ISSN 2321 – 2004
ISSN 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013

Weather Sensing System to Mobile and Internet

N.D.Raja[1], P.P.Vel[2]

Assistant Professor, ECE, Cauvery College of Engineering and Technology, Trichy, India[1 & 2]

Abstract— The objective of this paper is to wireless weather sensing network through mobile devices or internet. Mobile devices these days are not only used for calling or texting, they offer a wide variety of features and sophisticated sensors such as e-mail, internet, and GPS. Their improvement in hardware and software is remarkable, offering new ways to develop different kinds of applications exploiting all of their capabilities. In this paper, I present the approach to integrate a personal mobile device application and wireless sensing network for data monitoring purposes. This project is designed to collect weather data from remote stations which can then be viewed on a mobile phone in real-time. Of course we can see the results using internet also. This is a type of data transfer through wireless communication by PIC microcontroller. This system will be useful in biomedical applications, seashore areas, transport regions, food divisions, rice mills and power production department. Depends upon the corresponding sensor readings, amplifier/convertor sends the data to microcontroller. Those values will be indicated in LCD display at the transmitter side itself. Then using zigbee transceiver, those values are transmitted to receiver section. From receiver section that value is transmitted to PC by using RS232 cable. In PC we can see the value which was displayed at transmitter section. And finally this value will be uploaded in World Wide Web (internet).

Keywords - GPS, PIC Microcontroller, RFD, FFD, Zigbee, PAN

I. INTRODUCTION

With the increasing growth of the cellular and Wi-Fi network infrastructure, internet mobile based applications are continuously being developed daily. Mobile devices these days are not only used for calling or texting, they offer a wide variety of features and sophisticated sensors such as e-mail, internet, and GPS. Several research papers on mobile devices were published. First one is Accelerometer sensor to classify human activities such as running, walking, or biking. Without any retraining of new users, the proposed system was able to classify different subjects with 97% accuracy rate. Another one is demonstration the method of graphing and monitoring sensor data on the wireless sensor networks. Using the web browser in a mobile device, users can access history data stored in the database server and visually plot them on the screen. Our project will discuss the wireless weather sensing network over 802.15.4 which allows users to access the sensor data, over cellular or Wi-Fi networks, directly from the web application. Using iPhone SDK, the data can be displayed as well as a link to satellite map view of its location.

This intelligent system is loaded with different type of Sensors, Amplifiers, PIC Microcontroller, LCD and Zigbee. Adopting a standard interface for sensors and actuators allows reuse of common hardware and communication protocol such as communication interface and control algorithm software. Instrumentation and control standards for RS-232 serial (voltage based) and RS-485 (current based) communication protocols have been widely applied and well documented for integrating sensors and actuators, particularly in industrial applications. The WSN (wireless sensor network) eliminates the need to hard wire sensor stations across the field and reduces installation and maintenance costs. The WSN uses an *ad hoc* network, i.e., a mobile wireless network.

II. SYSTEM DESIGN

A. Sensor Node and Wireless Communications

As illustrated in Fig. 1, each of the wireless weather sensing nodes is composed of several components, e.g. solar power unit, sensor interface, microcontroller, and wireless transceiver. There are four sensor types integrated in each node; velocity, temperature, moisture, and radiation. The microcontroller, based on the Microchip PIC16F877A, periodically gathers data from these sensors. The sensors are connected to analog-to-digital convertor pins to convert to a digital number format. Once the CPU receives all of the sensors' data, the next step is to manipulate the information into a presentable format. Each sensor requires different types of calculations in order to convert the raw data. Furthermore, it is necessary to filter out some of the sensors' data due to their fluctuation during data Acquisition. The moving average filter is applied and based on the following equation,

$$y(t) = \alpha\, y(t\text{-}1) + (1 - \alpha)x(t)\ ,$$

Where $y(t)$, $x(t)$, and α are output sample, input sample, and the smoothing parameter, respectively.

The data packet, as shown in Fig. 2, contains not only the measurements from all of the sensors, but it also includes the station ID and the start and stop bytes. The packet is composed of 21 bytes where the data of each sensor occupies a 3 byte hexadecimal number which is sufficient to capture the full resolution of the sensor values. Once the package is complete, the data is sent serially to the transmitter unit. Using the IEEE protocol 802.15.4, the Zigbee transceiver module transmits the data every five minutes to the gateway coordinator. To extend the battery life, the transceiver module remains in sleep mode the entire time except for 100 ms during the transmission of the data packet. While the module is powered on, it draws approximately 0.2 watts which is 100 times the power it uses while in sleep mode. Several types of RF transceivers combined with different antennas were tested at the beginning. The frequency ranges tested were the 900 MHz and 2.4 GHz bands. Three types of antenna combinations were tested with the transceivers; embedded, U.FL, and SMA. After testing all the modules and antennas, the 2.4 GHz frequency range was selected. A SMA connector on the module was decided upon for it has much higher RF gain capabilities for outdoor use. For this project, a 2.1 dBi gain antenna was attached to the station module and gateway coordinator.

B. Gateway Coordinator

The coordinator used to control the mesh network is the Digi Connect Port X8 Gateway [9]. The Connect Port is a standalone coordinator that is connected to a separate network via an Ethernet connection allowing it to be a mediator between the Zigbee PAN (personal area network) and a separate network. For this work, the coordinator is connected to the University's local area network to gain access to database servers.

ISSN 2321 – 2004
ISSN 2321 – 5526

IJIREEICE

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013

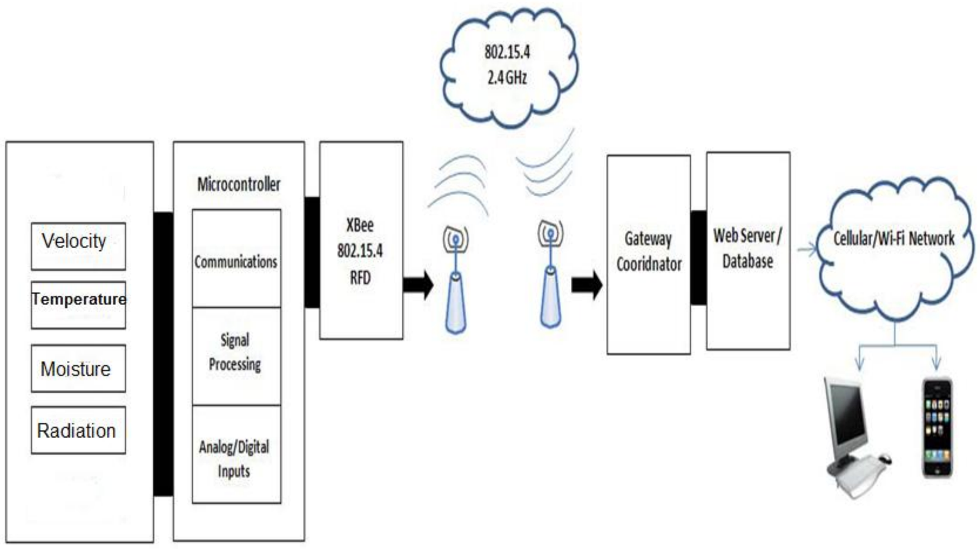
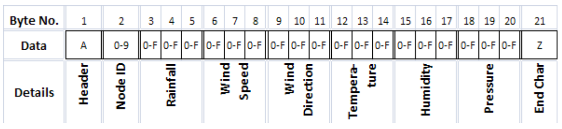


Fig. 1. A diagram of wireless weather sensing network on mobile devices

The communication between the Zigbee PAN and the separate wired Ethernet connection is controlled with custom Python scripts built into the coordinator performing several tasks. The program first opens a socket connection to accept all incoming Zigbee communications. The Zigbee is a high level implementation of the 802.15.4 specification meaning that all the low level programming is already taken care of in the embedded system. Once a data packet from a known FFD (full function device) or RFD (reduced function device) on the same PAN is received, the program stores the packet into memory and
blocks communications until the script is done running. After a new data packet is in memory, several checks are performed on the packet. The length of the packet, start and stop bytes, and range of data are checked for each packet. If the data seems to be acceptable, the packet is split into each individual piece of weather data and converted to the proper units. The weather data is then transmitted to the storages.



Byte No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Data	A	0-9	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	Z
Details	Header	Node ID		Rainfall			Wind Speed			Wind Direction			Tempera-ture			Humidity			Pressure		End Char

Fig. 2. Data Packet Format

C. Database Servers

To get the data to the storages, a standard TCP/IP socket on port 80 is opened from the gateway coordinator to the
server being used. A string is created and formatted as a Hypertext Preprocessor, PHP, query string and sent to a predetermined web page. Once the data is parsed at the web page, several checks are performed on each sensor data to assure that it is within acceptable ranges. If the data fits the correct criterion, the web page connects to the database and transmits the data. Java's mySQL database was chosen in this project. For the PHP page to insert data into the database, a query string is formatted and then sent to the mySQL server using the proper mySQL commands. These commands insert the data into the proper fields. At this point, a timestamp is also created and entered into the database whenever the data is being sent to the database server. All of this data is entered into a table according to which sensor sent the data originally. If the data belongs to a
new sensor that has never been in the database table, a new table with the proper fields is automatically created. A separate table holds the location of every remote station that has sent the data. This table contains the name, address and GPS coordinates for each station that can be used with Google Maps later. Finally, there is a table for each station holding the past week's worth of weather data. All of the user interfaces use these databases to pull the most recent data.

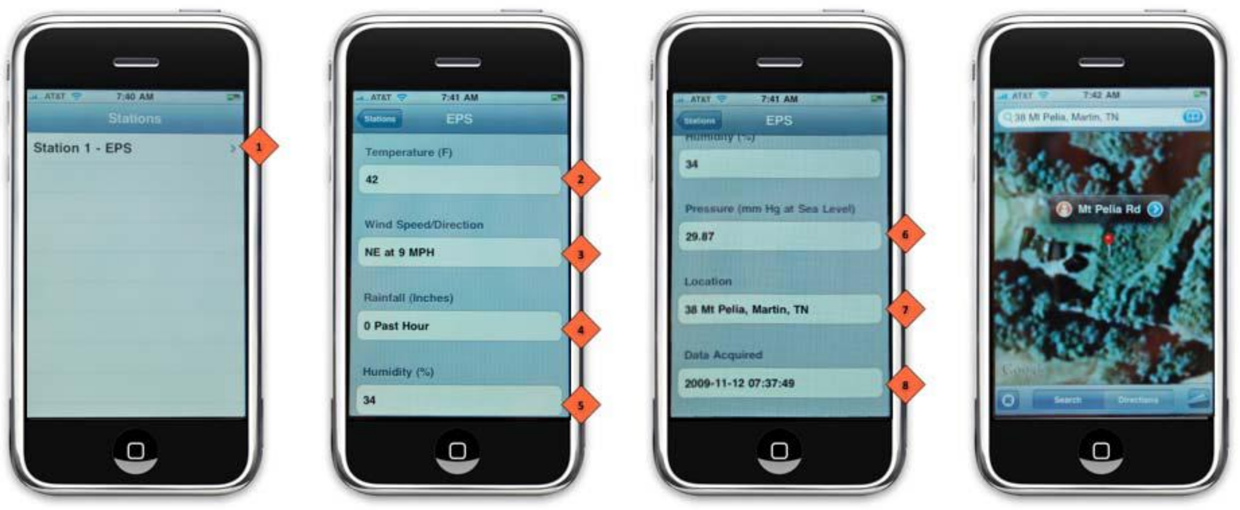
D. iPhone SDK

The iPhone SDK version 3.0 was used in this project
[10]. The OS is comprised of 4 layers including the core OS, Core Services, Media, and Cocoa Touch layers. Most of the programs in this paper were implemented using Cocoa Touch and Objective-C code. Cocoa touch is the set of Objective-C frameworks that provide the building blocks for iPhone applications. Contained are all the user interface widgets, events and event loop management, APIs to respond to touch, gestures, movement, access to the camera, file system, and other device features [10-11]. Once users launch a compiled application, the program grabs an XML package from a dynamic webpage, discussed in section C, representing all stations and their current data set. The following data shows an example of XML etrieved package and would show on the mobile after reformatting as
illustrated in Fig. 3. After receiving all data, the program parses each station data element and set the value of each corresponding station model object property. Then it creates a table view on the iPhone with a cell for each station. Tapping a certain cell presents a detailed view of that station. A detailed view presents each data point from the model object in a cell with a new table view. For example, tapping a cell for location would launch the Google Maps application. At this point, the program passes a URL to Maps so that station geographical location is highlighted in map view.

ISSN 2321 – 2004
ISSN 2321 – 5526

IJIREEICE

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 1, Issue 3, June 2013



(a) (b) (c) (d)

Fig. 3. Screenshots of mobile applications for wireless weather sensing data;

(a) remote sensor node selection,

(b) first four sensor data,

(c) the rest of sensor data, and

(d) satellite view of the remote sensor node location.

III. RESULTS

Once the design phase was complete, the remote station was successfully built and installed on top of the University of Tennessee at Martin Engineering and Physical Science building (EPS). To access the station’s data, users would first run the application already installed on their mobile phone. Fig. 3 shows the screenshots of the user interface. After they choose the remote station that they would like to access by tapping on the screen, Fig. 3a (point 1), the program will display the sensors’ data as shown in Fig. 3b and Fig. 3c. The temperature, velocity, moisture and radiation data are displayed on the screen illustrated by points 2-5, respectively, on Fig. 3b. To continue viewing the rest of the data, users would slide upward on the screen to change from Fig. 3b to Fig. 3c. The pressure, location, and acquired time data will be displayed on the screen illustrated by points 6-8, respectively, on Fig. 3c. To view the location of the remote station on the map, users would tap on the address location, Fig. 3c (point 7).

The application will bring up the Google Maps with the marker on the station’s location. At this point, users can change the view mode and find the direction to the station’s location easily. In addition to viewing the current weather data as we demonstrated in Fig. 3, the project is also capable of graphing the history data of each sensor at different time frames. Due to the large amount of history data needed to be downloaded, the method of graphing the data would be through the webpage instead of directly downloading it to mobile phones. The graph view method was not implemented by iPhone SDK, but optimized to be suitable for the web browser in mobiles. The webpage, written in PHP language, allows users to query the most up to date data from mySQL database and displays that data in visual format using PHP graphing utility [12]. As illustrated in Fig. 4a, users first access the setup website [13] and select the remote station and type of sensor to view. The application is capable of viewing data in different time frames, e.g. 1 day to 1 week past. Once users select to plot, the screen will show the graph of that sensor’s data as shown in Fig. 4b and 4c. In these plots, the wind speed data is shown for the past 72 hours and 7 days.

IV. SOFTWARE

MPLAB IDE is an integrated development environment that provides development engineers with the flexibility to develop and debug firmware for various Microchip devices MPLAB IDE is a Windows-based Integrated Development Environment for the Microchip Technology Incorporated PIC microcontroller (MCU) and dsPIC digital signal controller (DSC) families.

A. MP LAB SIMULATOR

MPLAB SIM is a discrete-event simulator for the PIC microcontroller (MCU) families. It is integrated into MPLAB IDE integrated development environment. The MPLAB SIM debugging tool is designed to model operation of Microchip Technology's PIC microcontrollers to assist users in debugging software for these devices.

B. COMPILER HI-TECH C

A program written in the high level language called C; which will be converted into PIC micro MCU machine code by a compiler. Machine code is suitable for use by a PIC micro MCU or Microchip development system product like MPLAB IDE

V. EXISTING SYSTEM

So many papers were published in IEEE site related to this project. In that we have given two major papers only here as existing system. i.e.

(A) Reversing Radar System Based on CAN Bus (2009 IEEE ICIMA)

(B) Security Management System for Oilfield Based on Embedded Wireless Communication (2009 International Conference on Networks Security, Wireless Communications and Trusted Computing)

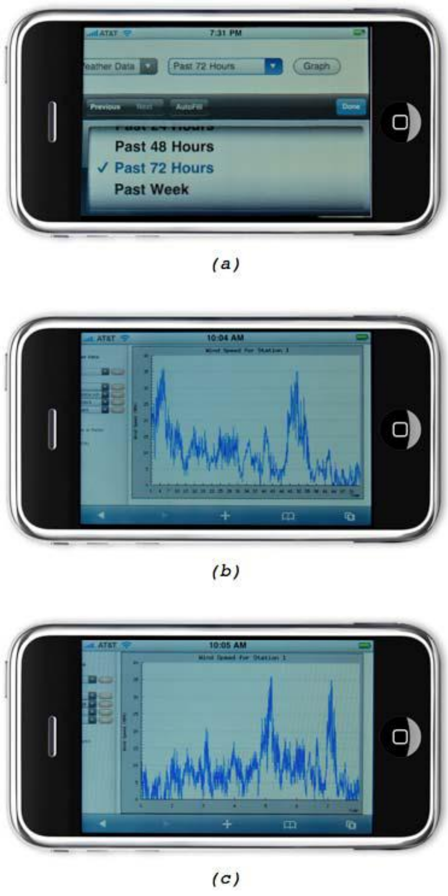


Fig. 4. Screenshots of viewing previous sensor data
(a) time frame of history data,
(b) wind speed plot for the past 72 hours,
(c) wind speed plot for the past 7 days.

VI. CONCLUSION

We presented the integration of mobile devices and a wireless weather sensing network for data monitoring and graphing. The mobile application was written in Objective-C language using iPhone SDK. Sensor nodes periodically sent data, over 802.15.4 network, to the database server. When a station was selected on the mobile devices, the weather data was pulled out and displayed as well as a location on a satellite map. The system was also capable of graphing history data on each sensor in a different time frame. The network and database server are fully expandable meaning it will automatically add new stations and tables without any significant modification. While the Zigbee performed as expected, it is limited by its range. For future work, different type of transmitters such as cellular network can be explored. In addition, we can apply a similar framework to different wireless sensing network applications, such as traffic or farm/vineyard monitoring systems.

REFERENCES

[1] http://www.engadget.com/2009/03/19/mobile-os-shootoutiphone- os-3-0-enters-the-fray/
[2] E. Miluzzo, J.M.H. Oakley, H. Lu, N. Lane, R. Peterson, A.Campbell, “Evaluating the iPhone as a Mobile Platform for People- Centric Sensing Applications,” in *Proceeding of International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems*, 2008.
[3] T. Saponas, J. Lester, J. Froehlich, J. Fogarty, J. Landay,“iLearn on the iPhone: Real-Time Human Activity Classification on Commodity Mobile Phones,” University of Washington CSE Tech Report UW-CSE-08-04-02, 2008.
[4] G. Wang, “Designing Smule’s iPhone Ocarina,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2009.
[5] D. Drager, A. Agah, and M. Asadi, “Sensor Graphing via Wireless Sensor Network to a Mobile Internet Device,” in *International Conference on Wireless Networks*, 2009.
[6] http://www.davisnet.com/weather/products/sensors.asp
[7] http://www.novalynx.com/110-ws-16bp.html
[8] http://www.meas-spec.com/product/t_product.aspx?id=2448
[9] http://www.digi.com/products/wirelessdropinnetworking/ connectportxgateways.jsp#models
[10] http://tuvix.apple.com/iphone/
[11] D. Mark and J. LaMarche, *Beginning iPhone 3 Development: Exploring the iPhone SDK*. Apress, 2009.
[12] http://www.aditus.nu/jpgraph/
[13] http://www.utm.edu/staff/knance/oldWeather.php

BIOGRAPHY

Mr.N.D.Raja has finished his UG degree in Anna University and PG degree in PRIST University. Currently he is working as an assistant professor in CACET, Trichy.

Mr.P.P.Vel has finished his UG degree in Anna University and PG degree in PRIST University. Currently he is working as an assistant professor in CACET, Trichy.