Wireless Sensor Network (WSN) Control for Indoor Temperature Monitoring

Yi-Jen Mon[1*], Chih-Min Lin[2], Imre J. Rudas[3]

[1*] Department of Computer Science and Information Engineering, Taoyuan Innovation Institute of Technology, Chung-Li, Taoyuan, 320, Taiwan, e-mail: monbuy@tiit.edu.tw

[2] Department of Electrical Engineering, Yuan Ze University, Chung-Li, Taoyuan, 320, Taiwan, e-mail: cml@saturn.yzu.edu.tw

[3] Óbuda University, Budapest, Hungary, e-mail: rudas@uni-obuda.hu

Abstract: In this paper, a wireless sensor network (WSN) is constructed to carry out certain applications. This WSN is composed of a sensor, monitor, controller, etc. It has the benefits of low cost and low power consumption. A WSN can be used in many applications in a range of different control technologies, such as temperature monitoring. ZigBee is used to test the performance of the WSN. The experimental results reveal that the design requirement can be achieved; they also demonstrate that the WSN control methodology allows good performance of data transfer using a liquid crystal display (LCD) and motor control.

Keywords: wireless sensor network; WSN; ZigBee; temperature monitoring

1 Introduction

The purpose of this paper is to construct a wireless sensor network (WSN), comprising a sensor, monitor, controller unit, etc. Taking advantage of low cost and low power consumption, a variety of perceived control networks can be concatenated into a sensor network and can achieve a variety of control techniques. In the past, applications have been developed in many areas, such as home security, environmental monitoring, home/building automation, indoor location identification, etc. These achievements make human life more comfortable and convenient [1].

The 89C51 chip produced by Atmel Corporation is a single-chip processor comprising a CPU, memory, I/O and other useful integrated peripheral interfaces. It is also known as a micro-processor or micro-controller unit (MCU) [2]. This type of

MCU is widely used in industry, such as in home consumer electrical applications and in industrial control products. The MCU was developed in response to the need for small, cheap and low power systems [3-5].

The use of assembly language involves a certain degree of complexity and difficulty when it is used to implement a number of features; meanwhile, it is also difficult to use in a cross-platform system, and the written code is difficult to understand. Using C language instead has the benefits of easy understanding and maintenance of programs. In this paper, the Keil u-Vision2 software platform [6] is used to compile the developed high-level C language. It will then automatically generate machine code which is easier and simpler to burn into the MCU's program memory using the Simple type-A PGMSx IC WRITER [7, 8].

The universal asynchronous receiver/transmitter (UART) includes a start bit, 8-bit data bits, parity bits, and stop bits. When the UART has received data or characters, the execution of serial to parallel conversions will be completed. Then, the UART will put the serial bits into a serial buffer (SBUF) to do parallel transmission, a process called parallel to serial conversion. The MCU can read all of the data transmitted by the UART by using a PC Super Terminal to set and display the communication results. The experimental results show that all simple ASCII code can be successfully sent and received [6-9].

The wireless sensor network (WSN) is envisaged to monitor the environment for many years. A challenge is to reduce the WSN's energy consumption so as to extend its lifetime [10]. The ZigBee Alliance is an association of companies working together to develop standards (and products) for reliable, cost-effective, and low-power wireless networking. The ZigBee technology will probably be embedded in a wide range of products and applications across consumer, commercial, industrial and government markets worldwide [11]. ZigBee builds upon the IEEE 802.15.4 standard, which defines the physical and MAC layers for low cost, low rate personal area networks. ZigBee defines the network layer specifications for star, tree and peer-to-peer network topologies and provides a framework for application programming in the application layer. Route discovery in ZigBee is based on the well-known Ad Hoc On Demand Distance Vector routing algorithm (AODV). When a node needs a route to a certain destination, it broadcasts a route request (RREQ) message that propagates through the network until it reaches the destination. As it travels in the network, a RREQ message accumulates (in one of its fields) a forward cost value that is the sum of the costs of all the links it has traversed. The cost of a link can be set to a constant value or be dynamically calculated based on a link quality estimation provided by the IEEE 802.15.4 interface [12]. Wireless sensor networks are an emerging technology based on the progress of electrical and mechanical engineering, as well as computer science, in the last decade [13]. Mobile Ad Hoc networks allow autonomy and independence from any fixed infrastructures or coordinating points. Considering topology changes due to the mobility of hosts, these last must self-organize to transfer data packets or any information with mobility and wireless physical characteristics management [14]. An Ad Hoc network is considered a very particular network, since it is a self-organizing network with no pre-deployed infrastructure and no centralized control; instead, nodes carry out basic networking functions such as routing. With this flexibility, Ad Hoc networks have the ability to be formed anywhere and at any time. In addition to traditional uses, such as for military battlefield applications, these networks are being increasingly used in everyday applications, such as in conferences, personal area networking and meetings [15]. Many routing protocols that are compatible with the characteristics of Ad Hoc networks have been proposed. In general, they can be divided into two main categories: topology-based and position-based. Topology-based routing protocols use information about links that exist in the network to perform packet forwarding. In general, topology-based routing protocols are considered not to scale in networks with more than several hundred nodes [16].

In this paper, the WSN is based on Ad Hoc structure, as aforementioned. The data transfer of liquid crystal display (LCD) and motor control are achieved by way of the MCU control methodology.

2 Introduction to WSN

The software development of the WSN is the most important issue. In this paper, the free software called Code::Blocks is used. This includes many application program interfaces (API). The Application Queue API provides a queue-based interface between an application and both the IEEE 802.15.4 stack and the hardware drivers (for the Jennic JN51xx wireless microcontroller):

• The API interacts with the IEEE 802.15.4 stack via the Jennic 802.15.4 Stack API (which sits on top of the 802.15.4 stack).

• The API interacts with the Peripheral Hardware Drivers via the Jennic Integrated Peripherals API (which sits on top of the Peripheral Hardware Drivers). This architecture is shown in Fig. 1. The Application Queue API handles interrupts coming from the MAC sub-layer of the IEEE 802.15.4 stack and from the integrated peripherals of the Jennic JN51xx wireless microcontroller, saving the application from dealing with interrupts directly.

The API implements a queue for each of three types of interrupt:

• Medium Access Control (MAC) Common Part Sub-layer (MCPS) interrupts coming from the stack. This is used for the MAC Data Services.

• MAC sub-Layer Management Entity (MLME) interrupts coming from the stack. This is used for the MAC Management Services.

• Hardware interrupts coming from the hardware drivers.

The prototype for the MCPS and MLME callbacks is a function that takes no parameters and returns void. The prototype for the hardware indications takes two 32-bit values as parameters and returns void. The application polls these queues for entries and then processes the entries [17-20].

A variety of network topologies are possible with IEEE 802.15.4. A network must consist of a minimum of two devices, of which one is the co-ordinator, referred to as the personal area network (PAN) co-ordinator. The possible network topologies are star topology, tree topology and mesh topology. The basic type of network topology is the star topology. A star topology consists of a central PAN co-ordinator surrounded by the other nodes of the network, often referred to as end devices. The tree network topology has an implicit structure based on parent-child relationships. Each node (except the PAN co-ordinator) has a parent. The node (including the PAN co-ordinator) may also (but not necessarily) have one or more children. Each node can communicate only with its parent and its children (if any). Any node which is a parent acts as a local co-ordinator for its children. In the mesh network topology, all devices can be identical (except that one must have the capability to act as the PAN co-ordinator) and are deployed in an ad hoc arrangement (with no particular network structure). Some (if not all) nodes can communicate directly. The nodes may not all be within range of each other, but a message can be passed from one node to another until it reaches its final destination.

A data transfer between network nodes can be unsolicited or the result of a request:

• When transferring data from a co-ordinator to a node, the node may not always be ready to receive data, since it may be in sleep mode for some of the time. In this case, responsibility may be given to the node to request data when it is able to receive. Therefore, the node polls the co-ordinator for data, and the co-ordinator then checks whether data is available and, if so, transmits a data frame. Acknowledgments may also be optionally implemented.

• When transferring data from a node to another node where reception is likely to be guaranteed (for example, from a node to a co-ordinator), it is usual to send a data frame directly (i.e., unsolicited). Again, acknowledgments may be optionally implemented. The data transfer methods are shown in Fig. 2 [17-20].

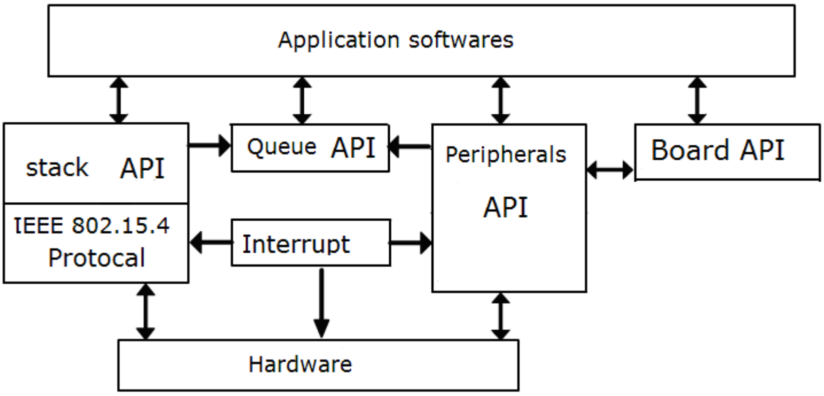

Figure 1
The architecture diagram of Jennic 802.15.4 API

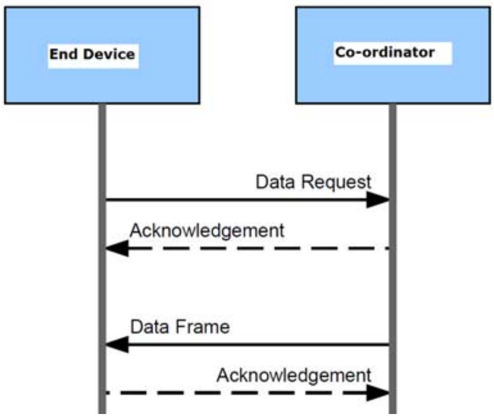

Figure 2
The diagram of data transfer methods of WSN

3 Experiment Result

The program is developed on the free software of Code::Blocks. First, the program for the co-ordinator and then the end device program are developed. Every network must have one and only one PAN co-ordinator, and one of the tasks in setting up a network is to select and initialize this co-ordinator. The network set-up process is shown in Fig. 3. The main co-ordinator and end device programs are shown in Fig. 4. The configure program diagram is shown in Fig. 5. The personal area network identify (PAN-ID) must be set adequately, such as in line 64 of this program. The development board is produced by Fontal Technology Inc. This is a high power ZigBee Kit (FT-6200). It can provide all the software tools and hardware required to get first-hand experience with wireless sensor networks (WSN). The entry-level kits contain one base development board (BDB) and one sensor development board (SDB). Each board is equipped with a high-power IEEE 802.15.4 RF module based on JN-5121 CPU (produced by Jennic Technology Inc.), which provides a much higher covering range, using a 2.4 GHz RF antenna that has an IPEX connector for easy mechanical design, rather than the normal power RF module. For I/O expansion ports, it has 10 useful pins of GPIO including UART, ADC, DAC and Comparator. The sensor development board features temperature and humidity sensors [12]. The development board is shown in Fig. 6.

For the software, Jennic Technology Inc. provides free Application Programming Interface (API) software for the peripheral devices on the JN5121 and JN513x single-chip IEEE 802.15.4 compliant wireless microcontrollers. This is known as the Integrated Peripherals API. It details the calls that may be made through the API in order to set up, control and respond to events generated by the peripheral blocks, such as UART, GPIO lines and timers, among others. The software invoked by this API is present in the on-chip ROM. This API does not include support for the Zigbee WSN MAC hardware built into the device; this hardware is controlled using the MAC software stack that is built into the on-chip ROM [17-20].

ZigBee can be used with different sensors, such as in home automation, security management, industrial or environmental controls, and personal medical care. The design concept diagram is shown in Fig. 7. Using UART, the data can be presented in the LCD in different sensors. First, the LCD test is implemented as in Fig. 8. Then, as temperature monitoring is the experiment's main purpose, the temperature sensor on the end device will transmit data to the co-ordinator and then also appear in the LCD through UART. The real implementation of temperature monitoring in the laboratory is shown in Fig. 9. This shows a measured temperature of 26 °C. If the temperature is higher than this, the motor should start up to drive a fan to lower the temperature. In this experiment, a light-emitting diode (LED) is used to identify the signal of the starting motor. The WSN's control of temperature monitoring is successfully established and good motor control performance is also demonstrated.

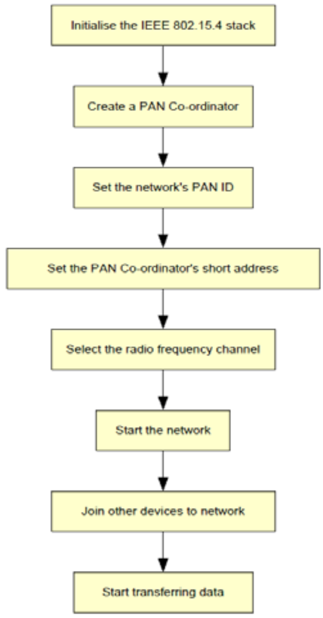

Figure 3
The diagram of network setup process

```
PUBLIC void AppColdStart(void)
{
   vWUART_Init();
   while(1)
   {
      vProcessEventQueues();
      switch (sCoordData.sSystem.eState)
      {
      case E_STATE_INIT:
         sCoordData.sSystem.u8Channel = CHANNEL_MIN;
         sCoordData.sSystem.eState =
E_STATE_START_ENERGY_SCAN;
         break;
      case E_STATE_START_ENERGY_SCAN:
         vStartEnergyScan();
         sCoordData.sSystem.eState =
E_STATE_ENERGY_SCANNING;
         break;
      case E_STATE_ENERGY_SCANNING:
         break;
      case E_STATE_START_COORDINATOR:
         vStartCoordinator();
         sCoordData.sSystem.eState =
E_STATE_RUNNING_UART_APP;
         break;
      case E_STATE_RUNNING_UART_APP:
         break;
      }
   }

}
```

Figure 4 (a)
The main program of co-ordinator

```
PUBLIC void AppColdStart(void)
{
   vWUART_Init();
   vStartActiveScan();
   while(1)
   {
      vProcessEventQueues();
   }
}
```

Figure 4 (b)
The main program of end device

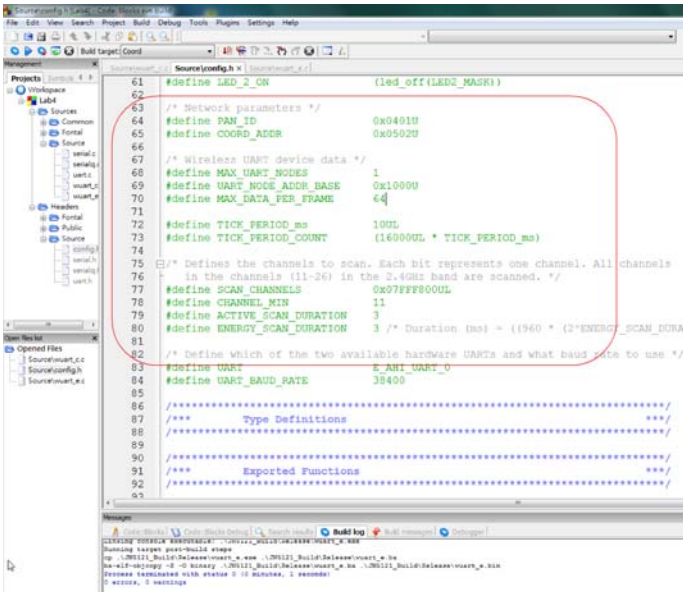

Figure 5

The configure program diagram

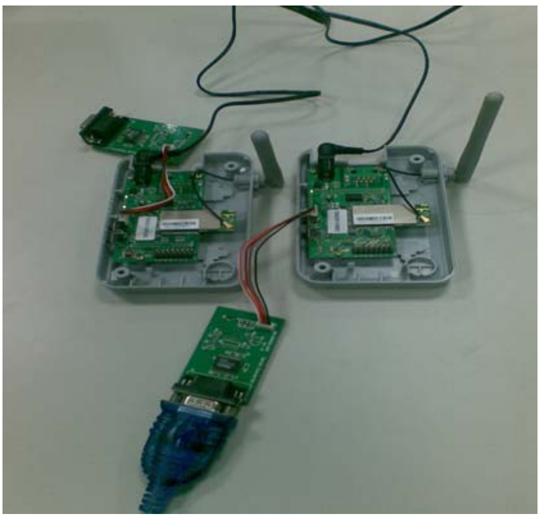

Figure 6

The development boards of WSN

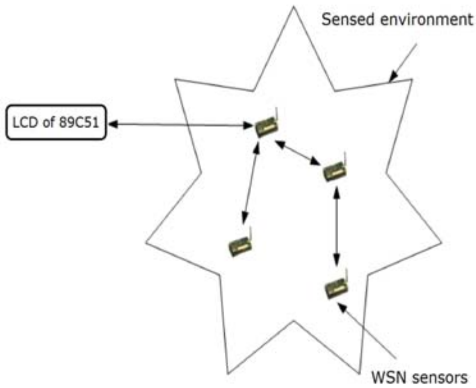

Figure 7
The concept diagram of WSN control

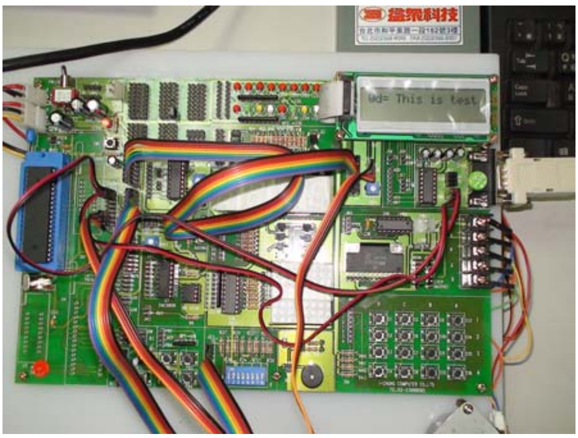

Figure 8
The diagram of LCD test

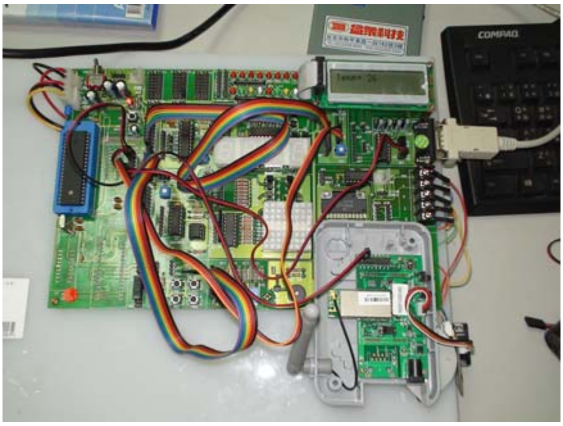

Figure 9
The implementation diagram of WSN for temperature monitor

Conclusions

In this paper, the design method for a temperature monitoring application using a wireless sensor network (WSN) is proposed. This paper has successfully demonstrated the application of the WSN to monitor the indoor temperature. The coordinator and end-device programs are developed using Code::Blocks software. The UART transmission and physical verification applications are also successfully demonstrated to possess good performance in data collection, temperature monitoring, motor control and display.

Acknowledgement

This paper is partially funded by teacher's research project of Taoyuan Innovation Institute of Technology.

References

[1] FT-6200 User Guide, 2012 (http://surewin.com.tw)

[2] Atmel 89C51 Microcontroller with 4Kbytes Flash Datasheet, Atmel Company, 2012 (http://www.datasheetcatalog.com/datasheets_pdf/8/9/C/5/89C51.shtml)

[3] A. Kalra and S. K. Kalra, Architecture and Programming of 8051 Microcontroller, Laxmi Pub. Ltd. 2010

[4] A. Sanz: A Complete Node for Power Line Communications in a Single Chip, International Symposium on Power Line Communications and Its Applications, 2005, pp. 285-289

[5] S. Y. Don: MCS-51 Practices and Designed by C language, I-Gung Publications, Ltd., Taiwan, 2008 (Trad. Chin. ver.)

[6] Simple type-A PGMSA (PGMSx IC WRITER) User Guide, I-Gung Publications, Ltd., Taiwan, 2008. (Trad. Chin. ver.)

[7] Keil u-Vision 2 Development Tools User Guide, ARM Inc., 2012. (http://www.keil.com/support/man_c51.htm)

[8] J. Kim, J. W. Choi and S. Lee: Universal I/O Design for Customizing MCU, Journal of Measurement Science and Instrumentation, Vol. 1, 2010, pp. 121-122

[9] W. Y. Chunga and S. J. Oh: Remote Monitoring System with Wireless Sensors Module for Room Environment, Senor and Actuator B, Vol. 13, 2006, pp. 64-70

[10] N. Golmie and I. Matta: Applications and Services in Wireless Networks, Computer Communications, Vol. 28, 2005, pp. 1603-1064

[11] I. J. Su, C. C. Tsai and W. T. Sung: Area Temperature System Monitoring and Computing Based on Adaptive Fuzzy Logic in Wireless Sensor Networks, Applied Soft Computing, Vol. 12, 2012, pp. 1532-1541

[12] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta and Y. F. Hu: Wireless Sensor Networks: A Survey on the State of the Art and the 802.15.4 and Zigbee Standars, Computer Communications, Vol. 30, 2007, pp. 1655-1695

[13] L. Aguilar, G. Licea and J. A. García-Macías: An Experimental Wireless Sensor Network Applied in Engineering Courses, Computer Applications in Engineering Education, Vol. 19, 2011, pp. 777-786

[14] R. Belbachir, Z. M. Mekkakia and A. Kies: Towards a New Approach in Available Bandwidth Measures on Mobile Ad Hoc Networks, Acta Polytechnica Hungarica, Vol. 8, 2011, pp. 133-148

[15] L. K. Qabajeh, M. L. M. Kiah and M. M. Qabajeh: Secure Unicast Position-based Routing Protocols for Ad-Hoc Networks, Acta Polytechnica Hungarica, Vol. 8, 2011, pp. 191-214

[16] M. L. M. Kiah, L. K. Qabajeh and M. M. Qabajeh: Unicast Position-based Routing Protocols for Ad-Hoc Networks, Acta Polytechnica Hungarica, Vol. 7, 2010, pp. 19-46

[17] Jennic Board API Reference Manual (JN-RM-2003), Jennic Inc., 2007

[18] Jennic 802.15.4 Stack API Reference Manual (JN-RM-2002), Jennic Inc., 2007

[19] Jennic Application Queue API Reference Manual (JN-RM-2025), Jennic Inc., 2006

[20] Jennic Integrated Peripherals API Reference Manual (JN-RM-2001), Jennic Inc., 2007