Real Time Data Acquisition over Cloud using Hardware Based WSN

Poonam Thakur
M. Tech Scholar
Priyadarshini Institute of Engineering & Technology
Nagpur, India
Poonamthakur1803@gmail.com

Prof. Rahul M. Pethe
Assistant Professor, Department of E & C
Priyadarshini Institute of Engineering & Technology
Nagpur, India
Rahul2480@gmail.com

Prof. Sunita I. Parihar
Assistant Professor, Department of E & C
Priyadarshini Institute of Engineering & Technology
Nagpur, India
Parihar_sunita@yahoo.co.in

***Abstract*—**Get ready to create distributed client node wireless sensor network using nRF24 wireless networking protocol to send their data on cloud through Ethernet based sink node. Over recent years, the WSN and their applications were under the focus of both academia and industry all over the world. Nowadays, the WSN are widely used in various applications areas: health care and medicine, home or office automation; industry; road traffic control; farming and forestry; civil infrastructure monitoring; disaster detection and alarm systems. Each of these areas required specific client node has specific environment and certain application requirements for the WSN. We are presenting the results and lessons learned during evaluating WSN by using Microcontroller, nRF24 with Ethernet Shield for WSN applications development and deployment that have been done within the use of Real Time Data Acquisition Process for Industrial Applications. For real time data acquisition system microcontroller Atmega32u4 is easily interfaced with nRF24. This sensor continuously generates enormous amount of data in the form of packets and frame (date, time, source address, destination address, data, ERC) which can easily reach to the destination and monitor on cloud by using Ethernet Shield and open source API i.e. Thingspeak which is a part of Internet of Things.

Keywords*—Wireless Sensor Network, Thingspeak, Packet transmission protocol.*

*****

I. INTRODUCTION

Consider a situation where many host can send their data and it can be download by sink node but some packet data may loss that can be retrieved by using some of the algorithm used for large data recovery. A wireless sensor network consisting of spatially distributed autonomous sensor equipped with low power transceivers can be an effective tool for gathering data in variety of environments. These nodes perform certain measurements and need to transmit all the collected information to base station over a wireless channel. The data are then processed in the base station to draw some conclusions about the current activity in the area. The idea of wireless sensor network is to interact with the environment. Wireless sensor network are usually embedded in an environment. Fig. 1 depicts Wireless sensor network connected to the internet.

Number of sensor nodes combines form a network. These wireless sensor nodes are equipped with sensors which are capable to measure environmental parameters i.e. temperature, pressure, humidity and more others. The measured data is processed into certain format and makes it available to different applications. The Internet of things is the network of physical objects with embedded technology to sense, communicate and interact with their internal states or the external environment [1]. In the WSN context, data may transmit by using different hardware nodes which can be transmit data to the cluster head of a network then cluster can transmit it to the base station. In this technique data can transmit over a small distance. Data can be transmitted over a large distance but it requires large power otherwise data may loss.

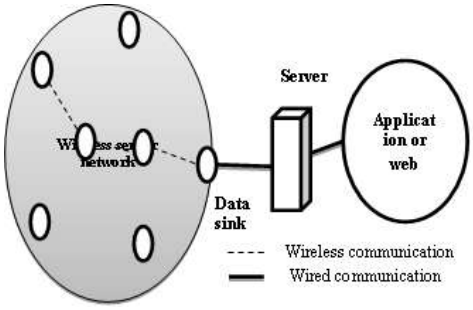

Fig. 1 Wireless Sensor Network

II. BACKGROUND

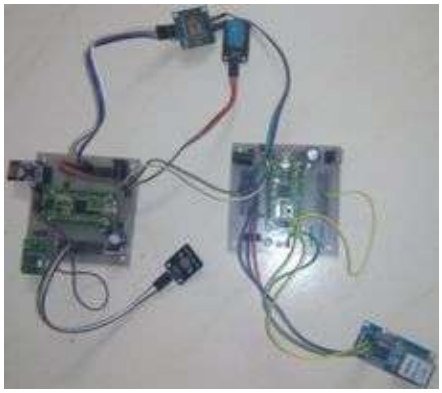

Fig. 2 Sink Node with Ethernet Node

A. Microcontroller

For this prototype, we are using Atmega32u4 as processor. There are many processors available like Atmaga8, Atmaga16, Atmega32, Atmega32u2 but because of some

advantageous specifications of Atmega32u4 (processor) over other processors, we have used Atmega32u4. The ATmega32U4 is low-power CMOS 8-bit microcontroller-RISC architecture. This microcontroller achieves throughputs of 1 MIPS per MHz. This optimizes power consumption and processing speed as it is executes many instructions in a single clock cycle. It instruction set with 32 general purpose registers. These 32 registers are directly linked to the Arithmetic Logic Unit (ALU), which allows two independent registers to be used in one single instruction executed in single clock cycle [4].

The ATmega32U4 provides the following features:

- It has 32K bytes of in system programmable flash.
- It has EEPROM of 1k bytes.
- It has SRAM of 2.5K bytes.
- It has 26 general purpose I/O lines.
- It has 32 general purpose registers.
- It has 4 flexible Timer/Counters with compare modes and pulse width modulation.
- It has high-speed Timer/Counter with PLL adjustable source
- It has 1 USART including CTS (clear to send) /RTS (request to send) flow control signals.
- It provides byte oriented 2-wire serial interface.
- It provides analog to digital converters instead of just analog comparators.
- It has an on-chip temperature sensor.
- It has a programmable watchdog timer.
- In idle mode the CPU is stopped while it allows SRAM, Timer/ Counters, SPI port and interrupts system to function.
- In Power-down mode register are saved but the oscillator is stopped, disabling all other chip functions until it receives the next interrupt or reset.
- In Standby mode, the Crystal oscillator is running while other devices are sleeping. This allows fast start-up and low power consumption.

The Atmega32u4 is based on RISC architecture is manufactured using high-density nonvolatile memory technique. Program memory is allowed to reprogram by on chip ISP flash through SPI serial interface, by a nonvolatile memory programmer, or it can also be programmed by the on-chip boot program. The 8-bit RISC CPU with ISP Flash on a monolithic chip, makes microcontroller to provide high flexibility and it is very cost effective for many embedded control applications.

B. Transmitter / Reciever – nRF24L01

We are using nRF24L01 transceiver in our design. It is a single chip 2.4GHz transceiver featuring embedded baseband protocol engine. It is designed for low power wireless applications. The nRF24L01 operates in the world-wide 2.4 – 204835 GHz ISM frequency band at [5]. It is operated through a Serial Peripheral Interface (SPI.) Register map is available to the transceiver through Serial Peripheral Interface. The map contains all the configuration registers in the nRF24L01. The embedded baseband protocol engine is based on packet communication and supports advanced autonomous protocol operation. Internal FIFOs of nRF24 makes sure that the data is flowing smoothly between the radio end and the system's Microcontroller. The radio end uses GFSK modulation. The air data rate supported by this transceiver can be configured to 2Mbps. The high air data rate featured with two powers saving modes makes nRF24 well suited for low power applications. Internal voltage regulators provide a very high Power Supply Rejection Ratio and a wide power supply range. As nRF24 is well suited for low power designs, we are using to reduce power consumption of the prototype.

Fig. 3 nRF24L01

C. Ethernet Shield

The Arduino Ethernet Shield connects your Arduino to the internet in mere minutes. Just plug this module onto your Arduino board, connect it to your network with an RJ45 cable (not included) and follow a few simple instructions to start controlling your world through the internet. As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Hundreds of thousands of Arduino boards are already fueling people's creativity all over the world, everyday.

- Requires an Arduino board
- Operating voltage 5V
- Ethernet Controller: W5100 with internal 16K buffer
- Connection speed: 10/100Mb
- Connection with Arduino on SPI port

D. Real Time Clock

In our design, RTC is connected to the server node, to collect the real time data. RTCs are used in WSN for periodic node wakeups. In WSN context, periodic node wakeups refers to wakeups based on events and it allows the processor to enter into deep sleep modes [13].

E. SD Card

We are using SD card for storage purpose. All the measured dat is simultaneously stored in the memory card so that it can be monitored further.

III. SENSORS

The best part of this application is sensors employed in this wireless module. In Wireless sensor networks, Desirable functions for sensor nodes include: ease of installation, self-identification, self-diagnosis, reliability, time awareness for coordination with other nodes. The wireless nodes are equipped with sensors due to which they are able to measure different parameters.

A. DHT11 Temperature and Humidity Sensor

DHT11 temperature and humidity sensors are digital sensors. DHT11 elements are extremely accurate on temperature and humidity calibration. Calibration Coefficients are stored in the OTP memory of DHT11 in the form of programme, which are used by internal signal detecting process of sensors. The sensor provides single-wire serial interface, this makes the system integration easier. DHT11 sensors are small in size, it consumes less power and up to20 meters signal transmission. Because of these advantages of DHT11 sensors are used in various applications. It is a 4-pin single row pin package [6]. It is convenient to connect to the sensor node. The interfacing of DHT11 sensor with microcontroller is shown in Fig. 4.

Fig. 4: DHT11Temperature and humidity sensor

B. PIR Sensor

The pyroelectric infrared sensor detects infrared radiation on the basis of the characteristics that the polarization of pyroelectric material changes with temperature. Dual compensated sensing elements are applied to suppress the interference resulting from temperature variation. As a result, the operating stability of the sensor is greatly improved [7].

IV. LITERATURE SURVEY

In 2001, *Schurgers* et al worked on 'Energy Efficient Routing in Wireless Sensor Networks'. His first approach was to use a concept termed as Data Combining entities or DCE's. This concept is similar to clustering, instead of designating a cluster head; it picked up a node that has other streams of network traffic flowing through it as the DCE [8]. The second approach discussed by *Schurgers* et al to reduce energy consumption in a wireless sensor network is the spreading of network traffic over the entire network. In the same year, *Slijepcevic et al* in the paper 'Power Efficient Organization of Wireless Sensor Networks' discussed on reducing the overall power in the network system by grouping the sensor nodes into mutually exclusive sets in 2001 [9]. This technique assumed that the nodes are placed stochastically.

"Maximizing System Lifetime in Wireless Sensor Networks" by *Dong*, is one of the first papers to differentiate between the "time" and "transmission" approaches to overall lifetime of a wireless sensor node network in 2005 [10]. He considered many different time based and packet based models. *Chao-Lieh Chen et al* authored the paper "Energy-proportional Routing for Lifetime extension of Clustering-Based Wireless Sensor Networks" in 2007. He presented an algorithm to determine the energy usage for nodes in an upcoming round of data collection and transmission; it then determines if a cluster-head or a node should be used for forwarding tasks or transmits data to intermediate hops [11].

In 2008 *Kim et al* in his paper 'Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks with Anycast' pertained a unique type of wireless sensor network, namely an event-driven network that uses ''Anycast". Utilized a wake-sleep cycle for the sensor nodes, i.e. instead of having one node that a sensor node will transmit its messages to, a node will have a group of candidate nodes that it can transmit to [12]. *Jiang et al* worked on classification of WSN clustering schemes based on 8 clustering attributes in 2009. He also analyzed popular WSN clustering algorithms and also comparision between those algorithms [13]. *Vipul Gupta* worked on developing a web-based service called Sensor network that facilitates a heterogeneous mix of devices to interact with one another in the paper 'Sensor Network: An open data exchange for the web of things' in year 2010 [14].

V. SOFTWARE

In this project we used Arduino software available as an open source platform which makes it easier to write program and upload it to the circuit boards. Basically there are two nodes used in any network i.e. Server node and Client node. Server node is always connected to the serial port of the computer. During initialization, server node initializes RTC (Real time Clock) for real time data acquisition, nrf24L01 (transceiver) for packet data transceiver and SD card for data storage and other parameter which are connected. On the other hand the client node also initialized by software with to observe packet data transmitted or not then it will be powered by an adapter and kept it away from server node in range of transceiver. Client node will send the measured data in the form of packets to the server node.

Address (Hide)	Date (YY:MM:DD)	Time (HH:MM:SS)	Node Name	Data	ERC

Fig. 6 Packet Format

Fig. 6 shows the packet format of data transmits and received by client node and server node.1st block indicated by source and destination address, 2nd and 3rd block for date in YY: MM: DD and time in HH:MM:SS format, 4th block indicate node name which is mentioned in program, 5th block indicate data collected by the sensors connected with client node and send it to server node, 6th block for error check whether data is properly send by client node and received by server node.

VI. RESULT

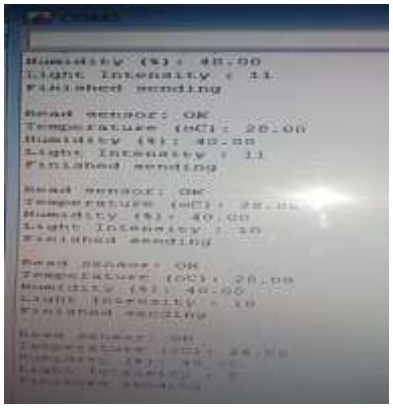

Fig. 7 Packet format send by Client node.

Fig. 7 shows the packet format send by client node and also monitor whether it has been properly send by client node or not. Whenever the first packet finish sending then and then only second packet send by client node otherwise stop sending.

Fig. 8 Packet format received by server node.

In fig. 8 we can see the format of packet, i.e. Date, Time, Node name, data (temperature, Pressure, Humidity and light intensity). To reduce the size of packet format we can also skip some or the other parameters. In above figure we can observe

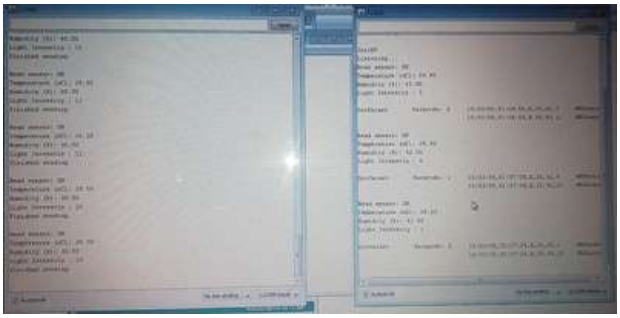

Fig. 9 Server and Client node Data Monitor

This same data can also monitor on cloud by an open source API for example "ThingSpeak". Here we create an account on Thingspeak API for monitoring real time data acquired by Ethernet module which is interfaced/ connected with server node. So the data which is send by client node and received by server node. Since server node interface with Ethernet node and Ethernet node connected with MODEM so the data from client node easily available on Thingspeak API in separate graphical format.

ThingSpeak is an open application platform designed to enable meaningful connections between things and people. ThingSpeak has an open source API to store and retrieve data from things using HTTP over the Internet or via a Local Area Network. With ThingSpeak, we can create sensor logging applications, location tracking applications, and a social network of things with status updates. The ThingSpeak API is available on GitHub for download and installation on your own servers. We can also take the source code and make changes and contribute new features. ThingSpeak is a modern Ruby on Rails 3.0 application and includes everything to get started including, a full web application, User Management, API Key Management, Channel Management, and Charting. The license for ThingSpeak is under GPLv3 for open source use and can be licensed from ioBridge for closed source applications. ThingSpeak has been installed on over 500 servers and licenced commercially since its release on GitHub in March 2011.

Few Result observed on Thingspeak API as shown below

VII. CONCLUSION

It has been conclude that we can design the WSN's for different application where the data transmitted by client node, received by server node and the same is made available on an open source API i.e. Thingspeak. The same data can also monitor by serial monitor by using COM port (Shown in aoe figure) and also store in SD card for future use. This shows that WSNs are fully capable of robust and reliable communication in the harsh environment found on industrial platforms. Although the use of WSN provides many benefits for industry and there are many various potential application areas for Industrial WSN, the problem of implementing reliable wireless communication in real-life industrial environment is still very complicated and requires further research. In future, it could be further expanded to communication between nodes placed in different places where we can also apply security and also work on different protocols.

REFERENCES

[1] E. Altman, T. Basar, T. Jimenez, and N. Shimkin, "Competitive routing in networks with polynomial costs," IEEE Trans. Automat. Control, vol. 47, no. 1, pp. 92-96, 2002.

[2] R. Bronson and G. Naadimuthu, Operations Research, 2 ed., Schaum's Outlines, McGraw Hill, New York, 1997.

[3] Gerard M. stegmaier ' The sentient economy: law and policy for the "internet of things"

[4] 8-bit Microcontroller with 16/32K Bytes of ISP Flash and USB Controller, Available inhttp://www.atmel.com/images/doc7766.pdf

[5] nRF24L01Single Chip 2.4GHz Transceiver product specification. Available in www.nordicsemi.com/ipn/.../ nRF24L01_Product_Specification_v2_0.pdf

[6] DHT11 Temperature and humidity Sensor available in www.datasheetspdf.com/PDF/DHT11/785590/1 MA, 2000.

[7] Pyroelectric infrared radial sensor datasheet. Available in www.micropik.com/PDF/D203B-e.pdf

[8] Schurgers, Curt and Mani Srivastava. "Energy Efficient Routing in Wireless Sensor Networks." MILCOM'01, Vienna, VA (October 28-31, 2001): 357-361, http://circuit.ucsd.edu/~curts/papers/MILCOM01.pdf.

[9] Slijepcevic, Sasa and MiodragPotkonjak. "Power Efficient Organization of Wireless Sensor Networks." IEEE International Conference on Communications (ICC'01), Helsinki, Finland, June 2001 (2001): 472-476

[10] Dong, Qunfeng."Maximizing System Lifetime in Wireless Sensor Networks".2005.

[11] Chen, Chao-Lieh, Kuan-Rong Lee, Jung-Hsing Wang, Yau-Hwang Kuo. "Energy-proportional routing for Lifetime Extension of Clustering-based Wireless Sensor Networks". International Journal of Pervasive Computing and Communications, 2007, Volume 3, Issue 3, pages 304 – 321.

[12] Kim, Joohwan, Xiaojun Lin, Ness B. Shroff, PrasunSinha. "Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks With Anycast". Proceedings of IEEE INFOCOM '08. February 21, 2011.

[13] Jiang, C.; Yuan, D.; Zhao, Y. Towards Clustering Algorithms in Wireless Sensor Networks— A Survey. In Proceedings of IEEE Wireless Communications and Networking Conference, Budapest, Hungary, 5–8 April 2009; pp. 1–6.

[14] V. Gupta, P. Udupi, and A. Poursohi, "Early lessons from building Sensor.Network: an open data exchange for the web of things," in Proceedings of PerCom 2010, Mar 2010.