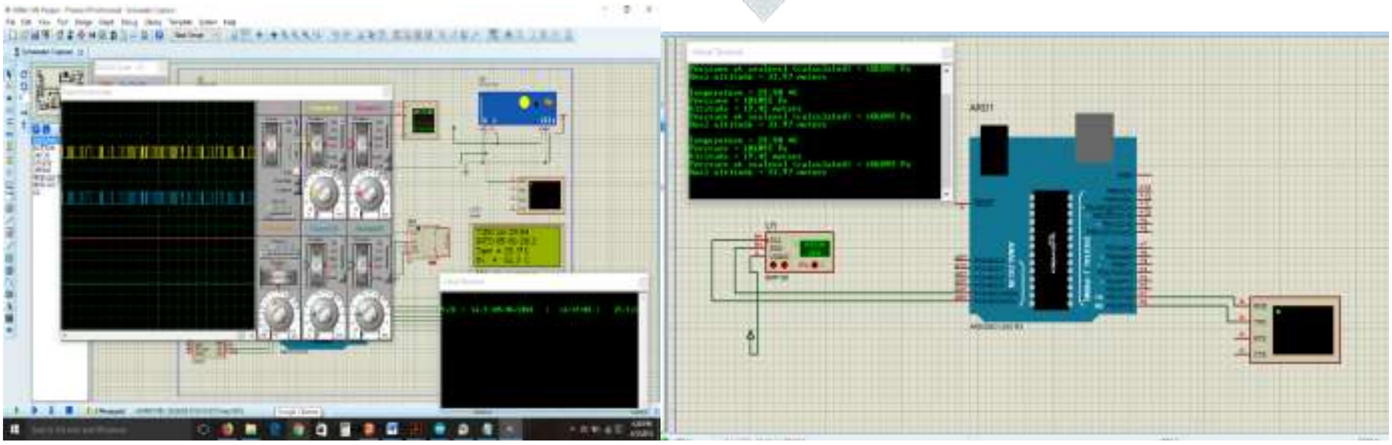


Fig 6.2 after simulation of circuit

VI.2. **Discussion**: The result of the project shows that the measured weather temperature and humidity is login into SD card and transfer to the gateway (receiver) wirelessly. The sensor data is stored in the SD card depend on the recording and transmitting interval time given. The data is date, time, second supported.

We were able to see only the simulation result of the Temperature, Humidity sensor and BMP180 pressure sensor. The reason why we did not get sensors of these parameters in the market hence, not actual measurements of Temperature, Humidity and BMP180 pressure sensor.

VII CONCLUSION AND RECOMMENDATION:

VII.1. Conclusion: From this project, we concluded that the developed system has simple structure, stable and reliable operation, high accuracy, low power consumption, low cost, full featured function and extensive application occasion.

Thus, this can be one of the best methods for automatic weather station applications where the man/data collector can analyze the weather condition of the climate from the remote place can they address update information to the recognized body timely. so that the human power and time can be reduced to the larger extent.

VII.2. Recommendations: Based on the developed system, the following recommendations were made:

- Taking our project as a basic foundation, it is better for interested ones (especially in metrology agency technical experts) to design and/or develop real-time automatic weather station system.
- As mentioned in chapter four, results and discussion section, we were unable to test temperature, humidity and bmp practically due to unavailability of the parameter sensors in the commercial market. So, any interested engineer or technologist can do more on this matter.

VII REFERENCE:

[1]A. Baggio, "Wireless sensor networks in precision agriculture", Workshop on Real-World Wireless Sensor Networks (REALWSN'05), June 20-21, 2005, Stockholm, Sweden.

[2]A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic, "Wireless Sensor Networks for Assisted-Living and Residential Monitoring", 2006 Department of Computer Science University of Virginia Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.3161&rep=rep1&type=pdf

[3]F. L. LEWIS, " Wireless Sensor Networks", Associate Director for Research , The University of Texas at Arlington

[4] Sohraby, K., Minoli, D., & Znati, T. (2007). Wirelesssensornetworks, Technology, Protocols, and Applications. John Wiley & Sns, Inc.

[5] Wikipedia. (2012). Retrieved april 2012, fromwikipedia.org: http://en.wikipedia.org/wiki/Automatic_weather_station

[6] Wikipedia. (2012, May 22). Retrieved May 24, 2012, from wikipedia.org: htttp://en.wikipedia.org/wiki/Atmospheric_pressure

[7] Wikipedia. (2012). Retrieved April 2012, from www.wikepedia.org:
http://en.wikipedia.org/wiki/Weather_station

[8] Buratti, C., Martalò, M., Verdone, R., & Ferrari, G. (2011). Sensor Networks with IEEE 802.15.4 Systems: Distributed Processing, MAC, and Connectivity (Signals and Communication Technology). Springer.

[9]Ferguson, S. S. (2007, December 27). wikipedia.org. Retrieved April 2012, from wikipedia.org: http://en.wikipedia.org/wiki/File:MQ-9_Reaper_in_flight_(2007).jpg

[10] Halvorsan, H. P. (2011). Database Communication in LabVIEW.

[11] Halvorsen, H. P. (n.d.). Weather Station at Telemark University College.http://home.hit.no/~hansha/.

[12] Mahalik, N. P. (2007). Introduction. In Sensor Networks and Configuration Fundamentals, Standards, Platforms,and Applications (pp. 1-4).

[13]National Instruments. (n.d.). Retrieved May 2012, from National Instruments: http://sine.ni.com/nips/cds/view/p/lang/en/nid/206919

[14]National Instruments. (n.d.). Retrieved May 2012, from National Instuments: http://sine.ni.com/nips/cds/view/p/lang/en/nid/206921

[15]NI.(2006, Septempber 6). National Instruments Corporation. Retrieved February 1, 2012, from http://zone.ni.com/devzone/cda/epd/p/id/2669

[16] Norwegian Meteorological Institute. (n.d.). Retrieved May 2012, from www.eklima.no

[17] Wang, Q., Hassane, H., & Xu, K. (2005). Handbook of sesnor networks: Compact wireless and wired sensing system. CRC Press.

IX BIOGRAPHY

[1]Dr K. Rameshbabu working as lecturer, in Communication stream, EET, Ethiopia, He holds B.E (ece), M.Tech, Ph.D having 20+ years of Experience . he is Guest professor for various colleges affiliated universities in India. He is member of IEEE, ISTE, VSI.IJCSIT, IRED he is reviewer for many journals & published 30+ papers in international journals and conferences & also conducted, attended many Faculty Development Programmers. his area of interest image processing, VLSI, digital electronics etc. He can reach at: kp.rameshbabu@gmail.com

[2]Mr Misay.Mangesthu working as a lecturer & head, EETM Department, FTVETI, AA Ethiopia. He holds B.Sc. M.Sc. He is also very good Teacher, Administrator. His area of interest in research is Electronics Communication, especially signal & image processing etc.

[3]MELAKU BIMER, BELETE GOSHUION W/MARIAM, are the PG students in branch of ECTM, EET Department, FTVTI, AA, Ethiopia. This is the PG thesis of them. We implemented hardware as well as software parts for use of Ethiopians as best our knowledge. Our area of interest is electronic devices & communication topics.