Research Article

Design and Implementation of a Cyber Physical System for Building Smart Living Spaces

Zhi-yong Bai and Xin-yuan Huang

School of Information Science and Technology, Beijing Forestry University, 35 Tsinghua East Road, Beijing 100083, China

Correspondence should be addressed to Zhi-yong Bai, baizhiyong@vip.qq.com

Received 14 December 2011; Accepted 15 March 2012

Academic Editor: Chih-Yung Chang

Copyright © 2012 Z.-y. Bai and X.-y. Huang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, the integrated control systems with computing, Internet, and the electronic equipment with embedded sensors or identification device, called Cyber Physical Systems (CPSs), have gradually attracted more and more attention in a variety of different areas. It is efficient to manage, control, monitor, and query on machine, equipment, and personnel state. Based on the CPS technology, creating smart living space becomes an important trend of future development. However, the electronic devices nowadays execute different communication protocols, such as Bluetooth, Zigbee, RF, and infrared, and even some traditional devices have no communication function. As a result, the real-time information and status of devices in smart spaces could not be effectively integrated, and thus will increase the difficulties to establish smart living space with CPSs. To improve this problem, this paper designs and implements an Intelligent Control Box to convert different wireless signals. The developed Intelligent Control Box can be treated as a multiple control platform which integrates the systems of lighting, air conditioning, access control, video surveillance, alarm, and so on, further decreasing the difficulties in establishing smart living space with CPSs.

1. Introduction

The Cyber Physical System (CPS) technology has been gradually matured and attracted more and more attention from all over the world. Its related applications have a huge impact in term of human life. In 2007 AD, the President’s Council of Advisors on Science and Technology reported eight key technologies which could enhance the modern life, and CPS technology was the primacy. Consequently, the development of CPS technology will be the main trend in the future.

Nowadays, many CPS applications have been proposed, which are smart space, healthcare, smart transportation, and so on. In smart space application, people can use smart phone or tablet to control the daily appliances via remote Internet access, hence increasing the quality of life. Healthcare application can help doctors observe the vital signs of the patients (or the elders) even if the patients (or the elders) stay at home instead of at hospital. In smart transportation application, sensor nodes can be embedded in vehicles to detect the nearby environmental information. For example, the accelerometer and GPS receiver can be embedded in a vehicle. When the accelerometer detects the pothole on the road, the vehicle can send the current coordinates which is obtained by the GPS receiver to the nearby vehicles and thus the traffic safety and efficiency can be improved.

The CPS mainly integrates computing with hardware control. Not only it can enhance the efficiency of system operation, but also it has the ability of monitoring and electronic devices control [1, 2]. Unlike the traditional embedded systems, the CPS is mainly designed for connecting physical devices to build an interaction network. The common way to build CPS is to embed sensors and actuators into electronic devices in daily life. The information of environment and electronic devices usage collected by sensors will be sent to the Decision Making System or the user by the existing Wireless Sensor Network (WSN) techniques, such as routing, data gathering, and MAC protocols [3–7]. Upon receiving the information, the Decision Making System or the user analyzes the collected information and then reflects the decision to the actuators by a sequence of control processes, controlling the electronic devices to perform the corresponding task.
The CPS is efficient to manage, control, monitor, and query on machinery, equipment, and personnel state and will be the important trend of future development to create more smart living spaces with CPSs technology. However, the electronic devices execute different communication protocols, such as Bluetooth, Zigbee, RF, and infrared. Indeed, some traditional devices have no communication function. As a result, the real-time information and status of devices in smart spaces could not be effectively integrated, thus will increase the difficulties in establishing smart living space with CPSs. To improve this problem, this paper designs and implements an Intelligent Control Box to convert different wireless signals. The developed Intelligent Control Box can be treated as a multiple control platform which integrates the systems of lighting, air conditioning, access control, video surveillance, alarm, and so on, further decreasing the difficulties in establishing smart living space with CPSs.

For example, as shown in Figure 1, if the temperature sensor detects that the room temperature is too high, it will send the “temperature too high” signals to the Decision Making System. However, the Decision Making System may only support Wi-Fi communication protocol. Hence, to successfully send the message to Decision Making System, the temperature signals have to be converted from Zigbee signals to Wi-Fi signals through the developed Intelligent Control Box. After that, the Decision Making System analyzes the information and then sends “turn on the air conditioner” via Wi-Fi communication protocol. Since most air conditioners available nowadays only support infrared technology, the signals must be converted from Wi-Fi signals to infrared signals through the Intelligent Control Box again. As a result, the air conditioner could successfully receive instructions and then start. This example shows that the developed Intelligent Control Box can efficiently integrate different devices even if they execute different communication protocols, further decreasing the difficulties in establishing smart living space with CPSs.

The remaining part of this paper is organized as follows. Section 2 reviews some of well-known applications of CPSs while Section 3 introduces the hardware platforms and firmware development environments used in the developed Intelligent Control Box. Section 4 details the key techniques in implementing Intelligent Control Box. Finally, the performance of Intelligent Control Box and conclusions of this paper are drawn in Sections 5 and 6, respectively.

2. Related Work

With development of wireless communication and embedded technology, as well as mature of short-range wireless communication technology, the CPS is primarily used in many domains, including healthcare [8, 9], emergent system [10, 11], transportation [12–15], and smart space [16–23].

In studies [8, 9], people equip with several wearable sensors to acquire the vital signs. Once the wearable sensor detects a specific event, it will send a message to the Decision Making System via Internet. Upon receiving the message, the system offers the corresponding service to the people. Studies [10, 11] developed the emergency system, which guided people to the safe areas when any dangerous event suddenly occurs. References [12–15] designed intelligent transportation systems which make people safer and more convenient.

The Distributed Robotic Garden [16] is a set of garden automated management system with CPS technology developed by MIT. In this system, each plant is equipped with both temperature and humidity sensors to monitor the current states of the garden plants. When a sensor detects the soil moisture supply is low, it sends the current plant location and related information to the Decision Making System. After that, the Decision Making System analyzes the received information and makes a watering decision to the robot in the garden.

The CPS can also be applied to the applications of energy conservation and smart living space. In energy conservation application [17–19], the proposed system can integrate the air conditioner, lighting, elevators, and other electrical equipment on each floor to achieve the energy conservation purpose. For example, when the temperature sensor detects that the floor temperature is too high, it sends this message to the Decision Making System. Upon receiving the message, the Decision Making System analyzes the received information and then makes the decision to “lower the temperature.” Contrarily, when the temperature is too low, it makes the decision to “increase the temperature.” Based on this system,
the energy efficiency can be dramatically improved, saving unnecessary power consumption.

In application of smart living space, reference [20] developed intelligent devices which are suitable for smart space application. In addition, many technology companies have raised home appliance control system for integration of a variety of common electronic products in the daily life, such as Medusa, Google, and Samsung. Among them, Medusa developed a home appliance control platform and used a peer-to-peer architecture to control the networked multimedia devices [21].

In the Google I/O 2011 conference, Google released Android@Home concept [22], which extended the Android from a mobile or tablet P5 device to the household appliances to build smart living space. Android@Home developed a signal converter box. When users intend to control appliances, they can send signals through the Android device to the signal converter box. Upon receiving the signal, the box then sends a corresponding signal to control the appliances.

South Korea’s Samsung Company had also launched an intelligent home control system, namely, Homevita [23]. This system mainly integrates Internet and home networks to monitor and control various home electronics devices using remote connection technology.

However, each electronic equipment used in [21–23] shall be developed by the manufacturing company. Otherwise, the real-time information and device state cannot be effectively integrated, and thus increase the difficulty to build smart living space.

According to the aforementioned applications, we can clearly find that the CPS technology plays an important role in creating smart living space. Nonetheless, how to effectively integrate the devices with different communication standards is a big challenge. To this end, this paper designs and implements an Intelligent Control Box to convert different wireless signals. The developed Intelligent Control Box can be treated as a multiple control platform which integrates the systems of lighting, air conditioner, access control, video surveillance, alarm, and so forth, further decreasing the difficulties in establishing smart living space with CPSs.

3. System Hardware and Software

This section firstly introduces the hardware platforms used in the developed Intelligent Control Box and their firmware development environments. Then a detailed description of the software development will be followed.

3.1. Hardware Platform. This subsection aims to introduce the hardware platforms used in Intelligent Control Box, including Octopus II sensor node, Arduino embedded controller board, and the Intel 8051 microcontroller.

3.1.1. Octopus II Sensor Node. Octopus II sensor node is jointly developed by the teams of Taiwan National Tsinghua University and Taiwan National Central University. It adopts a wireless communication module, CC2420 RF chip module, and the Zigbee protocol [24]. Compared with Tmote Sky sensor node produced by Moteiv Company, Octopus II sensor node is more powerful in terms of computing and data storage space. Furthermore, it can be burned directly with computer programs to facilitate program development and integration. Octopus II adopts dual-antenna design, and its transmission distance can reach 200 meters. Figure 2 shows the hardware system architecture of Octopus II sensor node.

The Octopus II hardware system architecture mainly consists of Power Module, Programming Module, Sensor Module, Processing Module, and Communication Module. Power Module is mainly responsible for the supply of power required for sensor nodes. The power resource of each node can be provided by two AA batteries; Programming Module is responsible for burning part of the program through the UART of computer interface. Sensor Module is responsible for the collection and analysis of environmental information and converts information from the analog signal to digital signal. Then, the digital information will be sent to the Processing Module. Processing Module is responsible for the operation and control of sensor nodes. This module contains the MP25P80 chip produced by ST Company and MSP430F1611 microprocessor, where MP25P80 chip is used to store program code. When the MSP430F1611 microprocessor is on the operation, the corresponding program code will be selected from the MP25P80 chip. The Communication Module which contains CC2420 chip is responsible for data collection and delivery via wireless transmission technology.

3.1.2. Arduino Embedded Controller Board. Arduino embedded controller board is jointly designed by Professors Dai Cuartielles and Massimo Banzi. It is a control panel based
on open source developed by the I/O interface with the language similar to Java or C. Arduiuno control panel supplies 14 Digital I/O and 6 Analog I/O and supports USB data transfer. Through the digital outputs, users can connect different electronic devices, such as LED lights, speakers, and motors, and then control them by Controller. In addition, it also can be integrated with Flash or Processing to make real human-computer interactive works. The following details the Arduino hardware system architecture.

As shown in Figure 3, Arduino hardware system architecture mainly consists of Power Module, Programming Module, Memory Module, and Processing Module. Power Module is responsible for the power supply, and the power source can be connected via USB interface or using the 5 V to 9 V DC power supply; Programming Module is responsible for the burning part of the program through the computer’s UART communication interface; Memory Module is responsible for managing the EEPROM, SRAM, and Flash’s memory usage to decrease excessive use of memory and avoid resulting in low work efficiency; Processing Module uses the octet ATMEGA168 Microcontroller and is responsible for data analysis and computing. It also controls the operations of each hardware component and the voltage output of each pin of Arduino control board.

3.1.3. Intel 8051 Single-Chip. Intel 8051 Microcomputer is created by Intel Corporation in 1981. It has several advantages, including small size, being easy to learn, and good scalability. Therefore, it is widely used in various fields.

As shown in Figure 4, Intel 8051 hardware system architecture primarily consists of Central Processing Unit, Memory Unit, and Input/Output Unit. Central Processing Unit is the core of control program, which contains two subunits: Arithmetic Logic Unit and Control Unit. When the Central Processing Unit receives the coming information from Memory Unit or Input/Output Unit, it uses the Arithmetic Logic Unit to perform arithmetic and logical operations, while the Control Unit is responsible for directing and coordination of data transfer between the modules and operation; Memory Unit is responsible for the data storage sent by Input/Output Unit and stores data processed by Central Processing Unit. In addition, Memory Unit, internal program memory (ROM) with 4 K storage space, is expandable up to 64 K, while the internal data memory (RAM) providing 128 Byte storage space is expandable to 64 K; Input/Output Unit is used to send the information to the external Central Processing Unit as an operation, and then the result of operations is outputted to an external device.

The software platform development environment will be further described in the next section.

3.2. Firmware Platform Development Environment. In this section, the firmware platform development environment of the Octopus II sensor node, Arduino embedded controller board, and the Intel 8051 microcontroller will be described. The firmware development environment Octopus II sensor node is TinyOS, and the compile environment of Arduino embedded control panel is provided by the development Arduino team. The development environment of the Intel 8051 microcontroller is Keil C51.

3.2.1. TinyOS. TinyOS is an embedded operating system [25] designed for wireless sensor nodes developed, and it uses nesC programming language to develop the firmware. In order to improve inherent hardware problems of the sensor node’s low memory capacity and MCU slow operation, low flash to burn many programs and limited battery power
limitations, TinyOS has many frequently used functions modular in advance, such as LED control program, RF transmission control program, and control program of the sensing element, so that users do not need to write too much code to save the sensor node within the limited memory space.

3.2.2. Arduino Firmware Compiler Environment. It has a large library and offers a variety of pin single-chip ATMega information for developers to use. Because the Arduino is an open source platform, developers can write their own libraries. When the error occurs after compiling, the error will be marked immediately to prompt developers. In addition, Arduino firmware build-in environment with the function of monitoring serial ports can monitor whether message packet is correct through this interface.

3.2.3. Keil C51. Keil C51 is 51 series single-chip firmware integrated development interface developed by Keil Software Corporation in the United States, with C programming language to develop firmware, and can be used to develop Intel 8051 microcontroller firmware in Windows development environment. Its software provides a rich library and powerful debugging tools, and code editor is also available to develop the firmware. It provides a number of single-chip drivers, and C language, assembly language of system components for developers. In addition, it also provides simulator to allow users to check their code. If the code has no error, users can burn the program to the 8051 microcontroller, dramatically saving much time. In the user interface part, it is similar with Microsoft Visual C++ interface. Even if users use the Keil C51 first time, they can easily create a variety of single-chip firmware.

4. The Software and Hardware Development of the Multiple Embedded Home Appliance Control Box

In this section, we will detail the development processes including two parts: hardware architecture and firmware programming. In hardware architecture design, we will use TinyOS, Arduino firmware compiler, and Keil C51 to develop the firmware part of Intelligent Control Box.

4.1. Hardware Architecture Design. In this subsection, the implement of hardware is divided into two phases, Control Box Implementation (CBI) Phase and Enhanced CBI (E-CBI) Phase. Intel 8051 microcontroller is used in CBI Phase, while Arduino embedded controller board and Octopus II sensor node are used to connect to the Internet in E-CBI Phase to make user control Intelligent Control Box via remote connection.

4.1.1. Control Box Implementation (CBI) Phase. Shown in Figure 5, the Intel 8051 microcontroller is the main part of the control box. Through UART communication interface, the Decision Making System sends commands. The control box receives and follows the instructions to switch the relay (Relay) of the switch to turn on circuit to make installation able to operate. However, the Intel 8051 microcontroller cannot suffer from the higher voltage and current. To cope with this problem, additional IC ULN2803 chip is added on the Intel 8051 microcontroller. Based on this chip, Intel 8051 microcontroller can successfully control the relay control panel and will not cause excessive current chip damage. In order to detect whether the circuit is turned on, we also add PC817 photo coupler on Intel8051 control panel. The implementation approach is to put two power supply circuits paralleled with PC817 photo coupler and use Intel 8051’s Port1 as a detection pin. The internal structure of PC817 photo coupler is an LED and a phototransistor. When the input LED receives a current through it, it will light and the light will turn output transistor on. The potential change is to detect whether the circuit is on.

The circuit inside Intel 8051 single-chip controller board is shown in Figure 6. It is through Printed Circuit Board (PCB) Layout, and welded with relays, resistors, diodes, and other types of IC after circuit washing out. In the chart, the part with the red border is set aside for connection with the outlet or double-cut switch; the part with the blue border is Relay and can open or close the corresponding switch according to the signals sent by Intel 8051; the part with the orange border is PC817 photo coupler and used to detect potential changes and prompt the user whether the circuit is in conduction; the part with the green border is RS232 communication interface to communicate with the computer and control; the part with the yellow border is the power input of the control panel and can be connected to 5V DC power supply; the part with the pink border is IC ULN2803 and used to withstand high-voltage and high-current to avoid burning of Intel 8051; the part with the purple border is Intel 8051 Clip to analyze signals.

4.1.2. Enhanced CBI (E-CBI) Phase. In the E-CBI Phase, we use the Arduino controller board, Octopus II sensor nodes, Intel 8051 single-chip control panel, WiFi module, Bluetooth module, and infrared modules, to make the Intelligent
Control Box have capabilities of wireless communication. We focus on this part of the hardware architecture described in detail.

Shown in Figure 7, ZX-BLUETOOTH Bluetooth chip and WiFly GSX 802.11b/g wireless networking chip are integrated into the Arduino control panel. With the control panel of the Intel 8051 single-chip integration, it can connect with a variety of heterogeneous networks. For the Bluetooth chip, it supplies a UART transport interface. The standard operating voltage is 5 V, so we put its UART serial communication interface (5 V, GND, TX, RX) linked on the Arduino control board. After the Arduino UART baud rate (Baud Rate) is adjusted to 9600 bps as Bluetooth chips, it can communicate with the Bluetooth chip. For WiFi chip, the serial communication interface is converted to the SPI through the single-channel high performance SC16IS750 UART chip, and it will enhance the transfer rate between the Arduino and the WiFly GSX.

To control electronic devices (such as television and air conditioning) using infrared communication, we also integrate infrared module to the Intelligence Control Box. Infrared transmitter module is integrated with Arduino control board, and Arduino controller board can control the appliances through the PWM output pulse infrared to infrared LED.

4.2. Firmware Development. The implementation of the design of Intelligence Control Box sets aside UART as a communication interface. A variety of devices can use UART and Intel 8051 control panel to connect and control it. If users control or operate them with remote networks, they must have a unified command format. The following is the detailed introduction, how firmware programming is designed to analyze the received packet and format in the Intel 8051 and Arduino and converted into commands to control the box.

4.2.1. Intel 8051 Control Board. In the Intel 8051 control panel, the Decision Making System can also send command packet with the UART interface. To achieve this purpose, a set of algorithms is designed for the program developed by Intel 8051 control panel to correctly parse out the command, and the well-known Longest Common Subsequence (LCS) algorithm is as reference. Its main purpose is to find the longest common subsequence. This algorithm is to find the longest length of the packet format of all set at the beginning in the same time, to facilitate the program to parse packets. We will focus on the detailed description of the system flow chart of the Intel 8051 control panel.

Intel 8051 controller board system operation flow is shown in Figure 8. When the Decision Making System sends packet (User Command) to the Intel 8051 controller board via RS232 communication interface, the packet will go through System Queue Module and then will be temporary stored through the Data Capture Module and Data Storage Module in the Queue way to avoid 8051 microcontroller receiving too many data packets, resulting in system deadlock. When the Intel 8051 microcontroller receives the control packet, it analyzes and parses the packet via Parsing Module in Command Decode Module. After the packets are
parsed and not recognized, information is sent back to computer by Debugging Module and requires the message packet to be resent. Otherwise, control commands are transmitted to Device Control Module by Sending Module for identification to confirm which circuit shall be open or closed.

In this paper, a Buffer is used to save the current command received from the UART and reads and stores the received order with Producer and Consumer algorithm. This algorithm is designed to record current read and stored buffer position with out and in pointers and performs very smooth under the multithreaded CPU. However, Intel 8051 is a single-threaded CPU. Hence, using this algorithm will produce deadlock. To address this problem, a Count pointer is added to record current buffer capacity and avoid deadlock generation. The real way is that each location can be used as a Buffer start position using offset and makes LCS algorithm into a virtual Circle and encountered over all correct buffer points. The program can correctly parse the packet.

The instructions format consists of two formats, the sending instruction and receiving instruction. In sending instruction design, all start byte instructions are 0xFF in transfer instruction format design. The second byte is set to determine whether to read the electrical state connected with the control box or write instructions to control power circuit breaker. The third byte is the second one’s extensions, primarily to record every detail action code. The fourth byte is action entrainment data needed by each instruction.

When each instruction is sent to the control box and inquires circuit state or current control circuit in receiving instruction design, the control box will return 10 bytes information to the Decision Making System by default. The state labeled method is that the start bit of the first and second bytes is fixed at 0xAAAA, and each followed byte represents the current state of each power supply circuit in use. There are two values 00 and 01 to represent whether power circuit is currently open or closed.

4.2.2. Arduino Control Board. The Arduino Integration Module System Chart is shown in Figure 9. After embedded wireless gateway receives the Control Packet through the wireless network for remote control, it analyzes packets and exchanges information according to users’ connection approach. The Packet Analysis Module will automatically switch the networks protocol. After completing the steps above, the data will pass through the Signal Converter Module and be converted into instructions for 8051 single-clip identification which are resent to the control box and temporarily stored in System Command Buffer to confirm whether the queue is full. If there is space available for data storage, which is sent to the System Queue Module, the packet instructions will be analyzed and decrypted through Command Decode Module and finally transferred to the Device Control Module to do allocation for the circuit switching action.

In the firmware writing of the Arduino integrated wireless module, we must write the WiFly GSX 802.11 b/g wireless networking chip’s control firmware for transferring the data to the Internet via WiFi. In order to develop conveniently, we use WiFly GSX 802.11 b/g wireless LAN chip modules specific for Arduino-WiFly Shield to develop. Firstly, we must initialize the SC16IS750 SPI-UART bridge chip on the module through the SPI. We set Baudrate on SPI, transmission format and EFR register, and the serial transfer mode of the TX and RX pin is FIFO. After these settings, WiFly GSX 802.11b/g wireless networking chip can receive instructions and transfer files through Arduino’s SPI native function.

In order to make Arduino integrated wireless module having the Zigbee 802.15.4 wireless communications capabilities, we write the serial communication control firmware of the Octopus II to make the Arduino to communicate with the Octopus II using UART. The communication must open the UART mode of MSP430F1611 and then resend control instructions. One way is to directly use the bottom HplMsp430Usart1C components for the UART1 control instructions writing. To use HplMsp430Usart1C to be implemented, you must first declare HplMsp430Usart interface, and then call UsartResource.request() command to request for using the UART mode. TinyOS returns UsartResource.granted() event after request, so UART1 command transmission is well done.

To use Msp430Uart1C components to operate UART1 operation mode, Resource and UartStream interface are being declared. The Resource is responsible for the resource control of UART1, such as resource requirement and resources release. UartStream sends a whole string, and we use an array to hold control instructions.

5. Performance Evaluation

In this section, the performance of the developed Intelligent Control Box is examined. We place Intelligent Control Box at the center position of two smart living spaces whose sizes are 15 m² and 30 m², respectively.

Let Ns denote the total number of signal conversion during the simulation process and let Nc denote the number of successful signal conversion during the simulation process. Let Rs represent the success rate of signal conversion and it can be defined as Rs = Ns/Nc. Figure 10 shows the success rate of signal conversion. As shown in Figure 10, it has
high success rates from Zigbee and Bluetooth signals to WiFi signal. The reason is that WiFi signals have the feature of nonline of sight. Similarly, when the WiFi and Zigbee signals are converted into Bluetooth signals, the result is that Bluetooth signals also have the feature of nonline of sight and also have a good signal conversion success rate. Therefore, the infrared signals are in line of sight. Whether WiFi, Zigbee or Bluetooth signal is converted to Infrared signal, the signal conversion success rate is relatively low.

In the CPS system, users can also send instructions through User Interface and directly control the various electronic devices. This examination is the comparison of Phase I with Phase II. Phase I is developed for the CBI Phase in the control box with the initial home appliances control in cable transmission, while Phase II is developed for the multiple embedded home appliance control box in this paper. Shown in Figure 11, Phase I uses cable transfers to control electronic equipment, so the success rates of the control electronics both in the environment of 15 m² and 30 m² are always maintained at 100%. The success rates of Phase II both in the environment of 15 m² and 30 m² fall along with the decreasing number of electronic devices. This is because the developed control box can only handle a user issued command at one time. The more users here for a time, the lower success rate of electronic equipment control. In addition, the quality of the signal will be weak as the distance increases. Therefore, the performance of Phase II will be lower with the increase of smart living space.

6. Conclusion

In this paper, an Intelligent Control Box is developed to efficiently reduce the difficulties in creating smart living space with CPS technology. The implementation of the control box can be divided into steps, including the CBI Phase and the E-CBI Phase. In CBI Phase, the Intel 8051 microcontroller is used to make a preliminary control box which does not have wireless technology ability to control electronic devices. As a result, the Decision Making System or users only can rely on cable transmission to send commands to the electronic devices. To overcome this problem, the E-CBI Phase is developed in this paper. In E-CBI Phase, with the use of Arduino Control Board, Intel 8051 single chip controller board, Octopus II sensor nodes, WiFi module, Bluetooth module, and infrared modules, the control box is given the abilities of integration with a variety of different communications technologies. Experimental results show that the Intelligent Control Box can convert different wireless signals and decrease the difficulties in establishing smart living space with CPS technology.

Acknowledgment

This work was supported in part by the Beijing Forestry University Young Scientist Fund, under Grant BLX2011020.

References

[1] E. A. Lee, “Cyber physical systems: design challenges,” in Proceedings of the 11th IEEE Symposium on Object-Oriented Real-Time Distributed Computing (ISORC ’08), pp. 363–369, May 2008.
[2] E. A. Lee, “Cyber-physical systems—are computing foundations adequate?” in Proceedings of the NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, October 2006.
[3] E. M. Saad, M. H. Awadalla, and R. R. Darwish, “Adaptive energy-aware gathering strategy for wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 5, no. 6, pp. 834–849, 2009.
[4] M. Y. S. Uddin, M. M. Akbar, and S. M. Masum, “Hierarchical numbering based addressing and stateless routing scheme for wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 5, no. 5, pp. 391–428, 2009.
[5] J. Lian, K. Naik, G. B. Agnew, L. Chen, and M. T. ¨Ozsu, “BBS: an energy efficient localized routing scheme for query processing in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2, no. 1, pp. 23–54, 2006.
[6] S. S. Kulkarni and M. Arumugam, “Infuse: a TDMA based data dissemination protocol for sensor networks,” International Journal of Distributed Sensor Networks, vol. 2, no. 1, pp. 55–78, 2006.
[7] M. L. Pham, D. Kim, S. E. Yoo, and Y. Doh, “Power aware chain routing protocol for data gathering in sensor networks,” International Journal of Distributed Sensor Networks, vol. 1, no. 2, pp. 253–267, 2005.
[8] Y. M. Huang, M. Y. Hsieh, H. C. Chao, S. H. Hung, and J. H. Park, “Pervasive, secure access to a hierarchical sensor-based
healthcare monitoring architecture in wireless heterogeneous networks,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 400–411, 2009.

[9] G. López, V. Custodio, and J. I. Moreno, “LOBIN: E-textile and wireless-sensor-network-based platform for healthcare monitoring in future hospital environments,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 6, pp. 1446–1458, 2010.

[10] C. Buragohain, D. Agrawal, and S. Suri, “Distributed navigation algorithm for sensor networks,” in *Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM ’06)*, Spain, April 2006.

[11] M. Li, Y. Liu, J. Wang, and Z. Yang, “Sensor network navigation without locations,” in *Proceedings of the IEEE International Conference on Computer Communications (IEEE INFOCOM ’09)*, Brazil, April 2009.

[12] P. Mohan, V. N. Padmanabhan, and R. Ramjee, “Nericell: rich monitoring of road and traffic conditions using mobile smartphones,” in *Proceedings of the ACM Conference on Embedded Network Sensor Systems (ACM SenSys ’08)*, November 2008.

[13] A. Thiagarajan, L. Ravindranath, K. LaCurts et al., “VTrack: accurate, energy-aware road traffic delay estimation using mobile phones,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys ’09)*, pp. 85–98, November 2009.

[14] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G. S. Ahn, and A. T. Campbell, “BikeNet: a mobile sensing system for cyclist experience mapping,” *ACM Transactions on Sensor Networks*, vol. 6, no. 1, article 6, 2009.

[15] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, “Cooperative transit tracking using smart-phones,” in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys ’10)*, pp. 85–98, November 2010.

[16] N. Correll, N. Arechiga, A. Bolger et al., “Building a distributed robot garden,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’09)*, pp. 1509–1516, October 2009.

[17] H. Wicaksono, S. Rogalski, and E. Kusnady, “Knowledge-based intelligent energy management using building automation system,” in *Proceedings of the 9th International Power and Energy Conference (IPEC ’10)*, pp. 1140–1145, October 2010.

[18] J. Byun and S. Park, “Development of a self-adapting intelligent system for building energy saving and context-aware smart services,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 90–98, 2011.

[19] D. M. Han and J. H. Lim, “Design and implementation of smart home energy management systems based on ZigBee,” *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1417–1425, 2010.

[20] I. Chun, J. Park, H. Lee, W. Kim, S. Park, and E. Lee, “An agent-based self-adaptation architecture for implementing smart devices in smart space,” *Telecommunication Systems*. In press.

[21] S. Wray, T. Glauser, and A. Hopper, “Networked multimedia: the Medusa environment,” *IEEE Multimedia*, vol. 1, no. 4, pp. 54–63, 1994.

[22] M. Karch, *Android Tablets Made Simple: For Motorola Xoom, Samsung Galaxy Tab, Asus, Toshiba, and Other Tablets*, Apress, New York, NY, USA, 2011.

[23] Y. Son, S. Ko, J. Jang, H. Lee, J. Jeon, and J. Kim, “Half-push/half-polling,” in *Proceedings of the ACM Conference on Pattern Languages of Programs (ACM PLoP ’09)*, August 2009.

[24] ZigBee—2006 Specification, ZigBee Document 064112, 2006.

[25] P. Levis and D. Gay, *TinyOS Programming*, Cambridge University Press, New York, NY, USA, 2009.