Implementation of Wireless Sensor Network Communication Terminal based on RTOS

P. Hazarath[1], J.Subhashini[2]

[1,2]Department of ECE, SRM University, Kattankulathur, Chennai

Abstract: ***In this paper we are going to monitor the sensor values in real time by employing wireless sensor nodes. We use the ARM7 micro controller chip as core control module and design the overall structure of the system according to the characteristics of wireless sensor network communication. We design the hardware system structure and software of the communication terminal. Then we design the application which can be grouped for large data based on the real time operating system platform.***

Keywords: Wireless Sensor Network; Communication terminal; RT-Vx works; Embedded C; Sensor nodes

1. Introduction

The system proposed in this paper aims at monitoring the sensor values in real time by employing wireless sensor networks (WSNs). The values calculated by the embedded system are transmitted to a monitoring unit through an IEEE 802.15.4-based WSN. At the base unit, various sensor values can be monitored in real time. An experimental study was conducted for observing the relationship between the WSN performance and the spectral occupancy at the operating environment. This study demonstrated that the use of intelligent nodes, with local processing capability, is essential for this type of application. The embedded system was deployed on a workbench, and studies were conducted to analyze torque and system efficiency. Here, from pc data will be updated in sever[1].

With the integrating of sensor technology, MEMS, modern network, wireless communication and low power consumption embedded technology; it pushes the modern wireless sensor network production and development. It could collect and process perceived information in the network coverage, then sending to the users timely. Wireless sensor network have so many advantages in low power consumption, low cost and no restriction of space, so it has a very broad application prospect. Many countries academia and industry think highly of it in military defense, urban management, industry control, biological medical treatment, area by remote control etc fields, it's considered one of the top ten technologies to change the world in the 21st century.

Sensor network node can collect and transmit perceived information. In the sensor network, because the nodes are distributed intensively and need to deal with the large amounts of data, however the node itself is not suitable for it, so the design of communication terminal between the nodes has the extremely vital significance. We use the RTOS to it. It also has an application to group large data. We propose a design scheme about wireless sensor network communication terminal based on embedded technology this terminal is used to connect each sensor node and can guarantee information processing efficiency, timely and accuracy. Users can view, modify and control every node's information timely through a visual interface.

2. Hardware Implementation

In this paper, we use hardware components like ARM lpc2148 microcontroller, zigbee modules, gas sensor, temperature sensor, humidity sensor, ARM microcontroller which is 8 bit, Dual In Line Package with 40 pin, ADC 0809 with 8 channel multiplexer and Zigbee for its wide use in wireless networks.

2.1 Transmitter Section

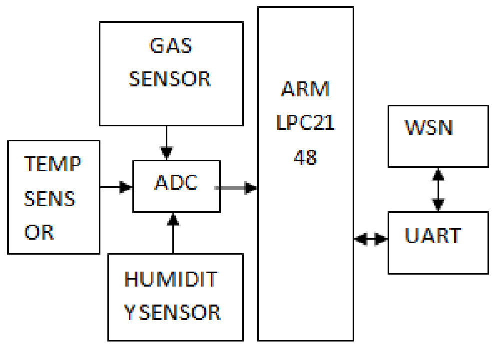

Figure 1: Transmitting unit

2.2 Receiver Section

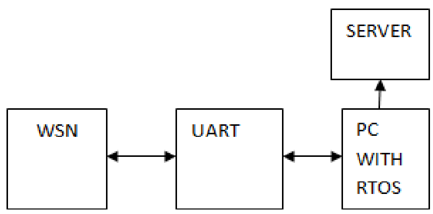

Figure 2: Receiving unit

ARM LPC2148: The LPC2148 is based on a 16/32 bit ARM7TDMI-S“CPU with real-time emulation and embedded trace support, together with 128/256 kilobytes (kB) of embedded high speed ßash memory. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at maximum clock rate. For critical code size applications, the alternative 16-bit Thumb¤ Mode reduces code by more than 30 % with minimal performance penalty. With their compact 64 pin package, low power consumption, various 32-bit timers, 4-channel 10-bit ADC, 2 advanced CAN channels, PWM channels and 46 GPIO lines with up to 9 external interrupt pins these microcontrollers are particularly suitable for automotive and industrial control applications as well as medical systems and fault-tolerant maintenance buses. With a wide range of additional serial communications interfaces, they are also suited for communication gateways and protocol converters as well as many other general-purpose applications [5].

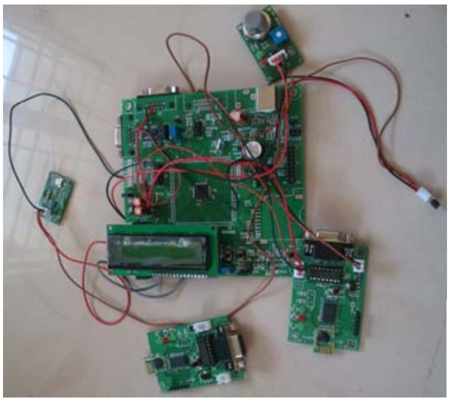

Figure 3: Working model

The above figure shows the working model of wireless sensor network communication terminal.

A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. UARTs are now commonly included in microcontrollers. A dual UART or DUART combines two UARTs into a single chip. Many modern ICs now come with a UART that can also communicate synchronously; these devices are called USARTs.

The Universal Asynchronous Receiver/Transmitter (UART) controller is the key component of the serial communications subsystem of a computer. The UART takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Serial transmission of digital information (bits) through a single wire or other medium is much more cost effective than parallel transmission through multiple wires. A UART is used to convert the transmitted information between its sequential and parallel form at each end of the link. Each UART contains a shift register which is the fundamental method of conversion between serial and parallel forms [8].

The MAX232 is an integrated circuit that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

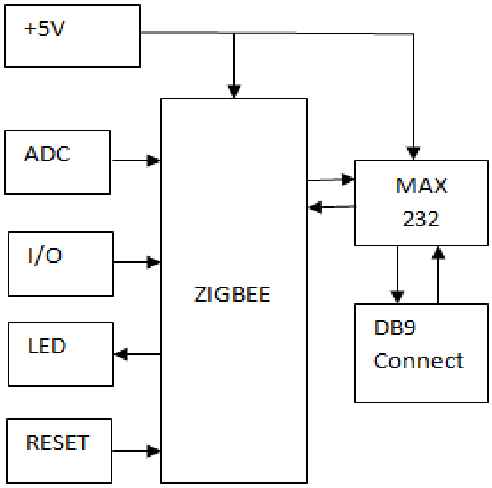

Figure 4:.Zigbee module

ZigBee network applications are rapidly spread out to the many areas: the home automation, industrial control, and commercial fields, for example. Presently, a great deal of literature is focused on studying the network development and management. However, the dynamic structure of ZigBee network is changeable and configurable and lead to the ZigBee network management to be difficult and complex. Furthermore, the system reliability and efficiency of ZigBee network will play the key role and technology to achieve the requirement and stability of system performance.

Gas detectors and gas sensors interact with a gas to initiate the measurement of its concentration. The gas detector of gas sensor then provides output to a gas instrument to display the measurements. Common gases measured by gas detectors and gas sensors include ammonia, aerosols, arsine, bromine, carbon dioxide, carbon monoxide, chlorine, chlorine dioxide, Diborane, dust, fluorine, germane, halocarbons or refrigerants, hydrocarbons, hydrogen, hydrogen chloride, hydrogen cyanide, hydrogen fluoride, hydrogen selenide, hydrogen sulfide, mercury vapor, nitrogen dioxide, nitrogen oxides, nitric oxide, organic solvents, oxygen, ozone, phosphate, silane, sulfur dioxide, and water vapor.

Important measurement specifications to consider when looking for gas detectors and gas sensors include the response time, the distance, and the flow rate. The response time is the amount of time required from the initial contact with the gas to the sensors processing of the signal. Distance is the maximum distance from the leak or gas source that the sensor can detect gases. The flow rate is the necessary flow rate of air or gas across the gas detectors or gas sensors to produce a signal [7].

The measurement of temperature is one of the fundamental requirements for environmental control, as well as certain chemical, electrical and mechanical controls. Many different types of temperature sensors are commercially available, and the type of temperature sensor that will be used in any particular application will depend on several factors. For example, cost, space constraints, durability, and accuracy of the temperature sensor are all considerations that typically need to be taken into account.

Some temperature sensors provide a wide range of temperature measurement, whereas other temperature sensors may only provide temperature information for a small temperature range. In addition to the temperature range sensed, the sensitivity and the accuracy of temperature sensors may also vary widely. Additionally, some temperature sensors work at high voltages while others only work at low voltages.

There're many types of devices that can be employed as temperature sensors. They include integrated circuits (ICs), pyrometers, resistance temperature detectors (RTDs), thermistors, thermocouples, electromechanical & volume (EMV).

LM35 is a precision IC temperature sensor with its output proportional to the temperature (in ^{o}C). The sensor circuitry is sealed and therefore it is not subjected to oxidation and other processes. With LM35, temperature can be measured more accurately than with a thermistor. It also possess low self heating and does not cause more than 0.1 ^{o}C temperature rise in still air[6].

In general, humidity sensors are used in home, office and industrial HVAC(heating, ventilating, and air conditioning) systems, meteorology station, museum and specific places where humidity plays a role crucially, such as green house, humidors, wine cellars, and hospitals. For example, for the patient with a respiratory illness or certain allergies, the humidity sensor is necessary because low humidity might cause breathing problem or joint pain; on the other hand, high humidity can result in growth of bacteria or fungus.

3. Implementation of Software

3.1 RTOS

A real-time operating system (RTOS) is an OS intended to serve real time application requests. It must be able to process data as it comes in, typically without buffering delays. Processing time requirements (including any OS delay) are measured in tenths of seconds or shorter.

Most operating systems appear to allow multiple programs to execute at the same time. This is called multi-tasking. In reality, each processor core can only be running a single thread of execution at any given point in time. A part of the operating system called the scheduler is responsible for deciding which program to run when, and provides the illusion of simultaneous execution by rapidly switching between each program [9].

The type of an operating system is defined by how the scheduler decides which program to run when. For example, the scheduler used in a multi user operating system (such as UNIX) will ensure each user gets a fair amount of the processing time. As another example, the scheduler in a desk top operating system (such as Windows) will try and ensure the computer remains responsive to its user. The scheduler in a Real Time Operating System (RTOS) is designed to provide a predictable execution pattern. This is particularly of interest to embedded systems as embedded systems often have real time requirements. A real time requirement is one that specifies that the embedded system must respond to a certain event within a strictly defined time (the *deadline*). A guarantee to meet real time requirements can only be made if the behavior of the operating system's scheduler can be predicted (and is therefore deterministic).

Traditional real time schedulers, such as the scheduler used in Free RTOS, achieve determinism by allowing the user to assign a priority to each thread of execution. The scheduler then uses the priority to know which thread of execution to run next. In Free RTOS, a thread of execution is called a *task*.

A key characteristic of an RTOS is the level of its consistency concerning the amount of time it takes to accept and complete an application's task. An RTOS that can usually or generally meet a deadline is a soft real-time OS, but if it can meet a deadline deterministically it is a hard real-time OS. An RTOS has an advanced algorithm for scheduling. Scheduler flexibility enables a wider, computer-system orchestration of process priorities, but a real-time OS is more frequently dedicated to a narrow set of applications. Key factors in a real-time OS are minimal interrupt latency and minimal thread switching latency; a real-time OS is valued more for how quickly or how predictably it can respond than for the amount of work it can perform in a given period of time.

3.2 Vx Works

VxWorks has been ported to a number of platforms and now runs on practically any modern CPU that is used in the embedded market. As is common in embedded system development, cross-compiling is used with VxWorks. Development is done on a "host" system where an integrated development environment (IDE), including the editor, compiler tool chain, debugger, and emulator can be used. Software is then compiled to run on the "target" system. This allows the developer to work with powerful development tools while targeting more limited hardware. The *Tornado* IDE was used for VxWorks 5.x and was replaced by the Eclipe based Workbench IDE for VxWorks 6.x. Workbench is also the IDE for the Wind River Linux and On-Chip Debugging product lines.

4. Results

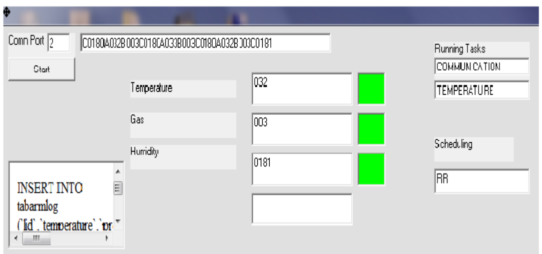

Figure 5: Sensed values using VX works

The above figure shows the output sensed values by using RT Vxworks and also we can observe that scheduling is Round Robbin. The right tops of corner in the above fig shows that communication tasks. The task which is getting high value that is the running task. In the above fig temperature is running task.

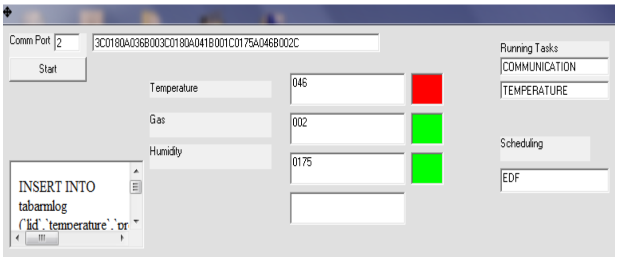

Figure 6: Indicating High Temperature

In the above figure red indicates at temperature that temperature vale goes high or crossed the limited temperature. Then EDF (Earliest Deadline First) scheduling has done here. The same procedure is followed for the remaining sensors.

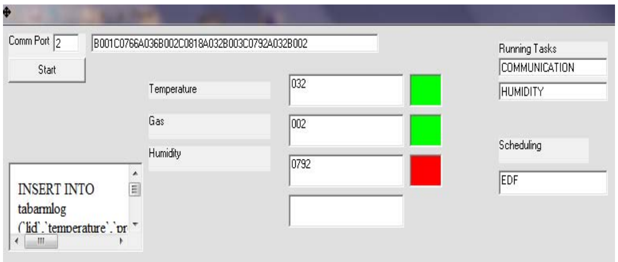

Figure 7: Indicating High Humidity

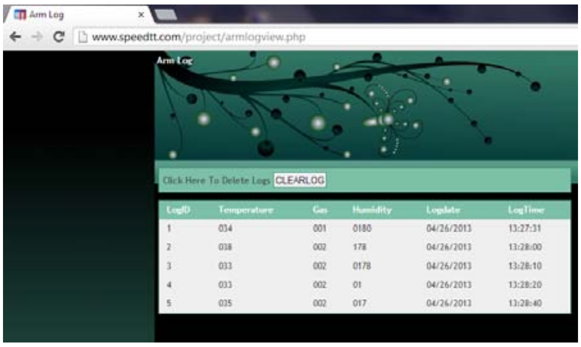

Figure 8: ARM values in internet

The above figure shows that ARM values by using internet. By using a server link we can access anywhere to monitor the values.

5. Conclusion

We adopt the low consumption and high performance ARM 7 as micro controller chip in this paper. Then we design the overall frame work of wireless sensor network communication terminal system and high performance communication terminal hardware structure. We also design the application to group the data according to the characteristics of communication terminal. This design reduces the wireless sensor node power consumption, improves processing ability of large quantities of data and strengthens the data transmission speed.

References

[1] Design of the Wireless Sensor Network Communication Terminal Based on Embedded Linux. Xiaohui Cheng, Fanfan Shen.College of Information Science and Engineering Guilin University of Technology Guilin, China.IEEE.

[2] Rohan Narayana Murty, Geoffrey Mainland etc. “CitySense An urban-scale wireless sensor network and testbed,” IEEE,2008.

[3] Geoffrey Werner-Allen, Konrad Lorincz etc. “Deploying a wireless sensor network on an active volcano,” IEEE,2006,18-25.

[4] Cui Zaihui, Li Tienan, Yang Feng, “Wireless automatic photographed terminal based on embedded Linux,”Computer Engineering, 2010.7, 213-215.

[5] ARM® lpc2148 microcontroller family user’s manual from www.arm.com.

[6] B. Zhang, “High Temperature Sensors based on Hydrogen Loaded

[7] Fiber Bragg Gratings,” Degree of Master of Applied Science, Dept.

[8] Elec. Comp. Eng., Concordia Univ., Montreal, QC, Canada.

[9] Dong Hyun Yun*, Chul Hun Kwon*, Hyung-Ki Hong*, Seung-Ryeol Kim*, Kyuchung Lee*Ho Geun Song** and Ji Eon Kim**

[10] *LG Corporate Institute of Technology, Devices and Materials Laboratory, MA Gr.16 Woomyeon-dong, Seocho-gu, Seoul 137-140, South Korea**LG Electro-Components Ltd., R & D Laboratory

[11] 379 Kasoo-dlong, Osan-city, Kyungki-do 447-1 50, South Korea

[12] A Robust UART Architecture Based on Recursive Running Sum Filter for Better Noise Performance, Himanshu Patel, Sanjay Trivedi, R. Neelkanthan, V. R. Gujraty

[13] MSDG/MRSA, Space Applications Centre (ISRO), Ahmedabad-380 015 .Gujarat. INDIA www.keil.com/rl-arm/rtx_rtosadv.asp