Journal of Computers and Applied Science Education Volume 4, Number 3, 2017

ISSN 2409-7284 (Online)

A ZigBee-Based 1-to-N Bluetooth Transmission Control Support Platform with TI CC2530 Chip and Windows Phone 8.1

Fu-Hsien Chen[1], Kune-Yao Chen[2], Chia-Hao Shih[3] and Sheng-Yuan Yang[4,*]
[1]Department of Electrical Engineering
[2]Department of Information Management
[3,4]Dept. of Information and Communication
St. John's University, TAIWAN
[*]ysy@mail.sju.edu.tw

Received March 2017; Revised July 2017

ABSTRACT. *This study breaks the constraint on the number of traditional Bluetooth connections, and develops a Windows Phone 8.1-based Bluetooth communication app supporting a multitude of end devices. A brand new transmission control platform is proposed for 1-to-N connection, as a result of a seamless integration between ZigBee and Bluetooth. This study is presented with the following features: (i) Windows Phone 8.1 is integrated with CC2530, (ii) ZigBee setup parameters can be configured in a smart phone software interface and relevant parameters can be browsed conveniently thereon, (iii) the information integration of data and interface is done by Bluetooth, and (iv) unlike in a computer cable connection, parameter settings are made easy in the presented ZigBee system.*
Keywords*: ZigBee, Bluetooth, Windows Phone, CC2530, BC417.*

1. Introduction

In a society with a tremendous amount of information, mobile phone has been the most indispensable 3C technology product. Since Motorola developed the first mobile phone, and IBM released the first smart phone Simon of the world in 1992 [1], people began to overturn the imagination of mobile phone, so that a mobile phone was no longer a pure means of communication. IDC statistics indicate that the global smart phone shipments exceeded 1.3 billion in 2014 [6], and is believed to go beyond that in the future. In addition, according to the statistics of market share of mobile operating systems by IDC in quarter 2 of 2016 [7], Android released by Google accounts and iOS of Apple account for the maximum 87.6% and 11.7%, respectively, while Microsoft, although with the highest market share of PC operating systems

[9], accounts for as low as 2.33%. Due to a huge business profit, mobile communication technology and network services have been developed at a rapid race, and the demand for a wide variety of mobile services running on various mobile operating systems is on the rise.

Bluetooth is known as one of the requisite wireless communication standard interfaces for current mobile equipments, especially for smart phones. As clearly indicated in a review, most of the common peripheral Bluetooth commodities on the market, including earphones, speakers, and so on, should take advantage of cross-platform design. In addition, these products are mostly developed for point to point purposes, meaning that only a small number of Bluetooth earphones can bridge two mobile phones at the same time. However, for the order service at a food plaza of department stores and the cargo communication service of logistics industry, there is a strong requirement that multiple end devices be supported and mediated simultaneously. Therefore, there is actually a commercial value in probing into how to break through the shortcoming of traditional Bluetooth. Moreover, the Bluetooth Low Energy V4.0 (BLE, or Bluetooth Smart) technology proposed by the Bluetooth Special Interest Group (SIG) in 2010 has already been mature, and system providers have released Bluetooth V4.0 related APIs in succession for app development. Taking the currently popular BLE wearable products as an example, Google released Android Wear [2] smart watch in 2014, Apple released Apple Watch [3] later in the same year, Microsoft with the highest market share of PC operating systems launched a wearable product Microsoft Band [8] in October 2014 for its systems after Windows Phone was released. It is obvious that there exists a non-negligible profit to develop and then commercialize Bluetooth-based products on today’s market, a major motivation behind this study.

In addition, there are a much smaller number of Windows Phone apps than Android and iOS apps, and a number of mobile device makers manufacture a comparable number of wearable products to Google and Apple. Taking Xiaomi Inc., China as an example, Mi bracelet was released at a destructive price, such that the market was stirred, while Mi bracelet only supports Android and iOS operating systems [5] rather than Windows Phone. In other words,

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

Windows Phone has been marginalized in today's competitive market, and does not appeal to a vast majority of mobile phone makers, although Microsoft made a great effort to inject new blood into Windows Phone. In recent times, Internet of Things (IoT) brings about the development of information and communication industry at a rapid pace. Typical IoT is basically ZigBee-based technology, and is designed to support a large number of network nodes, a clear advantage over of Bluetooth. In addition, Bluetooth SIG published V4.2 Bluetooth core, a combination with the IPv6 protocol, as the groundwork of IoT in December 2014 [4], so that a Bluetooth device can be remotely controlled via network as a way to break the constraints on the number of Bluetooth links. Moreover, for a remote control of a V4.2 Bluetooth device via network, a control platform must be developed according to the architecture of network services instead of Bluetooth architecture. Finally, after V4.2 is launched, relevant products are not expected to emerge in the market until the end of 2015. This is a major motivation behind this work, and it stands as a piece of pioneering work.

It is known that ZigBee supports a great number of network nodes, but unfortunately a smart phone is not equipped with a ZigBee interface. On the contrary, Bluetooth, as a standard interface of a smart phone, was developed with an inherent limitation on the number of links. More importantly, Bluetooth development environment has been already well developed. In this paper, a way is found to preserve the advantage, but to remove the disadvantage, of ZigBee and Bluetooth. In another aspect, ZigBee communication can be enabled directly on a smart phone as a consequence of a seamless integration between both technologies, and a ZigBee network topology can be built for multipoint connection implementation via a router, not in compliance with Bluetooth V4.2. In addition, ZigBee is completely free of a complicated mechanism and a high time cost for matching a Bluetooth device, and is as well characterized by the fast response and the security of Bluetooth communication. Finally, according to a deep as well as wide survey on smart phone operating systems, Windows Phone is found to be smaller in size than Android and iOS, and the issue of Bluetooth communication on Windows Phone is hardly addressed in literature, highlighting the significance of this study. To be brief, a Windows Phone

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

8.1 App is presented in this work, and a Windows Phone-based control platform is developed as well as a way to seamlessly integrate ZigBee and Bluetooth and to break the congenital limitation on traditional Bluetooth multipoint communication.

2. System Framework and Technologies Involved

2.1 Hardware Architecture for Bluetooth 1-to-N Transmission

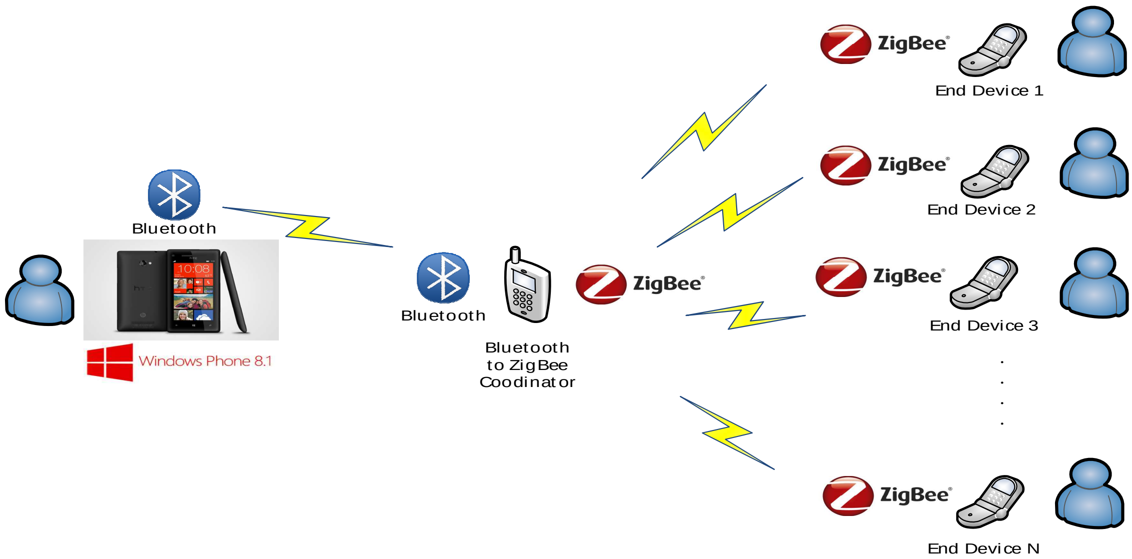

Figure 1 Framework of the presented Bluetooth 1-to-N transmission system

Demonstrated in Fig. 1 is a schematic diagram of the presented Bluetooth 1-to-N transmission system and the hardware architecture thereof. It mainly involves developing a Windows Phone 8.1 App for establishing a Bluetooth communication protocol and a transmission control, and involves normalizing a Bluetooth packet transfer for Serial Port Profile (SPP) Bluetooth setup file at a mobile terminal. The coordinator is developed using a CSR BC417 Bluetooth chip and a TI CC2530 chip, both linked by Universal Asynchronous Receiver/Transmitter (UART). BC417 converts a Bluetooth packet into UART protocol data, which are sent at 9600 Baud to CC2530, and CC2530 sends a feedback packet within the system via UART to BC417. The feedback packet is converted into a Bluetooth packet and then sent to the Windows Phone 8.1 terminal, so Bluetooth multitasks transmission jobs. CC2592 is a 2.4GHz RF preamplifier, amplifying the signals received from CC2530, with a power gain of

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

+22 dBm at the maximum output power, leading to an extended communication range of a ZigBee system [10]. Each ZigBee end device is equipped with a CC2530 chip. CC2530 takes control of the core of an end device, including the reception and processing of ZigBee packets. The parameter settings stored in the internal Flash can be viewed after carrying out planned work, such as taking a call, ZigBee parameter setup, LED flicker, power on/off of a buzzer and a vibration motor. The control signals of these peripherals are exported from the built-in general purpose I/O (GPIO) of CC2530. When an end device is called, the user can be notified by these modules, and the 1-to-N Bluetooth transmission controller is implemented based on ZigBee technology accordingly.

2.2 Software Architecture for Bluetooth 1-to-N Transmission

A coordinator software operation system involves data processing, memory access, I/O control and a packet processing composed of Bluetooth packet reception/transmission and ZigBee packet transfer, as illustrated in Fig. 2. In terms of data processing, CC2530 receives a Bluetooth packet, following which the packet content is analyzed, converted and sent to the system for the purposes of interface control and access to ZigBee parameters stored in Flash, etc. If a specific packet is about to be forwarded to a ZigBee end device, the coordinator encapsulates the required data, handed to a ZigBee packet sending program and waiting for sending. Besides, in a coordinator hardware architecture, as illustrated in Fig. 1, the program is designed to provide a CC2592 processing function for transmitted signals amplified by a CC2592 RF preamplifier to which relevant I/Os and circuits are connected.

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

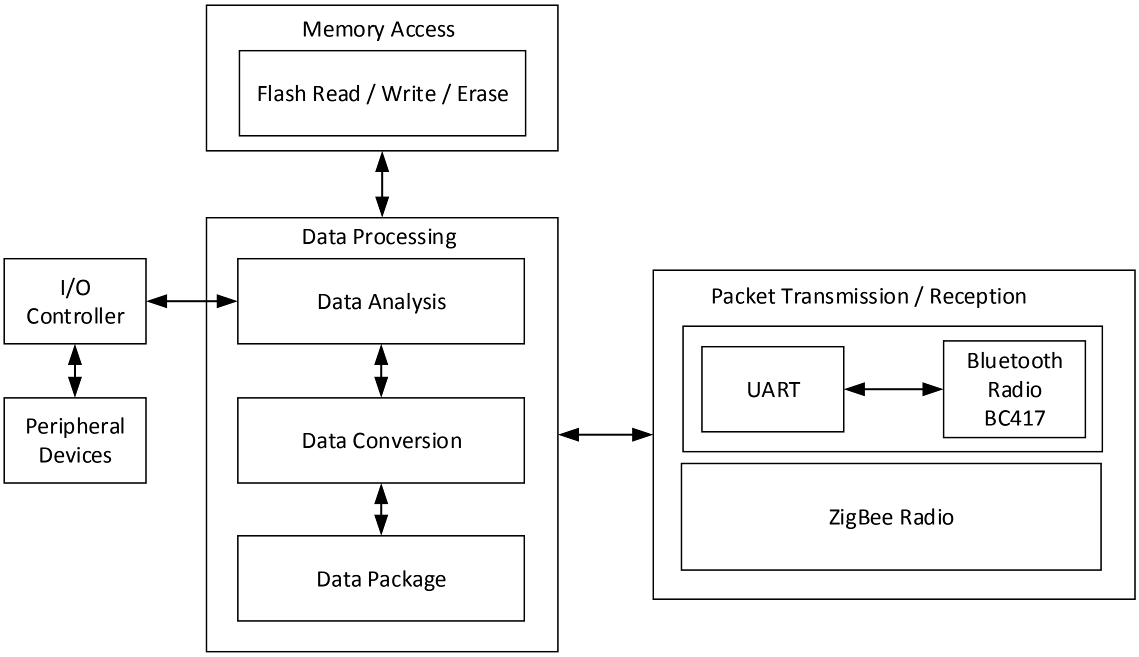

Figure 2 Software architecture of the presented coordinator

Illustrated in Fig. 3 is the software architecture of an end device, analogous to the coordinator one in Fig. 2, but a UART module is included for end device parameter configuration and as a system debugging interface in the future. For comparison purposes, each end device is equipped with a Bluetooth module, and Windows Phone 8.1 supports Bluetooth for either point to point parameter settings or transmission control. In addition, each end device is powered by a lithium ion polymer battery, and a built-in 12-Bit analog digital converter (ADC) I/O is provided by CC2530, by means of which the battery voltage can be detected. The moment the battery voltage goes below a threshold; the user is alerted to charge the end device so as to avoid disoperation.

2.3 System Communication Protocol

Fig. 4 illustrates a Bluetooth communication protocol between Windows Phone 8.1 and the coordinator, and a ZigBee communication protocol between the coordinator and end devices, including communication process, packet content and various forwarding mechanisms. When Windows Phone 8.1 has a Bluetooth packet available for sending, the system calculates the packet length for sending, meaning that the same length is not shared by all the packets.

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

Therefore, the packet length is calculated as a packet checking mechanism, say, the length of 1 byte is 1. The overall packet length is calculated, and put in the first field of a packet, waiting for the packet sending notification. In addition, Bluetooth sends the packet content in turn, and the ACK Monitor is enabled, which is a move different from Packet Reception, a subsystem receiving packets. When a packet is sent out, the ACK Monitor checks whether a feedback packet is received from the coordinator within a specified time span. If Packet Reception does not receive any packet in a timely manner, the ACK Monitor notifies the system to resend the packet, so as to guarantee the accuracy of the overall system operation. If Packet Reception does as expected receive a packet opportunely, the Data Processing processes the packet content. The system updates a user interface each time after processing, and notifies stopping the ACK Monitor, meaning that a feedback packet has been processed. The system will not send the packet again, so as to guarantee the integrity of the overall system operation.

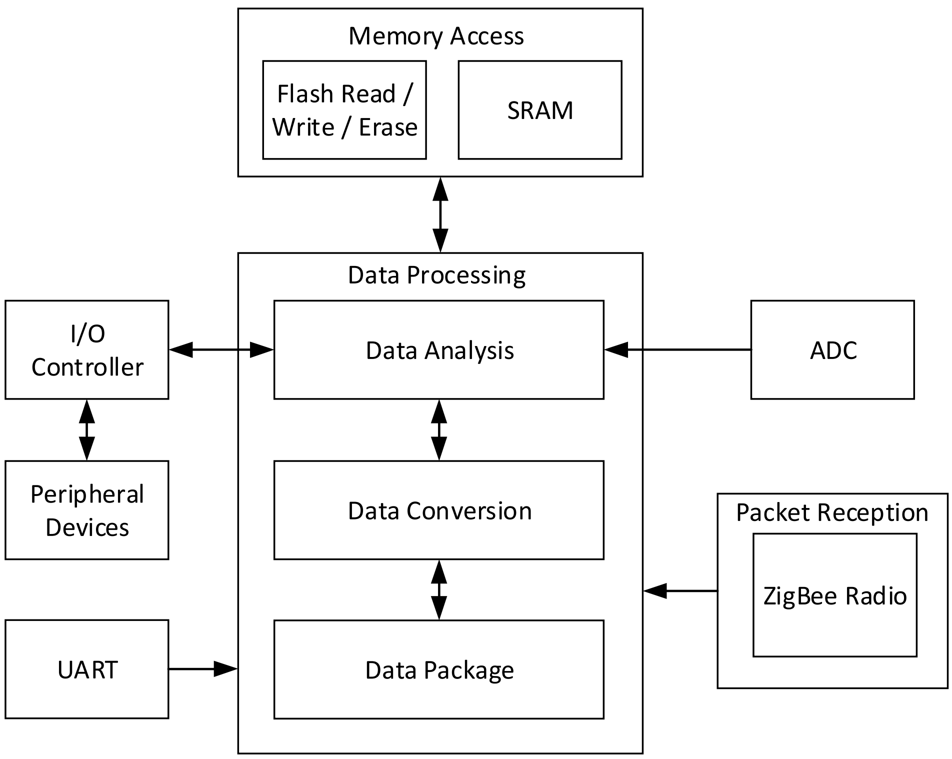

Figure 3 Software architecture of an end device

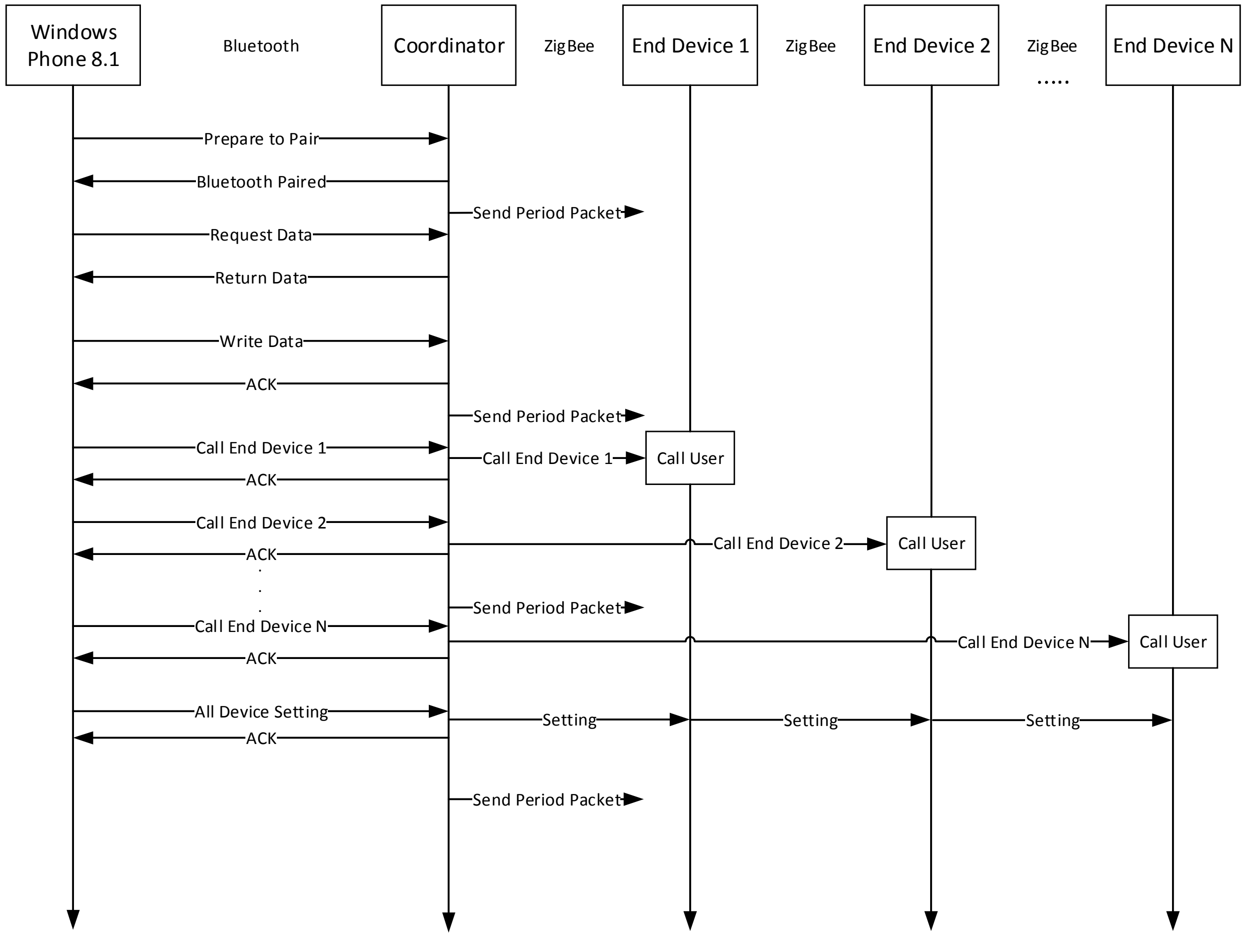

Figure 4 System transmission process

Finally, when the ACK Monitor is in operation, in order to avoid packet loss, the system sends the packet again at regular intervals. If a feedback packet is not received after a packet is sent many times, the ACK Monitor stops sending the packet due to time out, and notifies the system to update the user interface, so that the user is aware that the coordinator has not been connected to Bluetooth correctly, so as to perfect the processing autonomy of this system. However, when Windows Phone 8.1 receives a packet, the system reads the content of the first packet field as the content length of the packet. The system stores the packet content in Buffer, and calculates the actual packet length. When the actual length of a packet content equals the estimated packet length, the Buffer reads the whole packet content at a time, and the packet content is processed by a data handler.

3. System Presentation and Experimental Validation

3.1 System Hardware

A TI's SmartRF05 Evaluation Board is employed in the hardware development of a coordinator. As the hard core, CC2530-CC2591EM gets connected to peripheral I/O devices via the I/O Connector. A CSR BC417 Bluetooth module gets connected externally as a jig-tool for parameter settings of the coordinator or an end device. A finished coordinator is configured as represented in Fig. 5, involving a mobile power supply, left, a jig-tool for a Bluetooth device, center, and a coordinator, right. The USB Port of the coordinator can be used as a power input, and is able to connect a Bluetooth module or a jig-tool. Hence, the coordinator USB Port is connected to a Bluetooth module, Windows Phone 8.1 can be connected to a Bluetooth device by means of Bluetooth communication, and can communicate with CC2530 as well, so as to handle a one to many call or internal setting of the coordinator.

For making a call to end devices, a call bell system is developed using the aforesaid SmartRF05 Evaluation Board and CC2530EM. However, since the I/O Connector does not get involved in Bluetooth communication, only ZigBee is used for the communication to the coordinator.

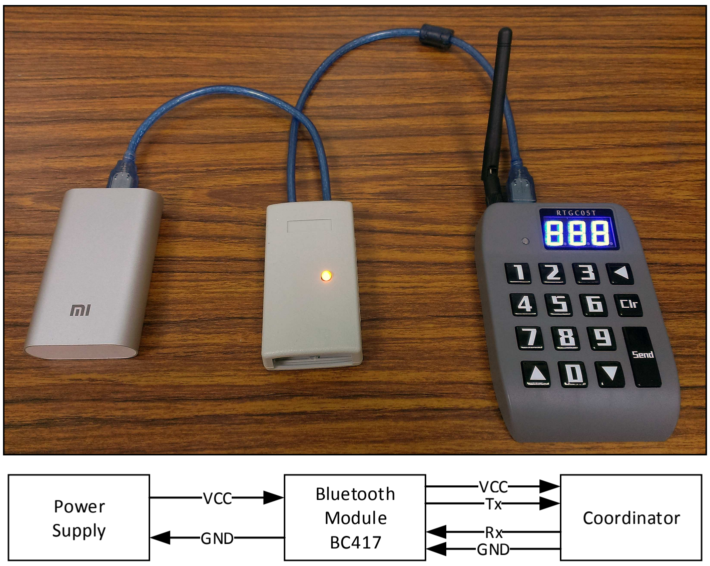

Figure 5 Final form of the presented coordinator and hardware configuration

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

As a way to avoid any missed call for end device users, the device is equipped with a vibration motor, a flickering LED and a buzzer, all connected to CC2530, and the call bell system is designed and then implemented. When an end device receives a packet from the ZigBee coordinator, specific jobs are performed according to the internal settings of the device, which will be described later. Demonstrated in Fig. 6 is the final form of a set of end devices.

Figure 6 Final form of end devices

3.2 Windows Phone 8.1 Software

There are three sets of software programs developed for Windows Phone 8.1 software, which are call bell system, coordinator setting and end device setting software programs. As illustrated in Fig. 7, a real keyboard and a seven-segment display in the original coordinator can be completely replaced using the user calling software, as long as the coordinator is configured as demonstrated in Fig. 5. Accordingly, the user calling software can communicate with CC2530 directly via Bluetooth. To call an end device number, all it takes is to type a phone number directly on the touch screen, and then presses the Send button. The coordinator can be controlled by a mobile phone directly after a connection is made, as long as it is covered within the communication range. Using a forwarded ZigBee packet to call an arbitrary end device, the aim of 1-to-N Bluetooth transmission control platform development is reached.

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

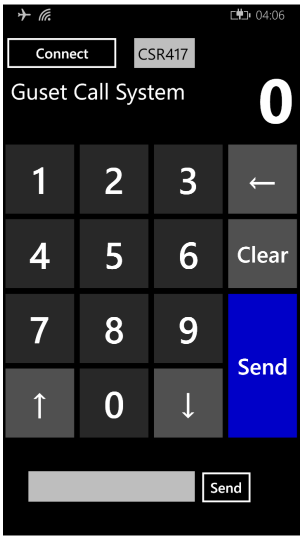

Figure 7 Initial screen of the Windows Phone 8.1 call bell system software

Table 1 Instructions on end device operations issued by the coordinator wirelessly

Data	**Function**
801	Call all the receivers to shut down.
805	Enforcedly call all the receivers to look for a receiver.
806	Call receiver 1 at regular intervals of 10 seconds to test transmission range.
807	Turn off the buzzer ringing calling function.
808	Turn on the buzzer ringing calling function.
809	Turn off off-limits ring (turn off buzzer only).
810	Turn on off-limits ring (turn on buzzer only).

This system is designed able to call an arbitrary end device, and a broadcast mechanism is developed to set the functions of an end device. In other words, an end device must be covered within the coordinator communication range, and the channel of an end device is the same as the PAN ID of the coordinator. Once the codes listed in Table 1 are imported into a set of Windows Phone 8.1 calling software programs by the coordinator, the functions installed in an end device can be controlled at the same time by a ZigBee broadcast, so that an end device is set up at a time. In this way, the ZigBee-based 1-to-N Bluetooth transmission control platform, providing function settings of an end device wirelessly, can be completed easily by a mobile phone.

As referred to previously, the ZigBee system is designed able to set relevant parameters in such a way that an arbitrary device can communicate with all the others. As a way to fulfill such

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

requirement, a set of coordinator setting software programs, analogous to a background mechanism, is developed.

Figure 8 Setting menus of the Windows Phone 8.1 coordinator setting software

The coordinator is configured as represented in Fig. 5, and a click on the "Connect" button has the phone connected to a Bluetooth device. Once connected, the button displays "Connected". If Bluetooth is detected as off, a pop-up message is displayed to remind the user to turn on the Bluetooth communication, after which the Bluetooth setup menu of a Windows Phone is displayed. The moment the Bluetooth communication is turned on, the Bluetooth device is connected, and "Read Data" is clicked to read the internal parameters of the coordinator immediately, including type, number, version, channel, PAN ID, ringing, ringing time, buzzer stop, vibration switch, LED flasher and out of range alarm. The packet content is analyzed right after data reading, the interface is updated immediately, and “Read Data Completed” is displayed on the red status box at the bottom of the menu. As illustrated in Fig. 8, “Device No Response” is displayed, in response to a communication failure with CC2530 after the connection. Once the data are read successfully, the UI enables "Write Data" and "Write SN" functions, direct clicks on the gray numerical boxes and the switches underneath to modify or to change relevant parameters according to the user’s requirement. In case of an input

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

parameter out of range, the system corrects the parameter value automatically or reminds the user to re-input the parameter. When the parameter modification is completed, the modified parameter content can be sent to CC2530 by a click on "Write Data". Table 2 gives the respective parameter range of the coordinator.

Table 2 Coordinator parameter setting range

Channel	**PAN ID**	**Alarm Type**	**Alarm Time**	
11 - 26	01 - 99	1 - 7	6 - 54	
Beep	**Vibrate**	**Blink**	**Out of Range Alarm**	**Serial Number**
ON / OFF	ON / OFF	ON / OFF	ON / OFF	000000 - 999999

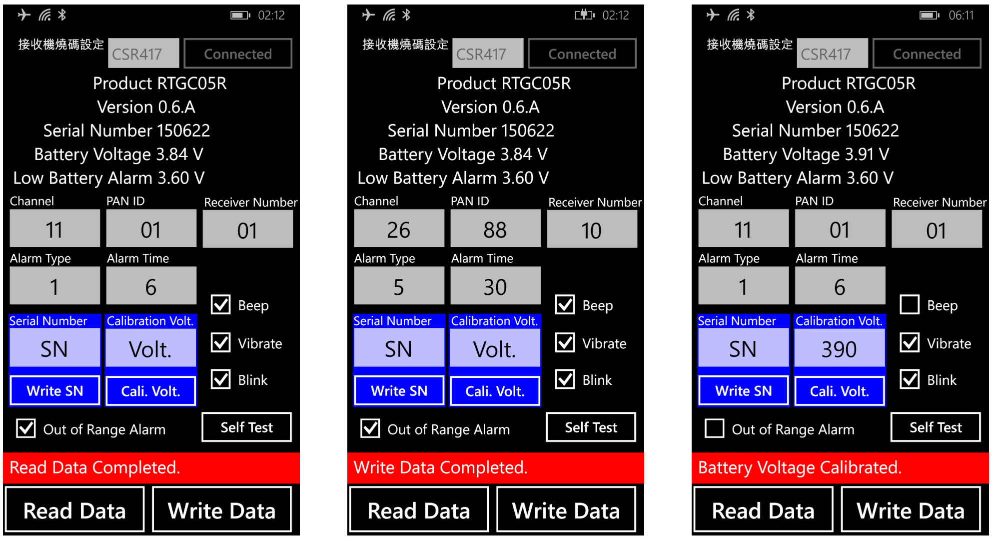

Figure 9 Setting menus of end device setting software

Using the interface illustrated in Fig. 9, the parameters of an end device can be set through broadcast by an input code issued by the coordinator. Receiver Number is another parameter in addition to the common internal ones of the coordinator, and each end device is allotted a receiver number. Once a calling packet is received, and the packet number matches its number, a call is made instantly to an end device on a condition that the corresponding receiver number matches the packet number. These parameter settings can be read by a mobile phone, and written in the end device. The battery voltage of an end device can be detected and represented as the “Battery Voltage” in the interface, with a default voltage of 3.6V, to see whether the end device has adequate power or not. The function Calibration Voltage is designed to calibrate the

 Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

voltage detected. A click on the "Cali. Volt." button enables the phone to send a reference voltage to the end device for the update of the battery parameters, and the calibrated voltage is fed back and updated in the mobile phone. As illustrated in Fig. 9, "Battery Voltage Calibrated" is displayed in the red status column after the battery calibration. Table 3 gives parameter setting range for an end device.

Table 3 End device parameter settings

Channel	**PAN ID**	**Alarm Type**	**Alarm Time**	**Receiver Number**	**Beep**
11 - 26	01 - 99	1 - 7	6 - 54	01 - 99	ON / OFF

Vibration	**Blink**	**Out of Range Alarm**	**Serial Number**	**Calibration Voltage**
ON / OFF	ON / OFF	ON / OFF	000000 - 999999	Variable by Battery

3.3 System Transmission Authentication

The corridor area on the 6th floor of the Business Administration Building of St. John's University is chosen as a space for field test. It is a "T" shaped and not a fully open space. The test is made with 2.4GHz signals at ultra high frequency (UHF) band. As illustrated in Fig. 10, Nodes A, B and D are deployed at the very corners of the building, while Node C is within the visual range of other nodes. A Bluetooth point-to-point test is made by Windows Phone 8.1 together with end devices via Bluetooth communication, and the test result is listed in Table 4. As can be found therein, Node D fails to receive a packet sent from Nodes A and B, as 2.4GHz UHF signals are unlikely to penetrate through obstacles, i.e. a signal transmission between two nodes along a nonlinear path or beyond a visual range is likely to be interrupted. In contrast, a successful transmission is demonstrated in the rest of test cases.

Subsequently, a test is conducted on the effective communication range as follows. The coordinator stays at Node C within the visual range of other nodes. Windows Phone and an end device can move freely to arbitrary nodes. As listed in Table 5, the coordinator is found able to relay a Bluetooth signal to a ZigBee system, meaning that a successful packet reception and transmission is achieved in any test case and the communication range can be well extended as expected.

Figure 10 A full view of the corridor on the 6th floor of the Business Administration Building, St. John's University

Table 4 Bluetooth point-to-point test result

Master (Send) Windows Phone	Slave (Receive) Bluetooth End Device	Result
Node A	Node B	Success
Node A	Node C	Success
Node A	Node D	Fail
Node B	Node C	Success
Node B	Node D	Fail
Node C	Node D	Success

Table 5 Bluetooth-ZigBee hybrid network topology test result

Windows Phone	Coordinator	End Device	Result
Node A	Node C	Node B	Success
Node A	Node C	Node D	Success
Node B	Node C	Node A	Success
Node B	Node C	Node D	Success
Node D	Node C	Node A	Success
Node D	Node C	Node B	Success

4. Conclusion and Discussion

The proposed system architecture is designed and then realized as an innovative and unprecedented piece of work, involving a composite transmission topology, a transmission protocol, packet format and processing programs developed on Windows Phone 8.1. The features are listed below:

(1) A Bluetooth packet is converted into a ZigBee signal using the composite hybrid topology, as a way to extend the Bluetooth communication range.

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

(2) The time delay due to prior matching required in traditional Bluetooth can be eliminated by a ZigBee communication control, all the end devices of the same system can receive signals on a condition that they are lie within the communication range of the coordinator, and the constraints on the quantity of the connected Bluetooth end devices can be broken.

(3) Windows Phone is employed to control a great number of end devices seamlessly simply through a one-time match with a Bluetooth device, and there is little transmission delay time consequently.

(4) For illustration purposes, the presented one-to-many Bluetooth communication mechanism is implemented in a version above Windows Phone 8, but actually it is developed as a cross-platform technology and hence supports other operating systems.

(5) The smart phone is not equipped with a ZigBee communication interface, while instead it can communicate with a ZigBee device in this architecture.

It is experimentally validated that the communication between Windows Phone and CC2530, between ZigBee devices and a data processing mechanism can work seamlessly, and a complete set of Windows Phone Bluetooth communication software programs are developed. This study stands as a piece of pioneering work in Taiwan, using Windows Phone 8.1 as a system development platform. Finally, Bluetooth signals are bridged by this architecture with an extended communication range compared to a typical Bluetooth communication, and the congenital defects in Bluetooth can be remedied by the characteristics of ZigBee. A routing device, supporting a large number of nodes, is scheduled to be developed, and is expected to offer an extended communication range in the future, using a regular ZigBee system. A succession of research projects are definitely scheduled by the authors as an extended application of this work.

Acknowledgement

This research is sponsored under grants 106-2221-E-129-008 and 106-2632-E-129-002 by the Ministry of Science and Technology, Taiwan. The authors feel deeply indebted to the

Volume 4, Number 3, 2017
ISSN 2409-7284 (Online)

Department of Information and Communication, St. John's University, Taiwan, for all aspects of assistance provided.

References

Chen, N.X. & Yang J.T. (2006). *Digital Learning Theory and Practice*. Taipei: DrMaster Press.

Chen, G.Z. (2008). *Digital Media Design Education Research*., Unpublished master's thesis, Yunlin University of Technology.

Conti, G. J., & McNeil, R. C. (2011). Learning Strategy Preference and Personality Type: Are They Related? Journal of Adult Education, 40(2), 8.

Aamoth, D. (2016). *First Smartphone Turns 20: Fun Facts About Simon*, Available at http://time.com/3137005/first-smartphone-ibm-simon/, 2014 (Visited on Sep. 16, 2016).

Android Wear (2016). *Android Wear*, Available at https://www.android.com/wear/, 2014 (Visited on Oct. 10, 2016).

Apple (2016). *Apple – Apple Watch*, Available at https://www.apple.com/watch/, 2016 (Visited on Oct. 10, 2016).

Bluetooth SIG (2016). *Specification of the Bluetooth system*, Core Version 4.2, Data Sheet, Available at https://www.bluetooth.org/en-us, 2016 (Visited on Aug. 13, 2016).

China Mi Technology (2016). *Mi Band*, Available at http://www.mi.com/shouhuan/, 2016 (Visited on Aug. 14, 2016).

IDC (2016). *In a Near Tie, Apple Closes the Gap on Samsung in the Fourth Quarter as Worldwide Smartphone Shipments Top 1.3 Billion for 2014, According to IDC*, Available at http://www.smartphonemarketresearch.com/in-a-near-tie-apple-closes-the-gap-on-samsung-in-the-fourth-quarter-as-worldwide-smartphone-shipments-top-1-3-billion-for-2014/, 2015 (Visited on Oct. 12, 2016).

IDC (2016). *Smartphone OS Market Share, Q2 2016*, Available at http://www.idc.com/prodserv/smartphone-os-market-share.jsp, 2016 (Visited on Oct. 12, 2016).

Microsoft Band (2016). *Microsoft Band | Official Site*, Available at https://www.microsoft.com/Microsoft-Band/en-us, 2016 (Visited on Oct. 10, 2016).

Net Applications (2016). *Desktop Operating System Market Share*, Available at http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=1, 2016 (Visited on Oct. 12, 2016).

TI (2016). *CC2592 2.4-GHz Range Extender*, Data Sheet, Available at http://www.ti.com/product/cc2592, 2016 (Visited on Oct. 10, 2016).