AN OPEN SOURCE WSN FOR INFRASTRUCTURE AND ENVIRONMENTAL MONITORING

[1]JISHNA.S, [2]RAJESH.M, [3]LEKSHMY.S

[1]Mtech Scholar, [2]Scientist/Engineer'C', [3]Assistant Professor ECE
[1,2]Vimal Jyothi Engineering College, Chemperi, Kannur ,Kerala ,India
[3]National Institute of Electronics and Information Technology Calicut, Kerala ,India
E-mail: [1]jishna.s92@gmail.com, [2]rajesh.m@nielit.gov.in, [3]lekshmyhari@vjec.ac.in

Abstract— A wireless sensor network consists of a set of sensor nodes that are widely distributed to sense any physical as well as environmental condition. They provide a bridge between real and virtual worlds. The power of wireless sensor networks has led to the development of large-scale systems for real time monitoring. The aim of this paper is to develop an open source embedded wireless sensor network for monitoring the infrastructure of the buildings and also the environment. The wireless communication scheme is implemented using the Zigbee . This is optimised to be a low power wireless sensor network. The system provides a web interface for any user to access the past and current sensor readings. Thus this network provides the real-time visualization of data.The simulation of the network is also been performed.

Keywords— Wireless Sensor Network, Sensors ,Zigbee , Simulator, microcontroller, Raspberry Pi, COOJA

I. INTRODUCTION

A wireless sensor network consists of a set of nodes that communicate by means of a radio interface. Low cost and wide range of applications of wireless sensor network has led to its increasing popularity. It has been proved more beneficial over the wired sensor network which has so many limitations. For the past few decades, there has been the development of various wireless sensor networks for infrastructure as well as environmental monitoring. Wireless sensing units, wireless communication networks and decision support systems are the important components in any wireless monitoring system.

The core component of any wireless sensor network is the sensor node. A sensor node consists of a microcontroller, radio transceiver, power supply module and a variety of sensors. It is always necessary to choose a low power consumption device as the microcontroller. The radio transceiver module uses the antenna for communication in different applications. The selection of power supply module is also important because the lifetime of the battery decides the lifetime of the sensor node. An operating system is the most important software running in a sensor node. Embedded operating system is always efficient than a general purpose operating system. In this project, embed OS is chosen as the operating system and its programming is done based on C language. In the recent years, a variety of sensors have been developed. These sensors can be used to monitor temperature, humidity, gas, pressure, movement etc. Several sensors can be integrated based on the application.

Buildings are often subjected to man-made as well as natural hazards. Hence it is necessary to employ a intelligent sensor network for monitoring the building infrastructure. The wireless sensor network developed in this project can be used to measure the temperature, gas content , humidity inside the buildings. It also measures the soil moisture content and humidity in the environment. These sensor networks play a very important role in mitigating the hazards. This sensor network is connected to the internet and hence it helps to reduce the future risks of occurrence of these hazards.

Some of the challenges faced in the adoption of wireless technology are as follows:

- Simplicity: the installation, deployment and maintenance should be simple
- Reliability: data must be delivered reliably by the sensors over the wireless channels at any condition
- Adaptability: adding and removing sensors to the system should be simple.
- Durability: components need to function properly for long duration without any replacement.
- Intelligence: scalable real-time validation, analysis and processing of the data needs to be available

Researches have always been carried out to develop a wireless sensor network that operate on low power and low data rate. This has led to the development of Zigbee technology. It provides low cost and low power consumption. Zigbee operates in three different unlicensed ISM radio bands : 868 MHz (Europe) , 915MHz (US) or 2.4GHz (global). The data rate is 20kbps at 868MHz,, 40kbps at 915MHz and 250kbps at 2.4GHz.

This project is an open source wireless sensor networks for environment and infrastructure monitoring. It enables the real-time collection and visualization of the sensor data. It also supports different hardware platforms. The integration of this system with the existing systems is also possible.

Different sensor units can be integrated into a simple unit. A reliable and energy-efficient communication scheme is also available. The Distributed Control Station (DCS) provides information monitoring in real-time.

The organization of this paper is as follows. In Section II, the hardware development of this project that includes its architecture and important components is discussed. In Section III, the results of both hardware and simulation is presented. Section IV draws the conclusions.

II. HARDWARE DEVELOPMENT

The system architecture of the wireless sensor network is as shown below:

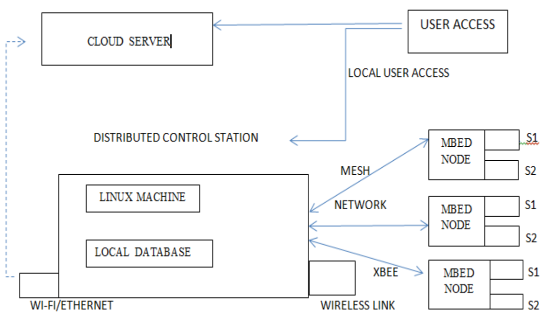

Fig 1. System Architecture

The heart of this wireless sensor network is the distributed control station(DCS). The other components include sensors, wireless sensor node, zigbee network and the cloud server The wireless sensor node consists of the sensing unit. The DCS coordinates a group of nodes to form a network. The data from the sensors is collected by the microcontroller and is transmitted to the base station via a wireless communication scheme zigbee. The data from the base station is then uploaded to the open source cloud Thingspeak and to the new local access webpage .The functionality and other features of each component is detailed in the remainder section.

2.1. Wireless Sensor Node

The wireless sensor node consists of a sensor unit , microcontroller, wireless transceiver and memory. Its functions include data collection from the sensors, compression and then transmission to the distributed control station.. The node contains Digital-to-Analog Converters (DAC) , Inter-Integrated Circuits (I2C) and Analog-to-Digital Converters(ADC) ports. These ports connect and communicate to the sensors.

The wireless sensor node used is LPC 1768. It is the board manufactured by the NXP Semiconductors. LPC 1768 is a cost effective, low power 32-bit ARM Cortex -M3 based microcontroller for embedded applications. It was chosen due to its high level of integration and low power consumption. It consists of 512KB Flash and 32KB RAM and a variety of interfaces that includes DAC, ADC, SPI, PWM, Ethernet, USB etc. The sensors used in this project are DHT 11 Sensor, MQ-7 Gas sensor and Soil moisture sensor. These sensors are connected to various ports of the microcontroller. The DHT 11 Sensor measures the temperature and humidity. The MQ-7 Gas sensor is used to measure the amount of carbon monoxide gas present in the air. It is helpful during the time of gas leakage and when unexpected fire occurs in the building. The soil moisture sensor is used to measure the amount of moisture present in the soil.

This project utilizes mbed operating system for the nodes. mbed is an open source operating system written in C and C++ programming language. mbed.org is an online compiler using which the programs can be compiled and then downloaded directly to the mbed LPC 1768 microcontroller. Since it is an online compiler ,the user can log in from anywhere and also there is no need to install or set-up a program. Since it uses ARM Real View Compile Engine ,it produces clean and efficient code. The mbed OS supports the connectivity technologies such as zigbee, Bluetooth, Wi-fi, Ethernet etc. mbed also provides reusable library functionality with clear interfaces and solid implementations. The core mbed Library supports the main LPC1768 peripherals, and the libraries already contributed by the mbed design community include USB, TCP/IP, and HTTP support.

2.2. Distribution Control Station

The functions of the distributed control station include coordinating the communication within the network, receiving data from the node and transmitting data to the cloud server. The DCS used in this project is the Raspberry Pi B+ board. This board was introduced by the Raspberry Pi Foundation in July 2014. It is a small credit card sized computer. Some of the features of this board are increased connectivity, improved power consumption and greater input output ports. This board uses a micro SD card for the operating system. This SD card is the brain of the Raspberry Pi board. The SD card used here is the Class 10 16GB card. Rasbian is the operating system that is loaded into the SD card to be used in Raspberry Pi. RPi executes the Python script to communicate with the node and the web interface. One of the zigbee modules is connected to the RPi. The collected data is also stored in a database in MySQL in the board itself. Hence a back-up of the sensor data is created. The MySQL database gets its data from the python program. The Raspberry Pi is connected to the internet via Ethernet cable. Hence a new webpage is created that contains the sensor data. This page is created using the PHP and HTML programming languages. This page can be locally accessed by the RPi users using the Ethernet. The new webpage is available at the url http://192.168.10.61 or at http://localhost.

2.3. Zigbee

Zigbee is the wireless communication scheme used in this project. Zigbee uses IEEE 802.15.4 standard radio as the hardware and carrier sense multiple access/collision avoidance (CSMA/CA) as the channel access protocol[1]. The Digi Xbee modules are used for transmission. Digi International's Xbee Configuration and Test Utility (XCTU) Software is used for configuration, communication and signal strength monitoring of the xbee modules. These modules can be configured as coordinator , router or end-device. The end device is connected to the wireless sensor node(LPC 1768) and the coordinator is connected to the distribution control station(Raspberry Pi). For certain networks, routers can also be employed in order to route the sensor data from the end device to the coordinator. The Xbee module supports two modes of communication for

sending and receiving data namely, Application Transparent (AT) mode and Application Programming Interface(API) mode. In this project, the Xbee modules are configured in AT mode. In this mode, the message data itself is send to the module and received by the destination device.
The XBee module is placed on Zigbee-USB interfacing board. This board is used to interface the xbee module with the laptop or desktop systems. Zigbee(Xbee) communicate through serial communication. Hence one end of the USB connected to the PC is detected as COM port. CP 2102 IC is the board used in this project. For using this board, the installation of CP 2102 Serial to USB Converter driver was performed.
In this network, the zigbee modules adopt a mesh topology. In this topology, in order to ensure that the messages reach their destinations, the network itself has a built-in intelligence. If there is a failure in the default route to a destination, the networks discovers alternative routes for delivering the messages. In a mesh topology, the top node acts as the coordinator.

2.4. Web interface

Raspberry Pi can also be used to access a web server.The sensor data available at the Raspberry Pi is uploaded to an open source web server called Thingspeak. It is an open source interface which listens to incoming data, timestamps it, and outputs it for both human users (through visual graphs) and machines(through easily parse-able code). Data is stored in channels and each channel allows to store upto 8 fields of data using 255 alphanumeric characters each. A Sequential ID is provided to all the incoming data. Once a channel has been created, data can be published by accessing the ThingSpeak API with a 'write key', a randomly created unique alphanumeric string used for authentication. In order to ensure the privacy of the data, 'read key' can be used. Simple Hyper Text Transfer Protocol is used to sent and receive data. This real time visualization of data is provided using the Representational State Transfer(REST) application programming interface.
The sensor data is also uploaded to a new webpage created. This new webpage is created using PHP and HTML script after installing apache-2 in Raspberry Pi. PHP is a server side scripting language that is embedded in HTML. The sensor data that is stored in the mysql database is then uploaded to this webpage. This webpage can be accessed by the local users using the Ethernet. Hence this webpage is available at http:192.168.10.61 or at http://localhost.

III. RESULTS

3.1 Simulation

Simulator is a very essential tool for developing a wireless sensor network. Since mbed OS does not have any built-in simulator, Contiki- COOJA simulator is used in this project. It is the network simulator available in the Contiki operating system. COOJA is a flexible Java-based simulator which supports C program language as the software design language by using Java Native Interface (JNI)[8]. The simulation reduces the development time of WSN.COOJA allows the simulation of the application software in both low level hard driver development and high level algorithm development simultaneously. In COOJA, it is possible to alter the parts of the simulation environment without any change in the main code. Hence new interfaces, radio mediums ,plugins etc can be added to the existing systems. It also allows the reconfiguration of the existing parts.
In COOJA ,the simulation interface consists of five windows. The Simulation Control window helps to control the rate of the simulation. It also helps to start, stop and reload a simulation. The Network Window shows the physical layout of the network. The Mote Output Window displays any output of the simulation produced as the result of printf command. The Timeline window shows the occurrence of events on each mote during the simulation. The Notes Window can be used to take down temporary notes.
This network consists of 25 motes. Z1 mote is used for simulation in this project. The Network window shows the random positioning of these motes. This is a simple data transfer taking place in a wireless sensor network. The power tracker gives the power consumption of each mote. The Base RSSI Configurator gives the Received Signal Strength values for each mote.

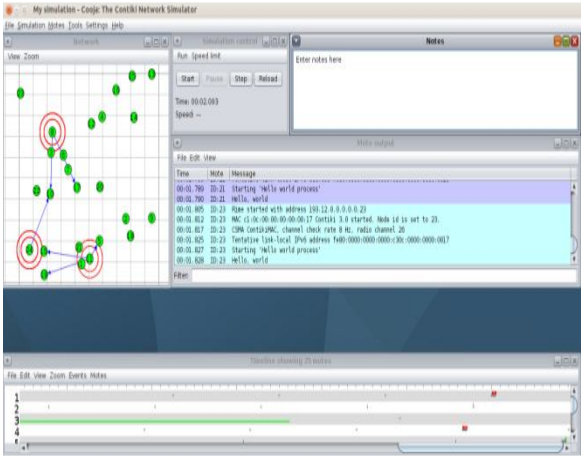

Fig 2 :Windows in COOJA Simulator for a network containing 25 motes.

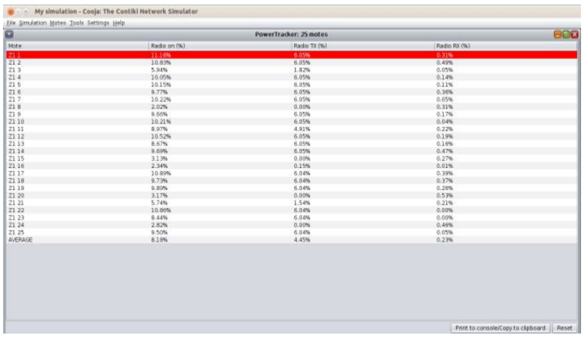

Fig 3 : Power Tracker window showing the power consumption of 25 motes

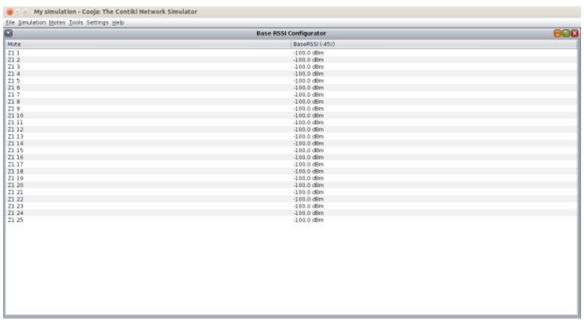

Fig 4. Base RSSI Indicator

3.2 Hardware

The sensors used in this project are DHT11 sensor, MQ-3 Gas Sensor and Soil Moisture Sensor. These sensors are connected to the mbed LPC 1768 microcontroller. Their output can be viewed in the microcontroller itself using the PUTTY terminal.

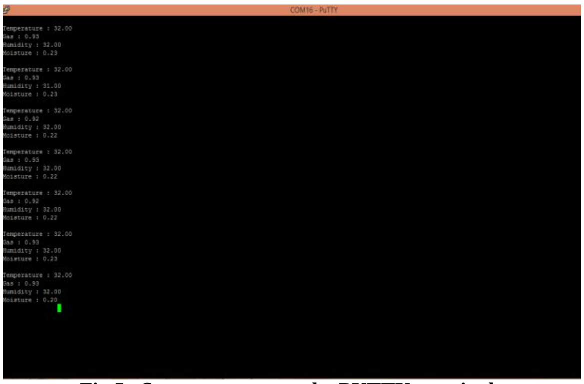

Fig 5. Sensor outputs at the PUTTY terminal

The Xbee end device is connected to the LPC 1768 mbed microcontroller. The sensor data is now transmitted to the Xbee device. It is then send to the Xbee coordinator in the distributed control station. In the DCS, the data is stored in the MySQL database. It is also uploaded to the new webpage. Real time visualization of data is available in the Thingspeak cloud server. The thingspeak output consists of the graphical representations of the sensor quantities namely temperature,gas,humidity and soil moisture.

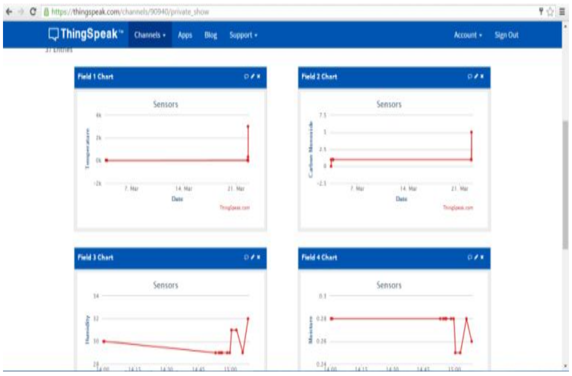

Fig 6. Thingspeak output

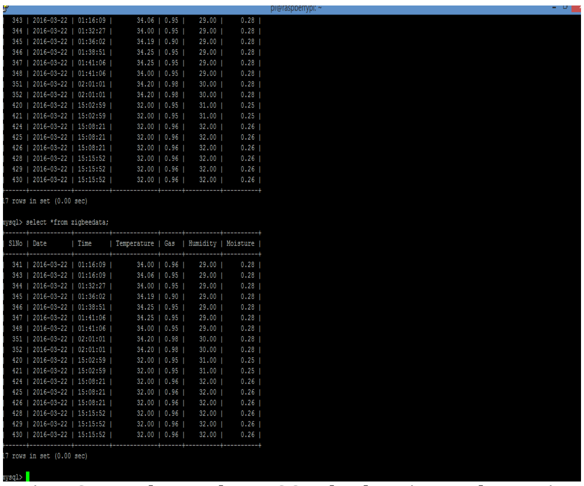

Fig 7. Sensor data at the MySQL database in Raspberry Pi

Fig 8. New webpage

CONCLUSION

In this paper , an open source WSN for infrastructure and environmental monitoring has been presented. The distributed control station plays the most important role. It provides the real time visualization of the sensor data and also interaction with the web interface. A back up of the sensor data is also created in the DCS itself. The adoption of zigbee for communication enables low power consumption and more reliability. This system also enables the integration of a variety of sensor units into an embedded device. The ouputs obtained at the various stages of the implementation has been displayed and explained.

REFERENCES

[1] Yizheng Liao, Mark Mollineaux, Richard Hsu, Rebekah Bartlett, "Snowfort: An Open Source Wireless Sensor Network for Data Analytics in Infrastructure and Environmental Monitoring", IEEE Sensors Journal, vol.14, no.12, December 2014.

[2] A.Rowe et al., "Sensor Andrew:Large-scale campus-wide sensing and actuation", IBM J.Res.Develop., vol. 55, nos.1-2, pp.6:1-6:14, Jan/Mar. 2011.

[3] T.Nagayama and B.F.Spencer, Jr., "Structural Health Monitoring using smart sensors," Dept. Civil Environ. Eng., Univ.Illinois at Urbana-Champaign , Urbana, IL,USA, Tech, Rep. NSEL-001,2007.

[4] Suresh Chaudhari, "Solutions for zigbee applications", Global Technology Centre, Vol.1, Juy 2011.

[5] Dr.Eben Upton, "Programming the Raspberry Pi", Raspberry Pi Foundation.

[6] Marcello A Gomez Maureira, DaanOldenhof, LiviaTeernstra, "Thingspeak- an API and web service for the Internet Of Things".

[7] AnujSehgal, "Using the ContikiCooja Simulator", Computer Science, Jacobs University Bremen Campus Ring 1,28759 Bremen ,Germany.

[8] ZhejunFeng, "Design and Implementation of an experimental platform for performance analysis in wireless sensor networks", TRITA-ICT-EX-2011:293.

[9] Kelvin Roussel, Ye-Qiong, Olivier Zendra , "Using Cooja for WSN Simulations : Some New Uses and Limits".

[10] Veronica Henry, "Introduction to Python Programming in Ubuntu".

[11] S.R.Dotty, "Python Basics", August 2008.

[12] VikramVaswani, "PHP-A Beginner's Guide",The McGraw-Hill Companies ,2009.

[13] DHT 11 Temperature and Humidity Sensor, D-Robotics, UK.

[14] MQ-7 Semiconductor Sensor for Carbon Monoxide, Henan HanweiElectronics Co.,Ltd.

[15] LPCXpresso Experiment Kit User's Guide

[16] DW 2013-Rapid Prototyping with mbed tutorial

[17] Configuring Serial Terminal Emulation Programs,Microsemi.

[18] A Quick Guide to MySQL Tables and Queries, EMBNet.