A Method for Dynamically Loading and Running Programs Between a Wearable Device and a Mobile Phone by BLE

QI FENG[†1] LEI JING[†2] ZIXUE CHENG[†2]

Abstract: Nowadays wearable devices need to run variety kinds of programs, but resources in a wearable device are limited. We have already developed a wearable device called Magic Ring, which can conveniently control household appliances. In some particular situations, users are not convenient to use the mobile phones, for example, when taking crowded subway train, or an emergency message is coming. We want to use wearable device to control mobile phone in such situations, so we have issues on how to make connection and transmit programs between wearable device and mobile phone, and automatically loading programs on wearable device. As the first step, we discuss about the communication mode to make connection and solutions on the automatically loading problems into a wearable device.

keywords: Computer networks, wearable computing, wearable devices, Bluetooth low energy, automatically loading programs

1. Introduction

Nowadays, many wearable devices are being developed and they are very popular with news media, world's technology manufacturers, and group of internet users. There are many latest wearable technologies, such as smart watches, rings, bracelets, shoes and more. They can be both snugly worn on the user's body and provide some convenient feature in our daily life. The devices can collect user data, control household appliances, and establish communications with other device. The devices can connect to the Internet with various communication methods, such as Wi-Fi, Bluetooth, and 3G/4G communication.

Recently, mobile phone especially smart phone has more advanced computing capability and connectivity than basic feature phone, is attract attention of the world. A mobile phone combined the features of PDA, media player, digital camera, GPS navigation unit. A mobile phone can be connected with Internet or other device using various communication technologies, such as Wi-Fi, Bluetooth, and 3G/4G communication.

Since many wearable devices have limited computation resources by reason of memory and the power of CPU. It's difficult for a wearable device to save plenty of applications, or execute a big program in such kind of environment.

In this paper, we aim to realize communications between wearable device and mobile phone. To this end, we need to solve some problems: In the wearable device without BLE module, design a BLE communicating mechanism to make a connection between wearable device and mobile phone using middle device. After realize this problem, another issue for us in this paper is dynamically loading and running programs between a wearable device and mobile phone. Obviously, cable-based program updating method can not meet this kind of scenarios. Therefore, we propose a wireless program loading method to remotely load the selected program onto the client device without user intervention. Our previous study have already realize a new type of wearable device called Magic Ring as shown in **Fig. 1**, which can control household appliances like TV, air-condition, etc. We use it as the wearable device to test our method.

Magic Ring is a wearable device can wear on a finger. [3-5] It has wireless sensors to collect the data of the finger movement in order to recognize people's finger gestures, which easily and flexible provide a natural and convenient interface. We use this device to connect with mobile phone, and dynamically loading programs from mobile phone. Our contributions are: establish connection between wearable device and mobile phone in BLE communication mode. Use wearable device as client, dynamically loading and running programs stored in mobile phone.

Transparent computing is a new paradigm, which was proposed by Yaoxue Zhang in 2004 [2]. That is to store a big application program or OSes in a server and dynamically download OS and applications into a client in a streaming block manner on demand. That existing transparent computing methods have been aimed to PC, Smartphone, Tablet, etc. assuming an OS will be downloaded and used. About our research, we aim to realize transparent computing, specially the block streaming execution in a non-OS wearable device.

The rest of the article is organized as follows. Section 2 give the model of communication between wearable device and mobile phone. Section 3 proposes the system design of our method. Section 4 shows the implementation and evaluation of the proposed method. Section 5 concludes the paper.

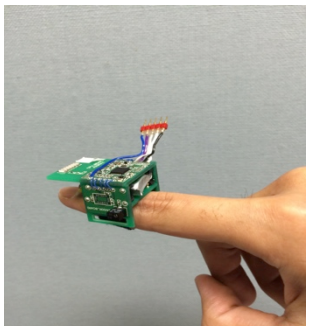


Fig. 1 Magic Ring

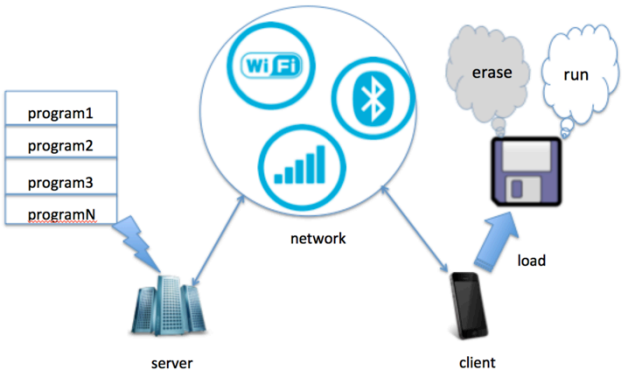
†1 The graduate school of computer science and engineering, The University of Aizu, Aizuwakamatsu city, Fukushima, 965-8580, Japan Email: m5172104@u-aizu.ac.jp
†2 The school of computer science and engineering, The University of Aizu, Aizuwakamatsu city, Fukushima, 965-8580, Japan Email: z-cheng@u-aizu.ac.jp

2. Model

2.1 Model of PDLRA

Model of PDLRA (Program Dynamically Loading and Running Architecture) is illustrated in **Fig. 2**. The whole PDLRA consist of three parts: servers storing programs are on one end, clients using programs are on the other end; the two ends are connected through wireless network such as BLE.

Fig. 2 Model of PDLRA



The server is the programs provider, in which programs are stored in the server. The server generally has customizable storage spaces and efficient computing abilities. Due to this point, the server will analyze the request received from client side, and make a computing process to elect the program which client actually required. In some particular situation, the client has extremely limited memory resources. The client will make a deblocking process to divide program into many blocks, then send these blocks to the client side use the network.

The client is the programs user, and it is request originator to establish the network communication. Actually the client has various programs to execute, but its own storage space is insufficient. Such as wearable electrical watch, ring, and glasses. Programs have occupied most of the CPU time and memories, so that it's necessary for it to loading and running programs dynamically from server's side. Due to the packages for communication are not enough to load a whole program. A big program will be divided into small independent blocks, downloaded into device timely and after the client received a block, it will write the block of a program into insufficient memory and running it. With the execution process, the transmission between server and client is continuously being carried out. In order to avoid waiting because of memory space crunch, the client side will erase the block from its memory when it is done, then other blocks will be wrote into the memory by repeating the process above.

The network between server and client, provide communication modes for establish connection. It could be Wi-Fi, Bluetooth, 3G/4G and etc. Sometimes the communication technologies are different between two sides, it's infeasible to establish connection. In this case, we use a middle device to connect with each side, actually the middle side has the communication modes to connect with each side, and convert one communication mode to another, In order to unify communication interface.

3. System

3.1 Basic Idea

A wireless dynamically programming (WDP) method is proposed for program loading and running. WDP method has some innovation points such as the program for transmit depend on clients' needs instead of by the servers' side.

Furthermore, this WDP method is designed for wearable devices have limited storage spaces and computing resources. Therefore, we propose a kind of lightweight wireless boot strap loader (WBSL) as shown in **Fig. 3**.

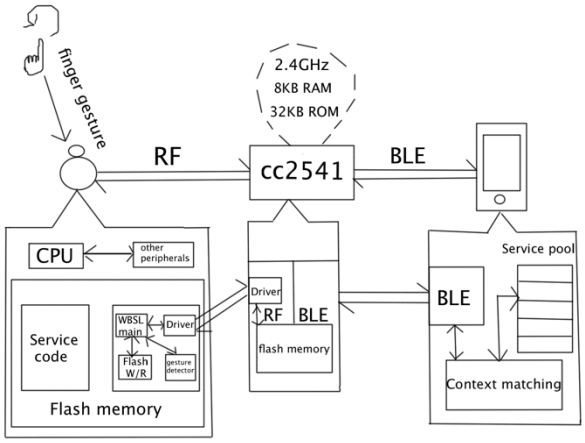


Fig. 3 System Architecture

We also use a middle device which name is CC2541 provided by Texas Instruments as shown in **Fig. 4**, The CC2541 is a power-optimized true system-on-chip (SoC) solution for both Bluetooth low energy and proprietary 2.4-GHz applications, combines the excellent performance of a leading RF transceiver with an enhanced 8051 CPU, in-system programmable flash memory, 8-KB RAM.

Fig. 4 CC2541 development kits

The system consist of three components:
Magic Ring: A prototype of ring-shape wearable device. It is a sensing mote integrated with sensing, processing, and communicating functions, it will enable diverse kinds of applications in the intelligent computing environment such as home appliance control, communication, and activity tracker. As the client in this system, the WBSL on Magic Ring is executed when specific gestures are detected. Especially the Magic Ring will reset to disable all interruptions and load the WBSL from address which is predefined in the memory space.

As shown in **Fig. 3**, after that the WBSL will send connection request to the middle device CC2541 using light-weight wireless communication technology. If it is also succeed for communication between middle device and client, WBSL in Magic Ring writes the received program into proper flash address. At last step, Magic Ring will reset to load the program from the flash memory into RAM and execute the program. **CC2541**: a board supports both RF and BLE, which be used as a middle device to connect Magic Ring and mobile phone. Once the power turned on, CC2541 switches RF to receiver mode and waiting for the request from client side. When CC2541 have received request message from client, it will change communication mode to BLE and make a connection with mobile phone using BLE protocol stack, if the connection is successfully established, it will send the request to mobile phone, if it is succeed, it will received the program and send them back to the client side. **Mobile phone**: have advanced computing capability and connectivity than basic feature phone, we put variety programs into it. Once the request is sent to mobile phone from middle device, Firstly it will agree to establish a connection in BLE mode, then use context matching module to select proper program and then send them back to the middle device, at last send back to the client.

4. Consideration on Implementation

4.1 Outline of the Implementation

In Figure 3, we proposed a method for dynamically loading and running programs from mobile phone to Magic Ring using middle device CC2541. For the communication between mobile phone and CC2541, we need to:

(1) To set up a connection between mobile phone and CC2541 using BLE protocol stack.
(2) Try to download programs from mobile phone to CC2541 and stored programs on CC2541.

Currently, although we have not implemented (1) between mobile phone and CC2541, we use two CC2541 for realizing (1) and (2) as the first step of the implementation, due to CC2541 have the BLE module similar to the mobile phone, and it's easy to make connection between 2 CC2541.So we do the experiment using two CC2541 boards, to transmit program from one board to another. As a result, the program stored in one board could be load to another board use BLE. About dynamically loading and running program between the middle device and Magic Ring, we need to:

(1) On server side, dividing a program which wait for transmitting into a set of blocks.
(2) On client side, loading each block of the program from the server side, and store them into specified flash address.
(3) On client side, when all blocks of program were sent, running the block from the specified flash address.
(4) On client side, try to realize loading and running in parallel execution.

At our current stage, we can realize the (1) to (3), i.e. dividing, streaming, and loading the program in a sequential way (shows in **Fig. 5**). But about the item (4), It's under developing. Block scheduling and memory multiplexing are the challenge works for us.

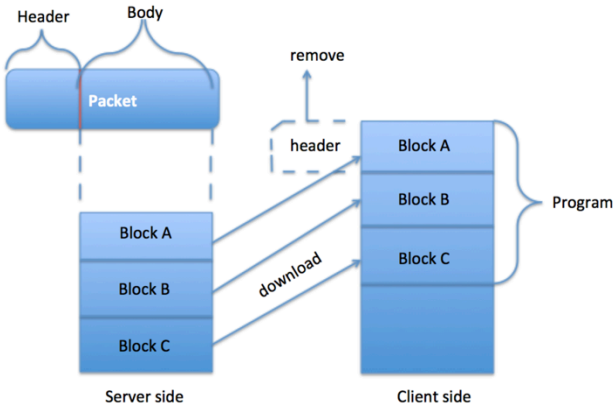


Fig. 5 divide, stream and load program

From **Fig. 5**, It's not hard to know that we divide programs into blocks in server side, and then store blocks into packets, as we start the communication between server side and client side, server will send the packets to the client side's flash memory in a good order. After removing the head of each packet, the blocks will be sequentially stored in the special addresses of client's flash memory. When all the blocks arrive to the flash memory, the program could be successfully assembled and loaded into RAM for execution by the boot loader.

The length of each block in **Fig. 5** depends on the packet length, and all the blocks have the same length. The segmenting process in this way is easy for server to divide blocks but also the disadvantage is obvious.

Fig. 6 Try to describe the scene which support item (4). But we also use the segmenting method in **Fig. 5**.

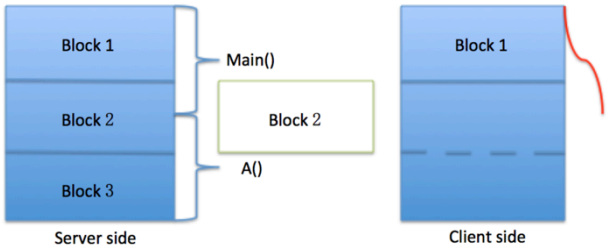


Fig. 6 Demonstration

From **Fig. 6**, server select a program required by client, which have a main() function, and a function A() will be invoke in the main() function. We assume that the loading and running are in parallel execution, when the first block arrived at client side, it will execute whatever the blocks are complete or not. But in Block 1, it only have part of main() function, when it execute till the end of block1, the block 2 has not go to the client side, the program will stopped at the edge of curve and report errors.

In order to successfully loading and executing programs in this scene, we should divide a program into such blocks that just matching each function of the program. But it's a challenge work for us, we are still researching on it.

4.2 Implementation tools and environment

1) Hardware employed

In our Magic Ring platform, we use micro computer CC1110, provided by Texas Instruments. It is a low-power sub-1Ghz system-on-chip designed for low-power wireless applications. It has 32kB of in-system programmable flash memory and up to 4kB of RAM, as well as a radio module for wireless communication.

2) Development tools

We use IAR Embedded Workbench provided by IAR System company, which is a compiler supporting many well-known semiconductor's microprocessor. In this experiment, we use its sub-series software IAR Embedded Workbench for 8051, it can properly work with CC1110, compile and link the code, generate (.hex) executable file that can be debugged on CC1110 board.

4.3 A prototype of the system

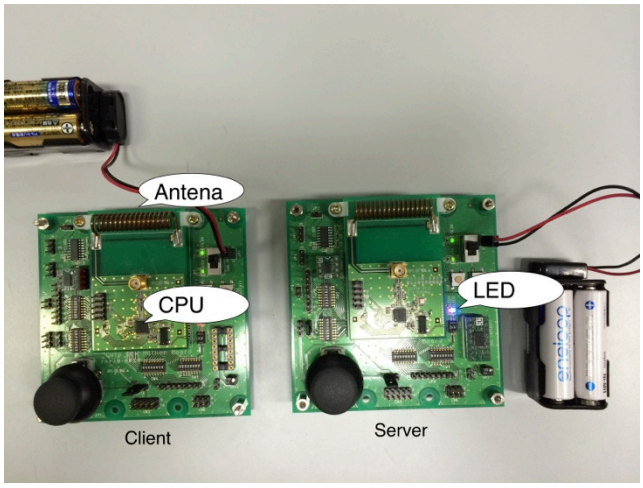


Fig. 7 Experiment System

We use two boards of CC1110 for the experiment, as shown in **Fig. 7**. One is used as client, and another is used as the server. Although the size of the boards is bigger than Magic Ring, the hardware and software are the same as Magic Ring, so our method can be easily implanted into Magic Ring, the bigger board allows us to debug and perform the experience more easily.

We installed a test program on the server side, the test program is a simple program consist of two functions: main(led_blink) and led2_blink. The two functions mainly let the two LEDs on the board to blink. Between server and client, it will establish connection under a specified protocol support wireless packet transfer. When the client side need to load the program, click the S1 button on the board of client side, trigger the interruption and send request to server side in order to get the program. After server side received the request, it will transmit the functions to the client side one by one. At same time, the client side load the two functions onto the specified flash memory. Then the boot loader preinstalled on the client will jump to the start address of the test program to execute. We will observed that two LEDs in client side are blink.

4.4 Experiment and discussion

In order to prove our proposal, we have set an experiment in the control group:

(1) Only load one block, that is the main(led1_blink) function from server side to client side. According to the debug snapshot shows in **Fig. 8**, we can analyze that the main function is loaded into flash memory address from 0x2000 to 0x207E. Because of function2 has not been loaded into the memory, the program will stopped when it run to cursor of call blink1(). The phenomenon is LED1 will blink some times and then stopped.

(2) Load both two blocks into memory from server to client, from the debug snapshot shows in **Fig. 9**, we can analyze that both main() function and blink1() have been loaded into the memory, and obviously the phenomenon is both LED1 and LED2 blinked.

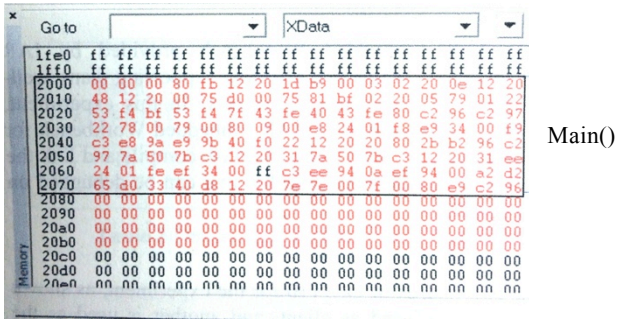


Fig. 8 snapshot when only load one block

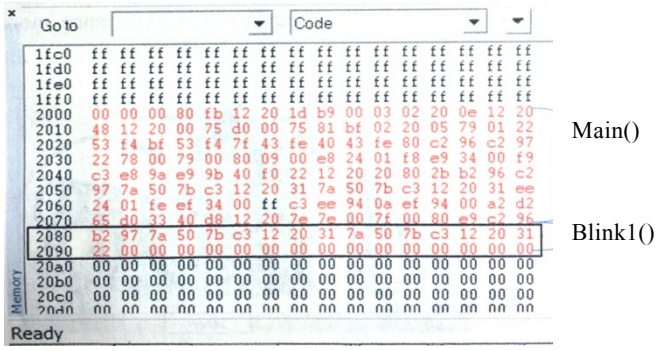


Fig. 9 snapshot when load both blocks

From the result of the above experiment, we know that we can divide a program into independent blocks. The blocks are sent from server to client in order. After receiving blocks, the client will store them into flash memory one by one. Finally, when all the blocks are stored in flash memory, it will be loaded into RAM for execution by the boot loader.

5. Conclusion

In this paper, we proposed a dynamically loading and running method, to extend the transparent computing paradigm into wearable device, we proposed PDLRA. Moreover, a WBSL-based WDP method is designed to enable programs to be loaded onto the device in a dynamical way. We also have done a feasibility research to see blocks of a program can be streamed. However, we are still working on sequentially load and run blocks in not a pure streaming way.

In the future, we will focus on block scheduling method and memory multiplexing method.

Reference

[1] Bluetooth SIG. Specification of the bluetooth system version 4[EB/OL].2010.http://www.bluetooth.com.

[2] Zhang, Y. (2004) 'Transparence computing: concept, architecture and example', *Acta Electronica Sinica*, Vol. 32, No. 12A, pp.169–174

[3] Jing, L., Yamagishi, K. et al. (2011a) 'A unified method for multiple home appliances control through static finger gestures', 11th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2011), Munich, Germany.

[4] Jing, L., Zhou, Y. et al. (2011b) 'A recognition method for one-stroke finger gestures using a MEMS 3D accelerometer', IEICE Tran. on Information, Vol. E94-D, No. 5, pp.1062–1072.

[5] Jing, L., Zhou, Y. et al. (2012) 'Magic Ring: a finger-worn device for multiple appliances control using static finger gestures', Sensors, Vol. 12, No. 5, pp.5775–5790.