

Research Article

A Data Transmission Technique for Personal Health Systems

Jih-Fu Tu¹ and Chih-yung Chen²

¹ Department of Electronic Engineering, St. John's University, Tamsui 251, Taiwan

² Department of Information Management, St. John's University, Tamsui 251, Taiwan

Correspondence should be addressed to Jih-Fu Tu; tu@mail.sju.edu.tw

Received 14 September 2013; Accepted 10 October 2013

Academic Editor: Teen-Hang Meen

Copyright © 2013 J.-F. Tu and C.-y. Chen. 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.

We used the modular technique to design a personal health data transmitter (PHDT) that is composed of the following components: (1) the core is an embedded signal chip, (2) three kinds of transmutation modules such as USB, RF, and UART, (3) an I²C interface is used to acquire the users data, and (4) through Internet it links to the cloud server to store the personal-health data. By the experiment, we find that the modular manner is feasible, stable of functional, integral, and accurate, while it is exploited to design the PHDT. For the experiment, we present each module algorithm to find that our system is very helpful to people.

1. Introduction

Recently, personal health management has become one of the popular research topics. For instance, to acquire and measure the personal health signal, the techniques of personal health data transmission, personal health data record manner, embedded biomedical analysis platform, and so forth. Shu [1] used mixed signal to design the system functions in the signal chip PSoC and then exploited NI USB-6251 and Tektronix TDS3054e scope to verify the functional module the authors finally linked the sensors to this PSoC to accomplish miniaturization and modular measurement system [2–5].

In this paper, we integrate the personal medical measurement instruction and the cloud server to support the transmission of the personal health data by Internet. In this result, we create a central information management and data sharing storage. This work is made from a module manner, in which we support an automatic PHDT that can rapidly and easily link to the cloud server to transmit personal health data. We also employ a mobile to substitute the personal computer and to access the personal health data via the Internet.

In the recent years, the previous works aimed at storing the physiology signals and then sending them to the remote information center that is either PC or server. So, the end users cannot employ a simplified manner to send the personal health measurement data to the cloud server [6, 7].

The wireless network electric appliances are developed for peoples, home, enterprise, and so forth. Therefore, the near-distance wireless communication protocol becomes very important to them [8, 9]. Thus, most of short distance wireless communication protocols and techniques are estimated in the free-pay ISM bandwidth [1, 10].

The rest of this paper is organized as follows. In Section 2, we introduce the short distance transmitting technologies. The structure of the PHDT is constructed in Section 2. We make the module algorithms and the characteristics in Section 3. The system functions are verified in Section 4. Finally, we make the summaries in Section 5.

2. The Infrastructure of PHDT

Figure 1 illustrates the structure of personal health data transmitter (PHDT) in which the MCU is used as central control core. The personal medicine data can be transmitted by one of RF, USB, or UART modules to a personal health data buffer. The recorded data through the I²C module inputs to the users data buffer. The MCU indicates the IP of the user and initials the network module to connect to the cloud server. Finally, the user can exploit the user ID, and password to login the server to transmit the personal health data to the cloud server.

Three kinds of input/output interfaces are offered in the transmitter as follows.

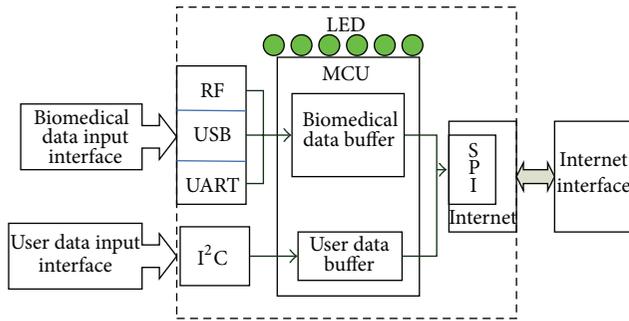


FIGURE 1: Transmitter system block diagram.

- (1) Personal health data input interface is composed of three modules, which are
 - (i) RF module: it supports wireless method to transmit the personal health data in channel;
 - (ii) USB module: it supports USB method to transmit the personal health data in channel;
 - (iii) UART module: it is an expandable data transform interface and classifies four types, UART to RS-232, UART to RS-485, UART to Zigbee, and UART to Bluetooth. We also employ this UART interface to connect RS-232 to PC and to service.
- (2) User data input interface is composed of an I²C and an alternation transmitted interface. The user data card supports different users and replaces the user data transform module, to acquire the primary data of user in the transmitted channel.
- (3) Internet module supports a data transmitted interface between transmitter and network.

2.1. MCU Module. The core of MCU module is achieved in PIC24FJ128GB106 [11]. This chip has an embedded USB 2.0 OTG, 128 KB core memory, 16384-byte data memory, and several communication interfaces such as UART, SPI, and I²C.

The main job of MCU module is to integrate the overall modules and to control each module. In order to fairly control each module, we exploit the polling manner by taking turn for each module. The specifications of MCU modules are described as follows.

- (1) MCLR chip reset circuit is operated while the SW1 is turning on. At the same time, the MCLR immediately issues a *Reset* signal to reboot the MCU.
- (2) ICSP chip online circuit has 6 pins to connect to the PICkit3 recorder. We can directly upgrade the driver of MCU via an online PCB.
- (3) XTAL is an external crystal circuit and the operation frequency is 16 MHz.

2.2. Network Module. Figure 2 shows the network transform module, which is composed of two parts, hardware and software.

- (1) Hardware:
 - (i) network chip control circuit,
 - (ii) network connection circuit embedded transformer.
- (2) Software:
 - (i) network chip driver,
 - (ii) small scalar TCP/IP protocol level *uIP* [12],
 - (iii) the application program includes DHCP client, DNS client, and FTP clients.

The previous characteristics are described as follows.

- (1) Network driver is used to control network chip; it also can write the external packages to the sent buffer of network chip.
- (2) *uIP* protocol is used during the package transmission in Internet, and it is necessary to add IP and TCP headers. So, we use the TCP/IP function while the job has been accomplished. Therefore, we can transplant the *uIP* protocol and handle the TCP/IP of a small scalar embedded system.
- (3) The network application program includes
 - (i) DHCP which client is realized by the *dhcp.c*. The DHCP client can acquire and obtain an IP address, gateway, and DNS from DHCP server;
 - (ii) the DNS client functions which are realized by the *resolv.c* to acquire the cloud server IP address;
 - (iii) FTP client which accomplishes the ftp function through the *client.c*.

The data is transmitted by the transmitter and the product that is presented as follows. First, the MCU is set in a polling manner to query each peripheral device whether a package be handled or not. If it is true, the control right passes to *uIP*. What the package will deliver that is dependent on *UIP_APPCALL()* or *UIP_UDP_APPCALL()*; hence, the application program takes over the data transmission process.

The network module is constructed in ENC424J600. The network control circuit is combined of a four-line SPI interface and an ENC424J600's interrupt request.

The control programs of AVR MCU and ENC424J600 are modified to fit the PIC24FJ128GB106 [11] in order to control the drivers. We also edit a system level interface program, *network.c*, which includes the interface subroutines between MCU module and network module that are listed in Table 1.

2.3. *uIP* Protocol. *uIP* is proposed by Adam Dunkels and it is a free open source code for small scalar TCP/IP protocol [12], in which he had deeply described the transplant process for porting *uIP*1.0 linking to LPC23xx/24xx. The process of *uIP* transplant is described as follows.

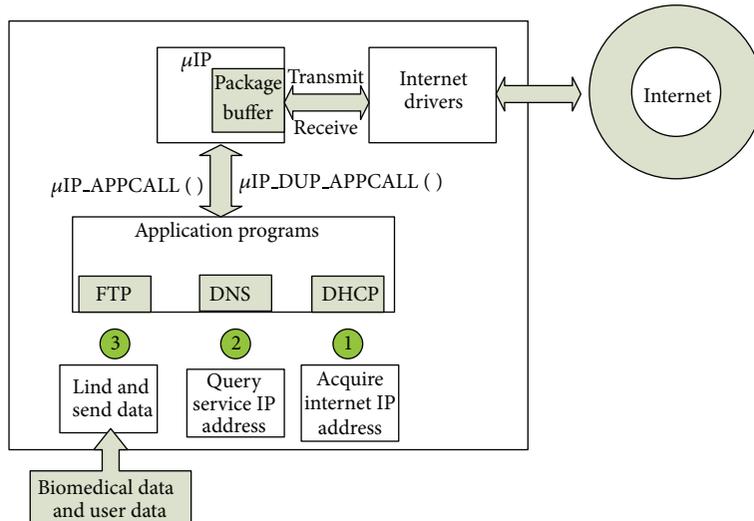


FIGURE 2: Network module structure.

TABLE 1: The sun-routines of network interface.

network_init()	Initializing the network chip
network_read()	Reading a received package
network_send()	Sending a network package
network_get_MAC()	Acquire the MAC address
network_set_MAC()	Setting the MAC address

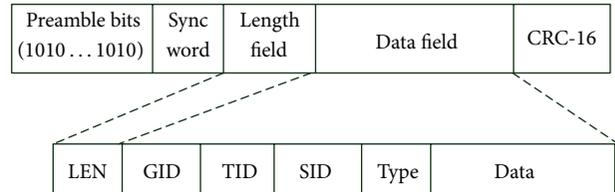


FIGURE 3: The formal package in RF module.

(1) Transplant timer:

- (i) at first, we complete the *clock_init(void)* function to initialize system clock module;
- (ii) and then the *clock_time_t* and *clock_time(void)* functions support a referred system clock.

(2) We transplant network chip driver to complete the network module interface.

(3) Adding *uip.c*:

- (i) we create *uipapp.c* to realize the return function of *uip_appcall()* and *udp_appcall()*; then
- (ii) we define the return-function in *uipapp.h*. It is added to *UIP_APPCALL* and *UIP_UDP_APPCALL*. Furthermore, we add them into the *uip_arp.c*.

(4) Adding the *dhcp.c* is to test the *uIP* function. At the same time, the DHCP server can acquire IP address, too.

2.4. RF Module. In order to obtain lower power consumption in near-distance transmission, we employ the 2.4 GHz ISM bandwidth in RF module. It is implemented by CC2500 RF controller, and a 4-wire SPI is used to connect between MCU and CC2500.

For most peoples at home, the basic request is a wireless network in short distance. So, we exploit a broadcast method

as the wireless network topology. The wireless network is recognized by the GID to detect whether the same network group or not. In the same GID, the network is composed of several members which are detected by the different SID. So, in the same group members have unique SID. In this working network, a master in central and its SID are 255. The rest members are set to the slave transmitter. The master uses polling manner to query whether the slave (i.e., medicine measurement instrument) transmits data or not. In order to implement the previous works, the RF module supports the following four subroutines:

- (i) *ccInit()*: initials RF module to set GID and SID;
- (ii) *ccSendPkg()*: sends data package via RF;
- (iii) *ccGetPkg()*: reads RF data package;
- (iv) *RFProc()*: is the main program of RF module.

The format of data package in RF module is defined in Figure 3, where the LEN indicates data length and the data field includes group ID (GID), target ID (TID), source ID (SID), data classify (TYPE), and transmitted data (DATA).

In RF module, the main program will call the *SaveData()* subroutine during data receiving. When a data packet is received firstly to recognize the GID, if the TID equals SID, then the package will be received.

2.5. USB Module. Most of medicine measurement instruments have the USB interface. For this reason, the transmitter

must own a USB host. In this work, the USB software is based on Microchip USB stack and we use the *USBConfig.exe* tool to complete the USB. In this module, we support a *usbmain()* subroutine of system level to call the *usbmain()* subroutine; while the device receives data by the USB, it will trig the *SaveData()* subroutine to handle the data receiving works.

2.6. UART Module. For most commercial products, they offer a UART interface as a converter such as Bluetooth to UART and Zigbee to UART. In this paper, we support two types of UART modules, UART1 and UART2, that are described as follows.

The UART1 is connected to an external RS232 interface. Then the transmitter can easily link this RS232 to the PC and connects to the superterminal.

We use the *UARTProc()* function to manage the UART2 and exploit an interrupt scheme to handle the received data. When data is received by UART2, immediately the *SaveData()* subroutine is called to trig the data reception process.

The UART2 module has VDD and GND to support the power to Zigbee. The MCU's UART2 directly connects to the Zigbee to realize the wireless work; there are from UART2's RX2 to Zigbee's TX1 and from UART2's TX2 to Zigbee's RX1.

2.7. I²C Module. Another important advantage of transmitter is a plug-in manner to the user. When the I²C module is used, the necessary power of user data card is supported by the VDD, GND of I²C, and the SCL3, and SDA3 of I²C is connected to the 24LCXX EEPROM's SCL and SDA, respectively. We use the *IDProc()* subroutine to handle the I²C module and initial the user input request.

2.8. LED Module. The LED module is used to indicate the operation status during transmission. The signals of each LED are defined as in Table 2. We also support four subroutines to control the LED, which are

- LEDInit()*: initials LED module,
- LEDON()*: turns on the LED,
- LEDOFF()*: turns off a specific LED,
- LEDTGL()*: turns on/off the specific LED.

3. Algorithms

In advance, we build up a practicality integration development environment (IDE) to verify and support our assumption. For the PCB of the physical devices, we use the KiCad [13] to do, and the experiment PCB is shown in Figure 4.

The system verification work has two stages, individual module function and system integration function.

- (1) The individual module function.
 - (i) UART module function,
 - (ii) I²C module function,
 - (iii) USB module function,

TABLE 2: LED indication.

LED1	Personal data input over
LED2	Insert internet line
LED3	Link to internet
LED4	Insert USB devices
LED5	USB data read
LED6	Data transmitting
LED7	Power on

- (iv) RF module function,
- (v) network module function.

(2) In system integration function stage, users can select one of the input channels.

- (i) RF transmitter + personal data card,
- (ii) medicine measurement instrument + personal data card,
- (iii) Zigbee transmitter + personal data card.

3.1. Main Function. Before the individual module function is verified, we must build a main program in terminal. So, we insert the calling operation in the main program to call module function. The main program is shown as in Algorithm 1.

3.2. UART Function. In this paper, we support two UART modules which are UART1 and UART2. The UART1 is used as a terminal to present the system operation message. In advance, we must confirm the UART circuit correctness. And then the transmitter is able to connect PC via the RS-232.

The UART2 is connected to the Zigbee transmitter. The UART2 also is one of the input channels of the personal medicine data. The works are among UART1 and UART2 as follows.

- (1) First, the UART2 issues a request to send data in a fixed interval time.
- (2) During data receiving stage, a received signal is sent from UART1 to the other transmitter.
- (3) Finally, the "data accepted" is issued from UART1 while being accomplished.

If we want to let the UART2 operate, we only insert the *UARTProc()* subroutine into the main program and the process is shown in Algorithm 2.

3.3. I²C Function. The I²C module is done by inserting the *IDProc()* subroutine into the main loop that is shown in Algorithm 3.

3.4. USB Function. We need to insert the *usbmain()* subroutine into the main program to implement the USB module. It is shown in Algorithm 4.

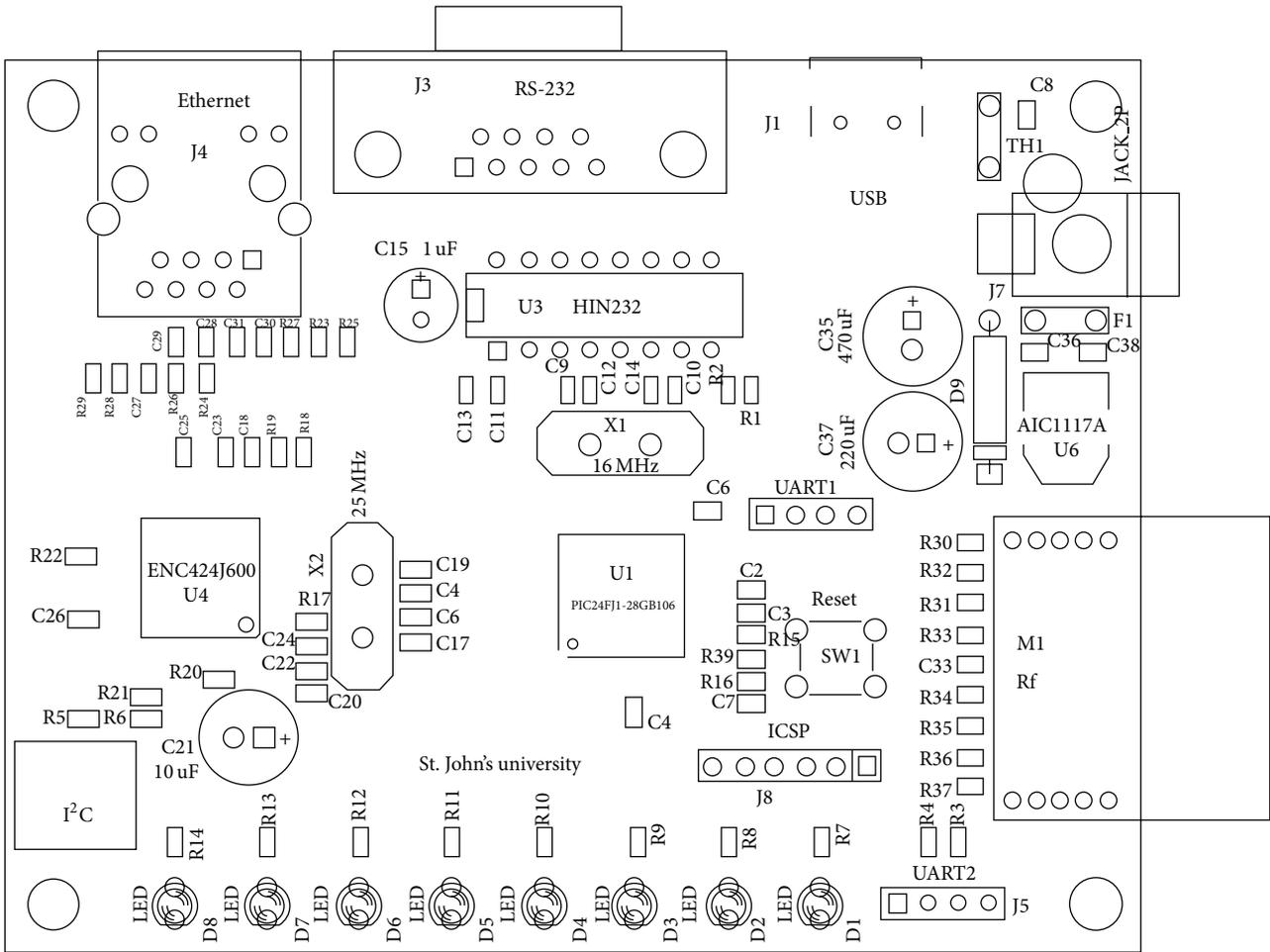


FIGURE 4: The implemented PCB.

```

int main (void)
{
  Init(); //system initialization
  //--- Main Loop ----
  while(1){
    Timer1Proc(); //call the system Timer module functions
  }
}
    
```

ALGORITHM 1

```

//--- Main Loop ----
while(1){
  Timer1Proc(); // system Time
  UARTProc(); //UART2 module handle
  function
}
    
```

ALGORITHM 2

```

//--- Main Loop ----
while(1){
  Timer1Proc(); // system Timer
  IDProc(); //I2C module handle function
  uipmain(); //ETHERNET module handle function
}
    
```

ALGORITHM 3

```

//--- Main Loop ---
while(1){
Timer1Proc(); // system Time
usbmain(); //USB module handle function
}

```

ALGORITHM 4

```

//--- Main Loop ----
while(1){
Timer1Proc(); // system Timer
IDProc(); //I2C module handle function
uipmain(); //Network module handle function
}

```

ALGORITHM 6

```

//--- Main Loop ----
while(1){
Timer1Proc(); // system Timer
RFProc(); //RF module handle function }

```

ALGORITHM 5

```

USER: 8860100002011000008
PASSWD: 1234567887654321
URL: ftp.ecaretek.com
UART1: Baudrate 57600
UART2: Baudrate 38400
UART2: request
UART2: request
UART data is linked. . .
Machine name: GM06L
Product: Glucose
1, 2009, 11, 30, 13, 50, Lo., mg_dL
2, 2009, 11, 30, 13, 45, Lo., mg_dL
3, 2009, 11, 30, 13, 43, 68, mg_dL
4, 2009, 11, 30, 13, 40, 37, mg_dL
5, 2009, 11, 30, 13, 27, 36, mg_dL
6, 2009, 11, 30, 13, 27, Lo., mg_dL
7, 2009, 11, 29, 13, 27, 33, mg_dL
8, 2009, 11, 28, 13, 27, 55, mg_dL

```

Box 1: Data transmission via UART2.

3.5. RF Function. For implementing the RF function, we need to insert the *RFProc()* subroutine into the main program that is shown in Algorithm 5.

3.6. Network Function. The work of network module is made from the following steps:

- (1) trig the network chip,
- (2) set the parameters of the dynamic host configuration protocol (DHCP) to acquire the network, and
- (3) make sure the IP of personal data card is accepted by the cloud server.

In order to implement the previous processes, we insert the *IDProc()* and *uipmain()* subroutines into the main program to implement the network module that is shown in Algorithm 6.

In this paper, we use the *Packetyzer* network package acquirement analysis tool to verify network module and to acquire DHCP query product.

4. System Function

As a reason, the transmitter has three types of data input channels, RF, USB, and UART, in order to prevent the data from being transmitted by different channels simultaneously. We exploit a software scheme to handle the personal health data in the collector. For personal health data collector maintaining, it is realized by the *SaveData()*, where we can receive a primary data from RF, USB, and UART2. After the primary data is processed, a signal is sent by the *call()* subroutine to notice the data being “accepted,” “unaccepted,” or “completed.” By this way, it is helpful to the software development and maintenance.

The personal health measurement instruction instrument using USB transform wire and transmitter connects to physical level verify manner follow the personal medicine data input different channels, exploiting the following operation model to verify.

- (i) RF transmitter + personal data card,
- (ii) health measurement instrument + personal data card,
- (iii) Zigbee transmitter + personal data card.

We use the *Filezilla FTP* transform software to upload/download the file to/from server. Both uploading and downloading the personal health data are via the UART2 channel. The result is shown in Box 1.

5. Conclusions

We use three normal interfaces, such as USB, UART, and RF to implement the purposed device. This paper issues a personal health data transformer, RF interface, and acquirement of the user personal health measurement data. Adding a replacement personal data card, we can, according to the data card, access the user accord, password, and server IP. For this reason, we can transmit the personal health data via the network sent to transform the cloud server and to crate the personal healthy information file.

The advantages of the personal health data transformer (PHDT) are listed as follows.

- (1) We used wireless technology directly to connect PHDT to acquire the records from personal health measurement.
- (2) Substitutable user data card: the user’s data card records the following data: user ID, password, and

server IP. The substitutable design manner allows dissimilar users with higher speed to replace the user data.

- (3) Internet transform functions: we can use the PHDT to acquire dynamic IP address. By the DNS address, we recognize the query cloud server IP address and enter into the cloud server by a dynamic FTP. Therefore, the measured records of personal health are delivered to cloud server to estimate the personal health information.

References

- [1] K. L. Shu, *A study of embedded platform of bio-medical signal measurement [M.S. thesis]*, Nantai Techniques University, Tainan, Taiwan, 2007.
- [2] A. Z. Alkar and U. Buhur, "An internet based wireless home automation system for multifunctional devices," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1169–1173, 2005.
- [3] M. B. Huang, *Brain Wave Recorder with USB Interface Module [M.S. thesis]*, ChoYn University.
- [4] Microchip Cop., "USB embedded host stack," Microchip Application Note AN1140, 2008.
- [5] K. Römer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, 2004.
- [6] E. Callaway, P. Gorday, L. Hester et al., "Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 70–77, 2002.
- [7] K.-I. Hwang, J. In, N. Park, and D.-S. Eom, "A design and implementation of wireless sensor gateway for efficient querying and managing through world wide web," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1090–1097, 2003.
- [8] J. -D. Lai, *A embedded platform of bio-medical with wireless [M.S. thesis]*, Jao-Ton University, 2008.
- [9] H. Z. Lee, *Regular report of foods in home carewith RFID [M.S. thesis]*, Asia University, Taichung, Taiwan.
- [10] E. F. Liu, *A transform system for body temperature and blood press with wireless [M.S. thesis]*, Asia University, Taichung, Taiwan, 2010.
- [11] Microchip Cop., "PIC24FJ256GB110 Family Data Sheet," <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en531079>.
- [12] NXP Semiconductors, AN10799, "Porting uIP1.0 to LPC23xx/24xx V2," NXP Semiconductors, 2009.
- [13] Microchip Cop., "PICkit 3 In-Circuit Debugger," <http://www.microchip.com/pickit3/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

