

# Controlling Air Conditioner (Temperature and Airflow) Using Wireless Sensor Network and iSense Technology

Karthik SK, Maheshwari Kittur, Suraj M Halakatti and Vineetkumar Patil

*B.V. Bhoomaraddi College of Engineering and Technology,  
Hubli, India.*

**Abstract**— This paper deals with the sensor technology to control the system wherein there are n-number of sensors which sense the temperature, aggregates them and transmits the aggregated data to the gateway. Weather sensor which can sense temperature, humidity and barometric pressure and uses IEEE standards 802.15.4 compliant radio. The information collected in the sensor is processed in iShell application, data is compared with the threshold value and control decisions are taken. This work contributes towards saving of energy in high power air-conditioner and air-conditioner in public places like malls, railway station and etc. The scope of this paper is ARM processor further it is made less complex, less size and cost effective embedded system. Camera sensor can also be used to capture the images of people, analyze the crowd density and control the air-conditioner depending on the crowd density.

## I. INTRODUCTION

Recently, several air-conditioning appliances that control the wind temperature and direction depending on the indoor temperature have been developed. Conventional air-conditioning systems may be effective for relatively small space (e.g. home-use). In a wide public space, however, such as railway station, airports, exhibition hall, flow of the people density does not remain constant and also prediction of increment or decrement of people density is not possible changes continuously so the temperature distribution does not keep constant. The crowd density is a parameter which influence the temperature variation. Conventional researches had tried to model-based approaches focusing on the wind convection, however, an integrated monitoring system of people positions and temperature distribution has almost never been proposed so far, though these two information are quite important to avoid ventilate a space where the temperature is sufficient or people do not exist. Therefore, in a wide public space, novel monitoring technologies for these dynamical situations, i.e. temperature distribution, are desired for effective air-conditioning system. That kind of technology has never been applied to a real-time control of building facilities. This system uses the wireless technology to the control the system wherein there are n-number of sensor which sense the temperature in the room, sends the signals to the gateways and further processed to take the readings. We are here using the weather module to sense the temperature in the environment. It has temperature range of -20C to 70.0C, frequency range of 433 MHz, Humidity ranges from 0 percent to 100 percent, Pressure ranges from 10 mbar to 1100

mbar. These motes are very advanced in nature, occupy very less space and work efficiently.

## II. MOTIVATION

With the modernization, use of energy has extensively increased. Fulfilling this energy has become a critical issue in shaping the revenue of the nation. Due to high energy consumption the nations are facing lot of problems regarding finance and all most all part of departments. After attending the workshop on iSense by BVBCET, we understood the importance these sensors in the current technology. A single sensor can sense a multiple parameters. Use of electricity has increased so much that almost all the equipment require electricity to run. Change in climate due to global warming has increased the use of electricity nearly 5 percent of all electricity produced in US only on Air Conditioning homes costing an astounding 15 billion annually.

## III. METHODOLOGY

a) Setting up wireless sensor network:

iShell application is used for storing the information sent by the gateway to the computer. iShell is an inbuilt application provided by the company for logging or recording the data sent by the gateway. iShell app is also used for burning the program on the motes. iShell app provides many other features like over the air programming, audio plugin etc. In iShell application initially we need to choose the plugins depending on our requirement for the program. So for our weather sensing application we need Flash Programmer and Serial monitor. Now let us understand how to start and burn the program on the motes.

Steps used in burning the sensor program on the motes:

- 1) Connect the USB gateway and start the program.
- 2) Go to edit and click on Preferences.
- 3) Choose COM 3 as port and click on OK.
- 4) Go to Flash Programmer
- 5) Click on Read MAC address to read and verify the USB connection of the gateway.
- 6) In File loader, browse file and choose weather sensing bin file.
- 7) In File information, browse file and choose weather sensing or32 file.
- 8) After choosing bin and or32 file, click on Start to ash the program on the motes.

- 9) After flashing gets completed, click on Serial Monitor to see the readings sent by the motes.
- 10) In serial monitor there is a red button which can be used to store the information in a file that is it can be used for logging the data (temperatures).

Steps used in setting up a wireless sensor network:

- 1) Burn the temperature sensing program on to the motes.
- 2) Place the motes properly in the room.
- 3) Trigger the motes to sense the temperature.
- 4) Once the motes have sensed the temperature they will forward it to the gateway.
- 5) Trigger the gateway to collect all the information sent by each mote.
- 6) Gateway sends all the collected information to the computer.
- 7) Information (temperature reading) received by the computer is stored in a file.



**Figure 1 iSense Gateway**

#### b) Embedded Systems:

We have developed a C application by name MySerialApp. This program is used for serial communication between the micro-controller and the computer. MySerialApp application provide features like analyzing the log file, setting the threshold temperature, writing the data to the port that is sending the data to the micro controller based on the average temperature obtained from the log file. We have also provided an email feature. A mail is sent to the recipient mail id notifying that the operator has to regulate the knob if any circuit has broken in the system.

Steps used in setting up the embedded systems:

- 1) The information sent by the gateway to the computer is recorded in a file.
- 2) iShell app is used for recording the information to the file.
- 3) MySerialApp program is executed (how to open and run the app will be described later).
- 4) The threshold temperature is set.
- 5) By default threshold temperature is 25 degree Celsius.
- 6) Log file is selected.

- 7) Selected log file is analyzed.
- 8) The threshold temperature and the average temperature obtained after analyzing the log file is compared.

Depending on the difference in the temperature, appropriate data is sent to the micro-controller.

As stated earlier, MySerialApp has many features but mainly we have divided our application into major two modules. One module deals with the selection of file, analyzing the temperatures and finding the average of all the temperatures. After finding the average temperature, it compares the average temperature with the threshold temperature and finds the difference. If the difference is greater than 25 degree Celsius and less than or equal to 30 degree Celsius then it falls in class A. If the difference is greater than 30 degree Celsius and less than or equal to 35 degree Celsius then it falls in class B. Similarly for every 5 degree Celsius difference, the class changes. Mainly we have four classes that is class A, B, C and D. Depending on the class the step of the motor changes that is after analyzing if the difference of the average temperature and the threshold temperature falls in class A, then step motor rotates at least speed and as the class goes on increasing that is from A to B, B to C and so on, the speed of the motor also goes on increasing that is the motor start rotating at a higher speed. When a file is analyzed and once the class is decided, a mail is sent to the recipient mail id indicating that the system is going to change the motor speed (fan speed).

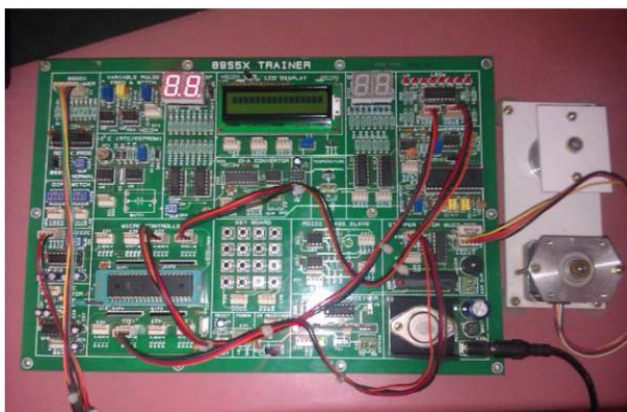
There is a separate section for sending mail. Default values are provided but user can change the recipient mail id, its subject and the message body. Fan speed in the application indicates that on which speed or class the motor is running/rotating. Filename label indicates that the file admin/user has selected for analyzing the temperature readings. Choose COM Port specifies that the user has to choose an appropriate computer port which is used for serial communication with the micro-controller. Usually it is COM 1 but may vary also.

We have provided a menu bar which has major two sections those are File and Help. File section provides the feature of restarting the application that is all the values, browsed file names, mail section etc will be reset. Its very similar to closing the application and again opening it. One more option provided by the File section is the exit of the application. The Help section provides all the necessary details about the buttons in the application like on the press of that button what all things are going to happen. For example if you choose Select File in the help section then a message box will be displayed with the text as This key helps the user to browse the log file that is required for the analysis.

Now lets us understand how to start and run the MySerialApp in detail.

- 1) Just double click on the MySerialApp icon, the application will be started.
- 2) First choose the COM port (Before choosing the com port make sure that you have connected the

- microcontroller to the computer and it should be switched on).
- 3) If you want you can even change the threshold temperature. By default it will be 250 C.
  - 4) After choosing the COM port then click on the select file button. It is like normal browse button.
  - 5) Go the folder where you have stored the logged files.
  - 6) Select the log file which you want to analyze and click on open. You can see the name and the path of the selected file in the text box in front of File name label.
  - 7) Note here at a time you can select only one log file.
  - 8) After selecting the log \_le, click on analyze button. The selected file will be analyzed that is each temperature is read and summation of all the temperatures done and average of all the temperature is taken. A message box will be displayed with the average temperature value, threshold temperature value and a conclusion note.
  - 9) Note here if the selected \_le is not in a proper format then an error message will be displayed in a message box.
  - 10) Depending on the temperature difference in the average temperature and the threshold temperature class is displayed in sthe text box in front



**Figure 2 8051 Microcontroller Kit**

c) Air Flow Control Systems:

To control the air flow we have used 8051 microcontroller. Since we are building a prototype now so we have used a stepper motor as a fan motor and made use of light emitting diodes (LED) for displaying the speed of the fan on the micro-controller. In the above module the temperature reading are recorded in \_le. Using MySerialApp application, recorded temperatures and the threshold temperature is compared and depending in the difference value the class value is decided and sent to the micro-controller. The data sent by the computer to the micro-controller is stored in a variable called as SBUF. Basically there are two types of communication, one is Parallel communication and the other

one is serial communication so we are using serial of communication. The data received stored in the SBUF is compared with the characters A, B, C and D. Depending on which character is matched the stepper motor is run with the appropriate delay. If character A is matched that means the fan speed should be one, so LED value is made as one which represents the fan speed and also stepper motor is made to rotate with high delay so that it rotates with the least speed similarly if character D is mapped then appropriate value is sent to the LED to display the fan speed and also stepper motor is made to rotate with very less delay so that the stepper motor rotates with highest speed.

We have made use of keil software to build the program for serial communication and rotation of the stepper motor and SpiPgm software to burn the built program on the microcontroller. The program is written in a normal C language.

Now let us understand how to burn the program on the microcontroller kit. AT first connect the micro controller to the computer. Connect programmers port and RS232 cable from the microcontroller to the computer. Connect the stepper motor to the pins given for stepper motor section in the microcontroller kit. Connect the pins from the port P1 (lower nib) to the stepper motor port K35 pin. Connect the P2 both upper and lower nibble pins to the LED upper and lower nibble respectively. Open the Keil software and choose serial hex project. Now compile and build the serial hex program. After building we will get a HEX \_le. Now open SpiPgm software. At first erase the program which was previously burnt on the kit. Then click on Open \_le button and select the serial hex \_le (.hex extension). After selecting the \_le, click on Program to burn the program on the microcontroller kit. Now your microcontroller kit is ready to receive the data serially from the computer.



**Figure 3 Stepper Motor**

**IV RESULTS**

1) Setting up a wireless sensor network:

After connecting all WSN devices, we obtain the temperatures from several areas or rooms. Different temperatures collected from

- BVB biotechnology auditorium  
 [2014-04-05 13:07:38.922] 0x158d0000148f08: 31.5 °C  
 [2014-04-05 13:07:39.281] 0x158d00001c88a8: 31.3 °C  
 [2014-04-05 13:07:40.295] 0x158d0000148f08: 31.5 °C  
 [2014-04-05 13:07:40.669] 0x158d00001c88a8: 31.4 °C  
 [2014-04-05 13:07:41.683] 0x158d0000148f08: 31.5 °C  
 [2014-04-05 13:07:42.073] 0x158d00001c88a8: 31.4 °C  
 [2014-04-05 13:07:43.025] 0x158d0000148f08: 31.5 °C  
 [2014-04-05 13:07:43.461] 0x158d00001c88a8: 31.4 °C  
 [2014-04-05 13:07:44.397] 0x158d0000148f08: 31.5 °C  
 [2014-04-05 13:07:44.850] 0x158d00001c88a8: 31.4 °C  
 [2014-04-05 13:07:45.770] 0x158d0000148f08: 31.5 °C  
 [2014-04-05 13:07:46.238] 0x158d00001c88a8: 31.5 °C  
 [2014-04-05 13:07:47.174] 0x158d0000148f08: 31.5 °C  
 [2014-04-05 13:07:47.627] 0x158d00001c88a8: 31.5 °C  
 [2014-04-05 13:07:48.516] 0x158d0000148f08: 31.5 °C

- BVB canteen  
 [2014-04-04 16:04:51.775] 0x158d00001c88a8: 32.8 °C  
 [2014-04-04 16:04:53.163] 0x158d00001c88a8: 32.8 °C  
 [2014-04-04 16:04:54.552] 0x158d00001c88a8: 32.9 °C  
 [2014-04-04 16:04:55.940] 0x158d00001c88a8: 32.9 °C  
 [2014-04-04 16:04:57.344] 0x158d00001c88a8: 32.9 °C  
 [2014-04-04 16:04:58.733] 0x158d00001c88a8: 33.0 °C  
 [2014-04-04 16:05:00.121] 0x158d00001c88a8: 33.0 °C  
 [2014-04-04 16:05:01.509] 0x158d00001c88a8: 33.0 °C  
 [2014-04-04 16:05:02.898] 0x158d00001c88a8: 33.0 °C  
 [2014-04-04 16:05:04.302] 0x158d00001c88a8: 33.0 °C  
 [2014-04-04 16:05:05.706] 0x158d00001c88a8: 33.0 °C  
 [2014-04-04 16:05:07.094] 0x158d00001c88a8: 33.0 °C  
 [2014-04-04 16:05:08.498] 0x158d00001c88a8: 33.1 °C  
 [2014-04-04 16:05:09.871] 0x158d00001c88a8: 33.1 °C

- Achiever lab  
 [2014-03-25 12:52:09.637] 0x158d0000148ef1: 33.3 °C  
 [2014-03-25 12:52:12.008] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:14.380] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:16.751] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:19.122] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:21.493] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:23.864] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:26.251] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:28.607] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:30.978] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:33.365] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:35.720] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:38.092] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:40.478] 0x158d0000148ef1: 33.2 °C  
 [2014-03-25 12:52:45.205] 0x158d0000148ef1: 33.3 °C

2) Embedded systems:

Here we have developed an application called MySerialApp to compare the threshold temperature with the

obtained temperature. It is an multipurpose application wherein we can :

- Select COM port
- Read from port and select the log file
- Analyze the obtained temperature with threshold temperature
- Intimate the user using E-Mail

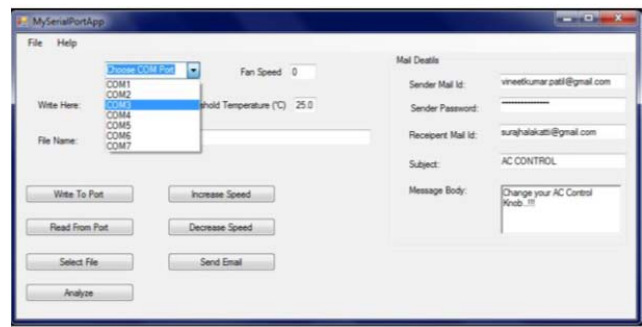


Figure 4 MySerialApp

3) Air Flow System:

Here we are sending the data to microcontroller in turn the stepper motor is rotated as to control the knob system.

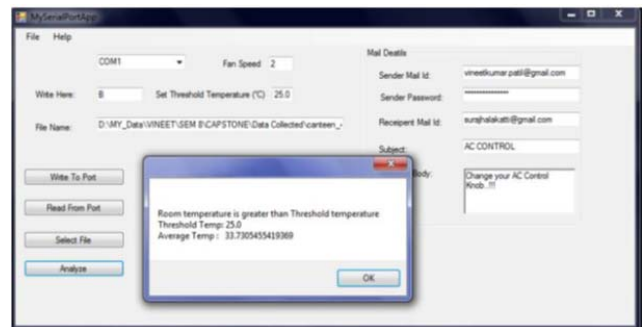


Figure 5 MySerialApp Threshold Value

CONCLUSION AND FUTURE WORK

The main objective of the project is to control the air flow was attained. We were successful in building wireless sensor network and controlling the stepper motor. Now we have used stepper motor as a prototype for our project. So the project should be extended to work with the real time air conditioning system as well as air flow system.

REFERENCES

[1] KyoichiroKatabira, Huijing Zhao, Yuri Nakagawa, Ryosuke Shibasaki, "Real-Time Monitoring of People Flows and Indoor Temperature Distribution for Advanced Air-Conditioning Control" in IEEE 802.15.4 Image Vision Compute ,Conference on Intelligent Transport Systems, Beijing, China, 2008.

[2] Vaishali P. Harak, Madhavi. H. Nerkar, "Energy Saving System for Air-Conditioning Control by Real Time People Detection" in IEEE 802.15.4, Image Vision Compute, S.V.I.T. P.D.V.V.P.C.O.E. Ahamadnagar, India, 2013

[3] G. Chen "The Role Switching Mechanism of Mobile Random-net", in IEEE 802.15.4, Image Vision Compute, Master Degree Thesis of

- Communication and Computer Engineering Dept., National Cheng-koung University, Tainan, Taiwan, 2005.
- [4] V.D. Cunsolo, S. Distefano, A. Puliafito, and M. Scarpa, Volunteer Computing and Desktop Cloud: the Cloud @ Home Paradigm, Image Vision Compute Proc. Of the Eighth IEEE International Symposium on Network Computing and Applications, Cambridge, MA, USA, 2009, pp. 134-139.
- [5] C.J. Huang, "Grown Blue-trees An Effective Topology for Bluetooth Scatter-nets" Image Vision Compute ,Master thesis, E.E. Department of National Chung-Sun University, Kaohsiung, Taiwan, 2003.
- [6] J.Y. Jung and J.W. Lee ZigBee device access control and reliable data transmission in ZigBee based health monitoring system, Proc. the 10<sup>th</sup> International Conference on Advanced Communication Technology Image Vision Compute, Phoenix Park, South Korea, 2008, pp. 795-797.
- [7] N. Kushiro, S. Suzuki, M. Nakata, H. Takahara, and M. Inoue, Integrated Residential Gateway Controller for Home Energy Management System Image Vision Compute, IEEE Transactions on Consumer Electronics, vol. 49, no 3, 2003, pp. 629-636.
- [8] C.H. Lien, Y.W. Bai, and M.B. Lin, Remote controllable power outlet system for home power management, Image Vision Compute, IEEE Transactions on Consumer Electronics, vol. 53, no. 4, pp. 1634-1641, 2007.