IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.
PP 01-08
www.iosrjournals.org

Energy Efficient Environment Monitoring System Based on the IEEE 802.15.4 Standard for Low Cost Requirements

P.Suchetha[1],T. Venu Madhav[2]
Embedded Systems, B V Raju Institute of Technology Narsapur /JNTU HYD, Telangana, India

Abstract : *In our daily activities, controlling and monitoring plays a vital role and it helps the end user to decide what to and what not to do for making the attempt successful. In this paper, sensors are the key elements which collect the data from surrounding and upload it to the web server. End user will monitor it from a simple customized web page with a login. ARM processor with Laptop/PC is used for logging the data into the server. Sensors used in this paper for prototype were Temperature Sensor, Alcohol Sensor, CO Sensor and a Humidity Sensor interfaced with ARM processor at analog port. Python API is used in the laptop for sending the data captured by the laptop's serial port into the web server.*

Keywords - *Python API, ARM Processor, Web Data, Internet of Things*

I. Introduction

Now-a-days, controlling and monitoring plays a main role in our day to day life. Everything we can control using advanced technologies and we can also monitoring the things we need [1,2]. Now we can control and monitor anywhere using Internet of things. If you have Internet in your PC/Mobile you can direct upload the data you need and control it from internet itself. When we talking about the Internet of things.

The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors and connectivity to enable it to achieve greater value and service by exchanging data with the manufacturer, operator and/or other connected devices. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.

Thing Speak is an open source Internet of Things application and API to store and retrieve data from things using HTTP over the internet or via a local area network. With think speak, you can create sensor logging applications and social network of things with status updates. In addition storing and retrieving numeric and alphanumeric data, the Thing Speak API allows for numeric data processing such as time scaling, averaging, median, summing, and rounding. Each Thing Speak channel supports data entries of up to 8 data fields, latitude, longitude, elevation, and status. The channel feeds support JSON, XML, and CSV formats for integration into applications.

The Thing Speak application also features time zone management, read/write API key management and JavaScript-based from High slide software

In this proposed solution, mainly three sensors are the key elements which collect the data from surrounding and upload it to webserver. End user will monitor it from a simple customized web page with a login. And those four sensors are Temperature Sensor, Alcohol Sensor, CO Sensor and Humidity sensor interfaced with ARM processor at analog port. Python API is used in the laptop for sending the data captured by the laptop's serial port into the web server [3.4].

II. System Architecture

The system architecture of this proposed system is divided into two different blocks.

2.1 ARM7 End:

Hardware implementation for this proposed system is shown below with the simple blocks. Power Supply block is designed and developed to generate power source for the ARM processor and its relevant components. Reset Circuit is designed and developed to reset the program whenever necessary and interfaced to the ARM processor for greater stable response. Clock Circuit is designed and developed to generate oscillations and interfaced to the ARM processor for needy response. LCD Display can also interface to the ARM processor for displaying the status of the system for better understanding. A simple block diagram shown below:

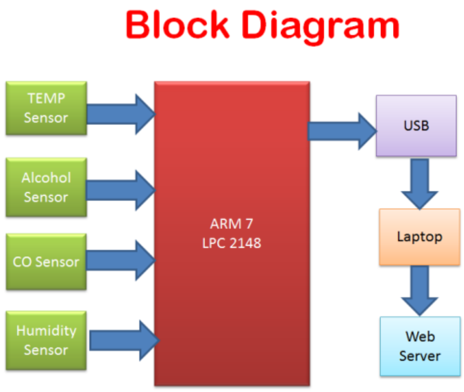

Fig – 1: Block Diagram

2.2 Server End:

A WEB SERVER is designed and developed for collecting the data from surroundings through Sensors and uploads it in to web server. Manual UI is designed for understanding of process with the help of HTML and PYTHON. Using the concept of Internet of Things we are uploading the each individual sensor values to the web server, there I can monitor the sensor values.

III. Implementation

3.1 Hardware:

In hardware implementation, ARM processor plays a key role in monitoring and controlling the security system. Low-power consumption ARM processor (LPC2148) operating at 3.3V, 50uA is designed and mounted on a PCB along with Reset Circuit and a Clock Circuit. LPC2148, a 32-bit microcontroller with advanced RISC architecture and having 48 GPIO lines with a program memory of 32KB and a data memory of 512Bytes.

And we have 2 UART ports i.e. UART0 and UART1. In this project XBEE connected to the UART0 port of ARM7 (LPC 2148). And 3 Analog to Digital channels, though I connected three Analog sensors to ADC channels of ARM7, so that it converts Analog Values to Digital Values. Those values i have uploaded into Thing Speak.

Each Sensors and its behaviour explained in below. And ARM7 (LPC 2148) internal architecture overview has shown below as well ARM7 (LPC 2148) with LCD has shown below.

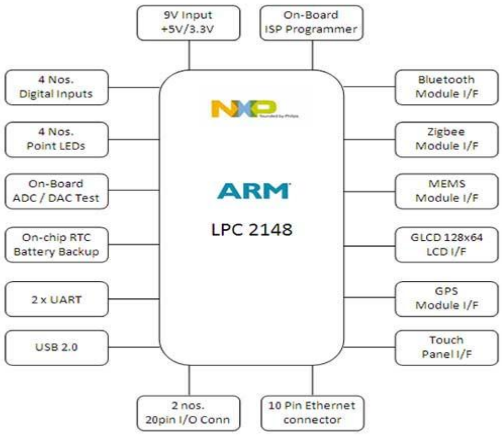

Fig – 2: ARM Overview [LPC2148]

Fig – 3: LPC2148 Development Board

3.2 Temperature Sensor:

The Temperature sensor (LM 35) has interfaced at ARM7. LM35 series are precision integrated-circuit temperature. The LM35 requires no external calibration since it is internally calibrated. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C Over a full -55 to +150°C temperature range.

The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The characteristics of LM35 sensor shown below:

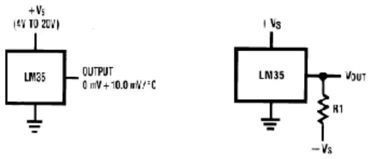

Fig – 4: LM35 characteristics

For each degree of centigrade temperature it outputs 10 mille volts.

In this project I've connected LM35 at P0.27 pin of ARM7 (LPC 2148). The output of LM35 would be in the form of Analog, to convert analog to digital I've used ADC. The output of LM35 given to the ADC and the output of ADC is in the form of digital. This output has given to the ARM7 and to the webserver through PC/Laptop.

The connection between ARM7 and LM35 shown below:

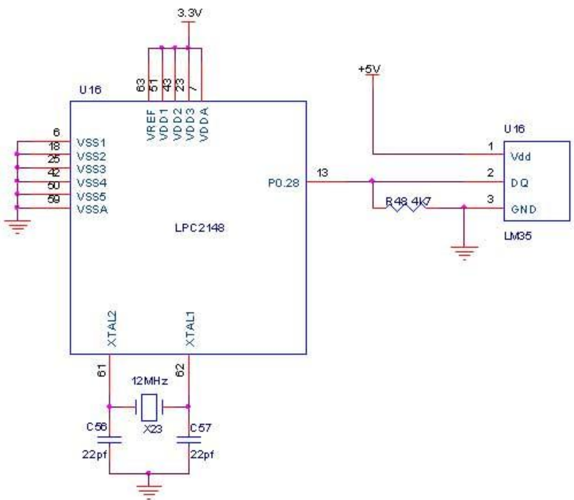

Fig – 5: LM35 interfaced to LPC 2148

3.3 Alcohol Sensor:

This is an alcohol sensor from futurlec, named MQ-3, which detects ethanol in the air. It is one of the straightforward gas sensors so it works almost the same way with other gas sensors. Typically, it is used as part of the breathalyzers or breath testers for the detection of ethanol in the human breath.

This alcohol sensor is suitable for detecting alcohol concentration on the human breath, just like the common breathalyzer. It has a high sensitivity and fast response time. Sensor provides an analog resistive output based on alcohol concentration. 5V DC or AC circuit, it requires heater voltage Operation Temperature: -10 to 70 degrees Heater consumption: less than 750mW, it is a high sensitivity to alcohol and small sensitivity to Benzine Fast response and High sensitivity Stable and long life Simple drive circuit.

Structure and configuration of Alcohol Sensor has shown below:

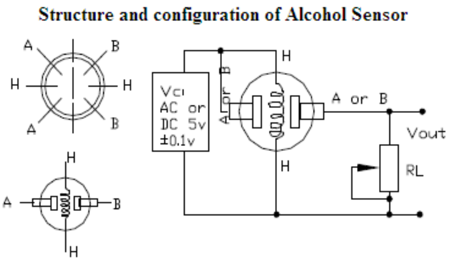

Fig – 6: Structure of Alcohol Sensor

3.4 Carbon Monoxide Sensor (CO):

CO sensor is a device that detects the presence of the carbon monoxide (CO) gas in order to prevent carbon monoxide poisoning. In the late 1990s Underwriters Laboratories (UL) changed their definition of a single station CO detector with a sound device in it to a carbon monoxide (CO) alarm. This applies to all CO safety alarms that meet UL 2034; however for passive indicators and system devices that meet UL 2075, UL refers to these as carbon monoxide detectors. CO is a colorless, tasteless and odorless compound produced by incomplete combustion of carbon containing materials. It is often referred to as the "silent killer" because it is virtually undetectable without using detection technology and most do not realise they are being poisoned [5,6]. Elevated levels of CO can be dangerous to humans depending on the amount present and length of exposure. Smaller concentrations can be harmful over longer periods of time while increasing concentrations require diminishing exposure times to be harmful [7,8].

CO detectors are designed to measure CO levels over time and sound an alarm before dangerous levels of CO accumulate in an environment, giving people adequate warning to safely ventilate the area or evacuate. Some system-connected detectors also alert a monitoring service that can dispatch emergency services if necessary.

While CO detectors do not serve as smoke detectors and vice versa, dual smoke/CO detectors are also sold. Smoke detectors detect the smoke generated by flaming or smoldering fires, whereas CO detectors detect and warn people about dangerous CO buildup caused, for example, by a malfunctioning fuel-burning device.

The circuit diagram of CO sensor has shown below:

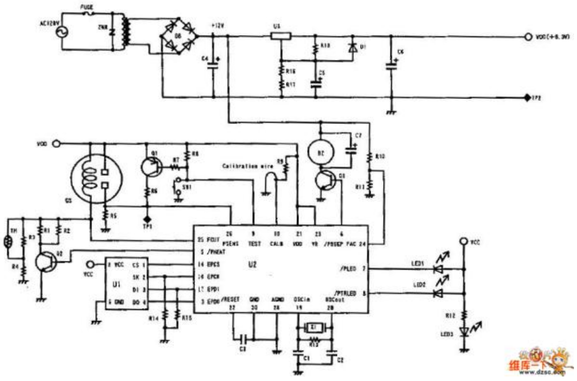

Fig – 7: CO sensor circuitry

3.5 Humidity Sensor:

This humidity sensor consists of 2 copper conductors that are located at small distance from each other. The relay switches as soon as the moisture makes a connection that is more or less electrically conductive between the 2 electrodes.

When the electrical resistance between the 2 sensors drops below a certain value then the Schmitt trigger (T1 and T2) switches. The RS N1/N2 bistable multivibrator is flipped through C1 so that in point B we have a low voltage and so T3 will close the relay.

The relay is opened when the 10K resistor is connected to point A, not to point B as shown in the schematic. You can use other sensors like LDR or NTC instead of the copper ones so you can use this circuit for detecting light or temperature.

The circuit diagram of Humidity Sensor has shown below:

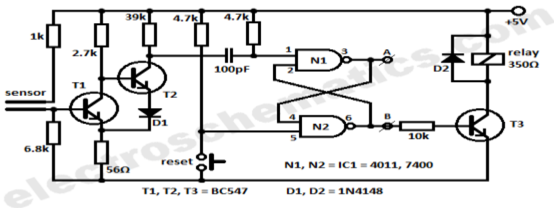

Fig – 8: Humidity Sensor Circuitry

All Sensors connected to ARM7 (LPC 2148) as per the following schematic diagram:

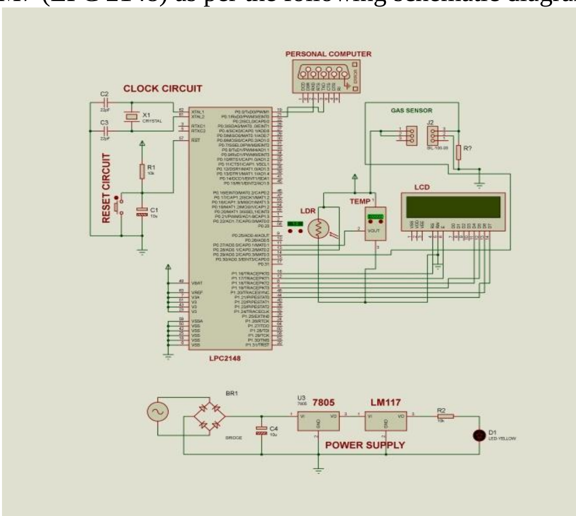

Fig – 9: Schematic Diagram

Temperature Sensor has connected at P0.28 of ARM7 (LPC 2148), CO sensor has connected at P0.29 of ARM7 (LPC 2148), Alcohol Sensor has connected at P0.30 of ARM7 (LPC2148) and Humidity sensor has connected at P0.31 of ARM7. All four Analog sensors are connected to Analog channel of ARM7. Reset Circuit and Clock Circuits were interfaced at RST, XTAL1, and XTAL2 of LPC2148

IV. Software:

Here, to program ARM processor Keil uVision 4 was used as a cross-compiler and Flash Magic was used as a programmer. Thing Speak is an open source Internet of Things application and API to store and retrieve data from things using HTTP over the internet or via a local area network

5. Algorithm and Flow Chart

5.1. Algorithm:

Step – 1: Initialize ARM, LCD and UART.
Step – 2: Wait until you see WELCOME on LCD.
Step – 3: You see three sensors default values.
Step – 4: Wait until sensors values have been changed.
Step – 5: UART port must be Open while sending the sensor values from processor to Thing Speak.
Step – 5: Now login to Thing Speak and Create channel and fields.
Step – 6: Now open the Python Program and run main.py, when sensors values changed then you can see the graph of sensors.
Step – 7: you can see sensors values in Thing Speak until power supply on.

5.2 Flow Chart:

The flowchart of this paper is shown below.

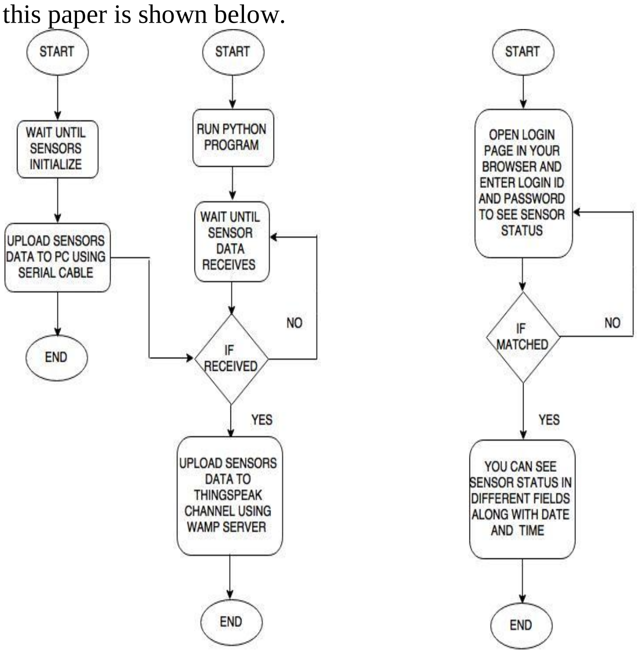

Fig – 10: Schematic Diagram

V. Results

Fig – 11: Server Login

Fig – 12: Sensor Data 1

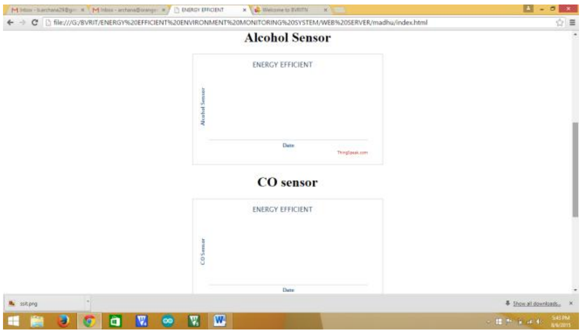

Fig-13: Sensor Data 2

Fig – 14: Sensor Data 3

VI. Advantages and Applications:

7.1. One can upload the data directly to Internet; each note can be converted into IOT Sensor Network.
7.2. Internet enabled platform will be more powerful than the remaining platforms
7.3. Wire Less Data Loggers with Internet enabled are the main areas where data will be sent to cloud easily.

VII. Conclusion:

Here, in this paper we tried to propose a new method for a Wireless Sensor Node to become an Internet of Thing Sensor Node where data acquired from IEEE 802.15.4 will be uploaded to Internet.

Acknowledgement:

We would like to express my special thanks of gratitude to Prof. Dr. I.A.Pasha (HOD), Department of ECE, Dr. B.V.Raju Institute of Technology who gave me the golden opportunity to do this wonderful project on the topic (Wearable Technology), which also helped me in doing a lot of Research and i came to know about so many new things we are really thankful to them.

And, secondly i would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

REFERENCES

[1] D. C. Uprety, S. Dhar, D. Hongmin, B. A. Kimball, A. Garg, and J. Upadhyay, "Technologies for climate change mitigation," UNEP United Nations Framework Convention on Climate Change (UNFCCC), Jul. 2012, ISBN: 978-87-92706-60-7.

[2] L. Hockstad and M. Weitz, "USEPA: Basic information and indicators in the United States," United States Environmental Protection Agency,Tech. Rep. EPA 430-R-13-001, 2013.

[3] N. Tanaka and F. Birol, "Climate change," in World Energy Outlook, Paris, France: IEA Publications, 2009, ch. 4, pp. 173–184, ISBN: 978-92-64-06130-9.

[4] R. D. Brook et al., "A statement for healthcare professionals from the expert panel on population and prevention science of the American Heart Association," Circulation, vol. 109, no. 21, pp. 2655–2671, 2004.

[5] NACA, Republic South Africa. (2012). National Association for Clean Air [Online]. Avalable: http://www.naca.org.za

[6] T. Tietenberg, M. Grubb, A. Michaelowa, B. Swift, and Z. X. Zhang, "International rules of greenhouse gas emissions trading: Defining the principles, modalities, rules and guidelines for verification, reporting and accountability," Documents United Nat., vol. UNCTAD-1,pp. 5–20, May 1999.

[7] O. A. Postolache, J. M. D. Pereira, and P. M. B. S. Girao, "Smart sensors network for air quality monitoring applications," IEEE Trans. Instrum & Meas., vol. 58, no. 9, pp. 3253–3262, Sep. 2009.

[8] N. Kularatna and B. H. Sudantha, "An environmental air pollution monitoring system based on the IEEE 1451 standard for low cost requirements," IEEE Sensors J., vol. 8, no. 4, pp. 415–422,Apr. 2008.