2005－UBI－8（66）
2005／6／10

U[1]-chip: A Simple Wireless Communication Module for Local Service Discovery

Yee-Loo Foo[1], Yuichi Taguchi[1], Shunsuke Saruwatari[1], Hiroyuki Morikawa[1,2], Tomonori Aoyama[1],
Mikio Hasegawa[2], and Masugi Inoue[2]

[1] The University of Tokyo, 7–3–1 Hongo, Bunkyoku, Tokyo, 113–8656, Japan
{ylfoo, yuichi, saru, mori, aoyama}@mlab.t.u-tokyo.ac.jp
[2] National Institute of Information and Communications Technology, 3–4 Hikarinooka,
Yokosuka, Kanagawa, 239–0847, Japan
{mikio, inoue}@nict.go.jp

Abstract
In pervasive computing environment, we will be surrounded by services/resources. How best to access to these services is an interesting research topic. Some of these services can be delivered and completed instantly, which we name them as 'instant services'. To facilitate the delivery of instant services, we develop a novel wireless communication module called U[1]-chip. U[1]-chip is capable of fast device discovery, and fast delivery of small amount of data which results in very low power consumption. By connecting U[1]-chip to both user terminal and service provisioning devices, user can easily discover and use these available services. The conceptual design of U[1]-chip is presented in this paper, followed by its actual implementation.

1. Introduction

In ubiquitous computing environment, there are many devices providing various kinds of services in our surroundings. The importance and potential of service discovery has increased tremendously, as reflected in the amount of research efforts driven towards this goal [1 - 11]. Using a portable user terminal equipped with wireless technology, we shall easily discover and enjoy the available services. There are simple services that can be delivered and completed in a short time, for example, to download a bus schedule from an information kiosk. We name this type of service as 'instant service'.

To deliver the instant service to a user, we introduce U[1]-chip (*U-one-chip*), a small and simple, low-power and low-cost wireless communication module for local service discovery. Both user terminal and the devices in our surroundings can be equipped with U[1]-chips. User can then wirelessly discover these devices and gain control over them or get services from them. A pair of U[1]-chips sets up communication fast and also releases the communication channel fast, making it suitable for small amount of data transfer.

Empowered with U[1]-chip, it is envisioned that users can enjoy various instant services which have not been available before (Fig. 1). Connecting a U[1]-chip with an electronic device gives users the control of that particular device, e.g. light control. On the other hand, associating a U[1]-chip with some daily non-electronic objects augments them with new functionalities that extend their usability. For example, tagging an item with U[1]-chip allows us to electronically find out its location. U[1]-chip provides a fast and yet simple means of setting up a short-range wireless communication, particularly suit for instant service delivery.

In the next section, we describe the related works. Then we move on to describe the idea, design, and actual implementation in the following section. Also in the same section, the capabilities of U[1]-chip are evaluated and presented. In Section 4, we illustrate the functionalities of U[1]-chip through an application developed by us. Finally, we conclude this paper in Section 5.

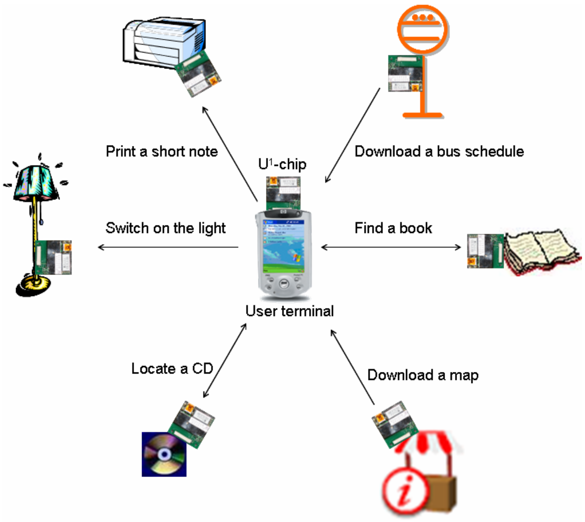

Fig. 1 U[1]-chip and its applications

2. Related Works

In this section we introduce some technologies that are close to our research works. They are Bluetooth, ZigBee, and RFID.

Bluetooth is a wireless technology for interconnecting devices and accessories using short-range and inexpensive wireless radios [21]. It targets personal area network, home network, and office network. Bluetooth devices can form a 'piconet' where a master Bluetooth device can simultaneously communicate with maximum 7 other active slave devices. The communication channel in Bluetooth can support both data (asynchronous) and voice (synchronous) communications with a total bandwidth of 1 Mb/sec. The maximum transmission range is 100 m.

In the instant service system that we envision, forming a network among devices is not necessary. Since continuous communication is not expected in the system, the slave U^1-chip devices do not have to remain synchronized to the master. At any time, one-to-one communication is more likely than one-to-many in instant service system, thus bandwidth sharing among the slave devices in U^1-chip system is not necessary. Bluetooth also suffers from slow device discovery problem. It has been reported by Basagni, Bruno and Petrioli in 2003 that in a Bluetooth network with high density, after 10 sec of device discovery, only a fraction of a node's one-hop neighbors have been discovered [12]. U^1-chip technology only targets services in the surroundings e.g. within 10 m range from a user, thus rendering a high-power transceiver unnecessary. And because of that, battery life is less of an issue to U^1-chip. It is also assumed that only a small amount of data (range from tens to hundreds of bytes) is to be communicated in the instant service. As a result, a low data rate transceiver is sufficient. The data rate in U^1-chip system is set at 4.8 kbps.

Aimed at control and sensor applications, ZigBee is a superset of the IEEE 802.15.4 specification, which defines the physical and MAC layers [13]. The technology allows a network to be formed among devices for better controlling and low data rate (20 kbps, 40 kbps, or 250 kbps) wireless communication purpose, where battery life is an issue [14]. The achievable distances range from 9 m to 70 m between stations.

ZigBee could be a suitable candidate technology for realizing instant service system, except that its networking functionality (one of the main functionalities) is not required in the instant service context. In general, Bluetooth and ZigBee suit continuous communication, different from the very short-term, or burst communication type encountered in the case of instant service. In short, we find that Bluetooth and ZigBee are overly fit for our ideal of instant service system. U^1-chip has a much simpler design, yet it is adequate in fulfilling the communication needs of delivering instant service.

Radio Frequency Identification (RFID) systems are a common and useful tool in manufacturing, supply chain management, inventory control, and even personal identification [15]. Industries as varied as microchip fabrication, automobile manufacturing, and even cattle herding have deployed RFID systems for automatic object identification [15]. RFID systems consist of RF tags (transponders) and RF tag readers (transceivers).

A problem with RFID system is that the transponders are merely 'responsive' and not interactive. In the instant service provisioning that we describe, we expect the service providing (slave) device to play a more active role in communicating with the master, e.g. to be able to transmit data to the master. Being not only for object identification purpose as what is the main purpose of RFID, U^1-chip suits this purpose since it is a complete wireless communication system.

Nevertheless, there is a feature in RFID communication that is found neither in Bluetooth nor ZigBee. It is the interaction between transceiver and passive transponder. Since a passive transponder is not externally powered but it relies on the signal power from the transceiver, both of them must come to very close in range, in fact virtually touching each other in order to enable communication. The act of touching actually allows user to explicitly appoint the intended communication partner, especially in the case where more than one are available in the proximity. In instant service environment where services are embedded in the surroundings of users, this kind of informative interaction is a desired feature. This feature will change the way we interact with our environment and open up a new range of possible applications [16], e.g. to get a service by simply gesturing. We intend to incorporate this feature in U^1-chip.

In the long run we aim to make U^1-chip a hybrid of the main features of Bluetooth, ZigBee, and RFID technologies.

3. Design and Implementation

In a local environment that is enriched with instant services, we consider multiple *service providers* and multiple *service clients* in the related communication model. In our communication model, the services are 'pulled' by the clients instead of being 'pushed' to them. This is to avoid the clients' frustration of receiving unwanted service advertisements. Every service provider listens to one common channel. A client broadcasts its service request to the common channel, and the related service providers shall reply.

It is also anticipated that a client deals with only one service at one time, thus a simple one-to-one communication is sufficient.

An appropriate communication model for instant service delivery system is necessary. In the U[1]-chip system that we propose, a pair of U[1]-chip devices carries out the communication task on behalf of a client and a service provider. A user who possesses a user terminal e.g. a PDA, is a client to some service provisioning devices e.g. an information kiosk. When a user expresses his/her desire for certain service, the U[1]-chip connected to his/her user terminal transmits the broadcast message. U[1]-chips connected to the service provisioning devices will then reply. The user then makes further selection from the offered services. Since only the U[1]-chip of the user can initiate a communication, we also name this U[1]-chip as the *master*. We name those attached to service provisioning devices as *slaves*. A slave U[1]-chip merely responds to the master. We also name the common channel as *control channel*, whereas the channel of which the master and slave communicate with each other one-to-one is named as *data channel*.

At the control channel, when more than one slaves reply to the master, collision occurs. We resort to adaptive tree walk (ATW) algorithm [17] to resolve the collision. We use a quadtree composed of the tree identities (IDs) of slaves. A tree ID consists of a service class of 2 bytes and an address of 8 bytes. The service class represents a category of service provided by the slave, and the address is a unique identifier for each U[1]-chip. When the slaves receive a query from the master, each of them replies in one of the four different time slots, depending on the first two bits of their tree ID [18]. If there is only one reply packet in a slot, the master can receive the reply successfully. But if there are more than one, collision occurs. The slaves that reply in this time slot are further divided into another four groups, and the same step is repeated until all slaves are discovered.

A complete communication process is represented in Fig. 2. After a master searches and discovers a slave, it is then able to request service from the latter. The master first scans for an idle data channel. When it is found, through the common channel the master signals the respective slave to tune to the desired data channel. Then the master sends the service request packet to the slave via the data channel. The control channel thus becomes free for others' broadcast transmissions. The slave replies over the same data channel too. When the communication is completed, both the master and slave release the data channel.

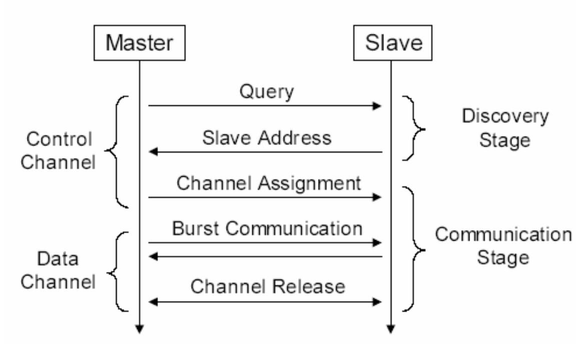

Fig. 2 Communication flow

We have prototyped the U[1]-chip, which is shown in Fig. 3. The size is 3 cm x 3 cm. The hardware design of U[1]-chip is very simple, where the main components are a microprocessor (Renesas H8S/2215U) and a programmable RF transceiver (Chipcon CC1020). We choose to operate U[1]-chip at 429.25 – 429.7375 MHz that falls into an unlicensed low-power service band for radiophones and data transmission equipment in Japan. This frequency band is divided into 40 channels: 1 control channel and 39 data channels. FSK modulation is used. Data rate is 4.8 kbps and the expected communication range is set at 10 m.

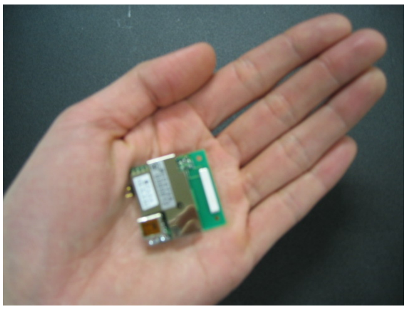

Fig. 3 U[1]-chip

3.1 Speed of device discovery

We have conducted an experiment to determine the speed of device discovery. A U[1]-chip is used as a master, whereas some other U[1]-chips are placed next to the master, acting as slaves. A 'Search' command is sent from the master and the time taken for all the slaves to respond is recorded. The results are shown in Fig. 4, which shows that as many as 12 slaves can be discovered within 2 seconds. This means that in average it takes only 167 ms to discover one slave. In the experiment, all the devices are put very close to each other, i.e. within 10 cm from each other.

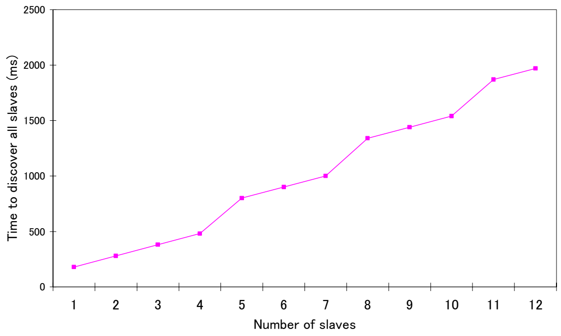

Fig. 4 Speed of device discovery

When a master identifies and specifically targets a slave by sending the latter a service request, a response will be received from the latter in 500 ms (the minimum result of 10 trials).

3.2 Power characteristics

U^1-chip is powered by GP's NiMH rechargeable batteries, GP850, rated at 850 mAh. At normal operating mode, the H8S/2215 microprocessor draws 27 mA of current [22]. When transmitting, the CC1020 RF transceiver draws 16.2 mA (to produce RF output power of 0.0 dBm) [23]. Thus the batteries can run for slightly less than 20 hours, assuming that a U^1-chip keeps transmitting without stop during that period. In receiving mode where CC1020 draws 19.9 mA of current, the batteries can last for about 18 hours.

4. Application

Instant service applications can be developed and run on the external devices where U^1-chips are connected to. Applications can also be developed on the slave U^1-chip. In this case the slave U^1-chip itself functions as service provider. Nevertheless, only simple application can run on U^1-chip due to the limitation of memory space. The H8S/2215 microprocessor has only 16 Kbytes RAM, where 15328 bytes of it is currently used by the U^1-chip operating system and communication program, leaving only 1056 bytes free for usage.

Here we show a very simple application that demonstrates the basic capabilities of U^1-chip. The application presented here is by no means a complete one, nor is it a commercial one. The variety of applications that can be developed on U^1-chip is virtually limited only by imagination. For the ease of application development, until now the application has been developed on PC, although it is possible to connect U^1-chip to a PDA or a mobile phone and to develop applications on these different types of user terminal.

The following is an item tracking application, which is potentially useful in environments where some common items are shared by a group of people. An example is books or some software installation CDs in an office environment. The usual practice is a log book is used to keep track of these common items, i.e. whoever has borrowed the item should write down his/her name on the log book. The fact is that this practice usually does not last very long. After some time people tend not to record their names on the log any more, and people have weak intention of seriously enforcing this practice. As a result, it is very hard for a user to trace where the item has gone if it is not in the place that it should be.

Motivated by this inconvenience, we invent the following item tracking application. Imagine a user terminal that is connected to a U^1-chip, which functions as a tracker. Items to be tracked are tagged with slave U^1-chips. Fig. 5 depicts how they function.

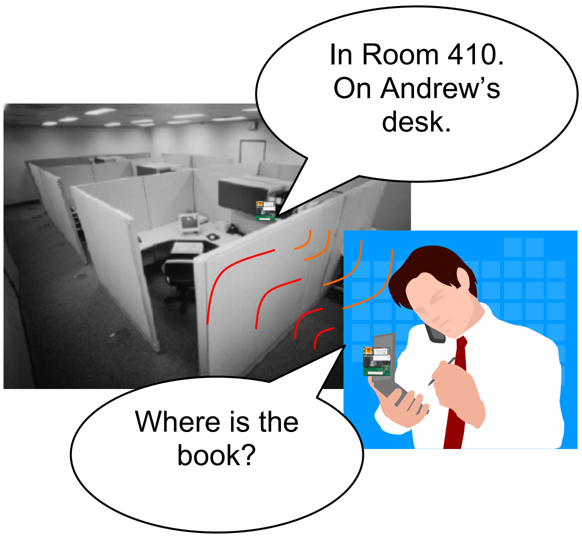

Fig. 5 Item tracking service

On the terminal, the user can see a very simple presentation of the U^1-chip application, as shown in Fig. 6. When the user presses the SEARCH button, the search process begins. All the U^1-chips in the vicinity reply by including their respective ID, service class (SVC), and the details about the service into their responses. Fig. 7 shows that two service provisioning U^1-chips have been discovered: a book tracking service and a CD tracking service. The name of the item, its borrower's name, and its current location are displayed. Based on this information, the user can track the item efficiently. In Fig. 7, the two U^1-chips that are tagged to a book and a CD respectively have not been associated with any information.

A user does not only read information from the service providers, he/she can also write information to them. Suppose the user now wants to keep the book with him/her. After clicking the 'Book Tracking Service' button, the user is prompted to key in the new information about the book. In Fig. 8, the user specifies that the name of the book is 'Computer Networking', the name of the borrower is 'Andrew', and the location of the book will be in 'Room 410'. When it is done and the 'OK' button is clicked, the new information is sent to the respective U^1-chip, which then returns a reply acknowledging the client U^1-chip. The application view now shows the updated information (Fig. 9). The above processes can be repeated. The next person who wants to borrow the same book will simply repeat the process by querying the slave U^1-chip using its own one. Then he/she will find out that Andrew has kept the book 'Computer Networking' with him in Room 410. After getting the book, he/she will change the necessary information accordingly.

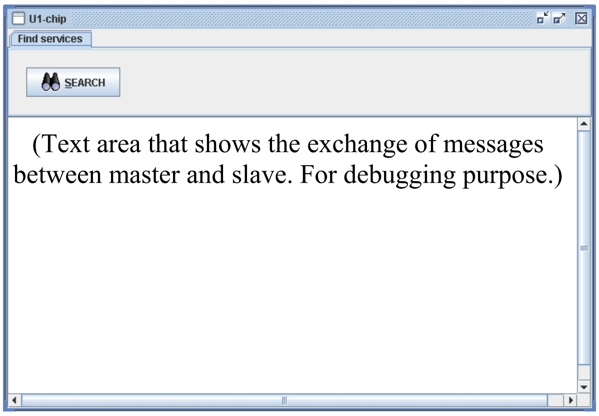

Fig. 6 Initial view

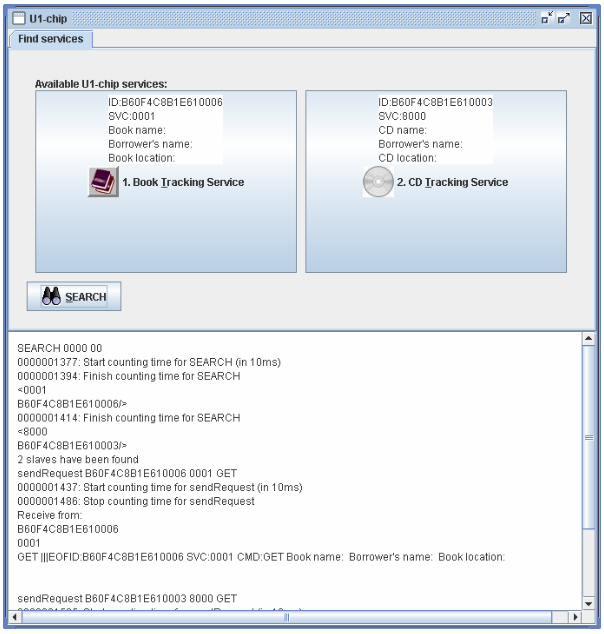

Fig. 7 Services found

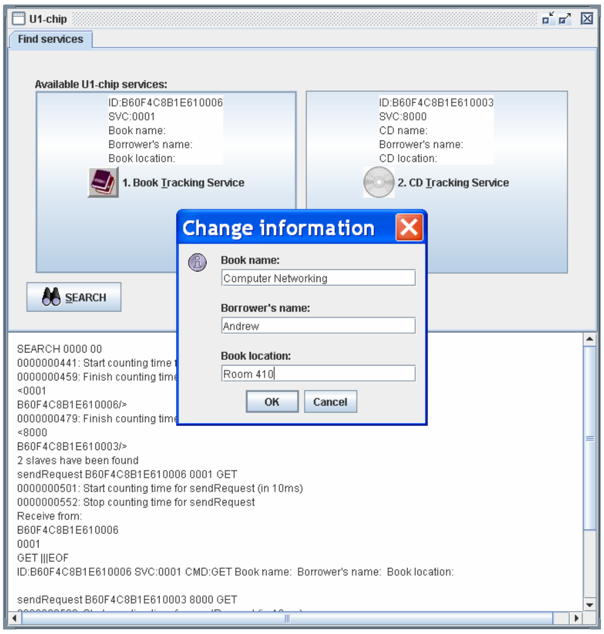

Fig. 8 Changing the service information

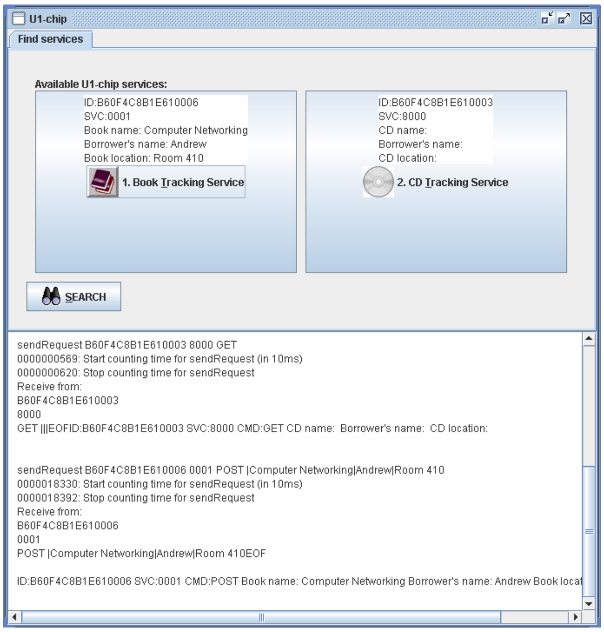

Fig. 9 Updated information

5. Conclusion

In this paper, we have shown a new potential service in the ubiquitous computing era which we name as instant service. To deliver instant service effectively, we have proposed and prototyped the U[1]-chip wireless communication modules. The idea and design of U[1]-chip have been illustrated, followed by the demonstration of an instant service application implemented on U[1]-chip. We have also measured the device discovery time and proven that we can accomplish fast device discovery in spite of low data rate. Currently we keep researching and expanding the capabilities of U[1]-chip. We consider a multi-clients environment where a master-to-master communication model is the interesting new dimension that we are exploring.

Acknowledgement

This work is supported by Ministry of Public Management, Home Affairs, Posts and Telecommunications.

References

1. UPnP Forum. http://www.upnp.org

2. Jini.org. http://www.jini.org/

3. Service Location Protocol (SLP). http://www.ietf.org/html.charters/svrloc-charter.html, 2001.

4. HAVi Consortium. http://www.havi.org/

5. Salutation Consortium. http://www.salutation.org/

6. Cooltown. http://www.cooltown.com/cooltown/

7. Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, and Randy H. Katz, "An Architecture for a Secure Service Discovery Service," *The Annual International Conference on Mobile Computing and Networking (Mobicom)* (1999).

8. Gribble et al., "The Ninja Architecture for Robust Internet-Scale Systems and Services," *The International Journal of Computer and Telecommunications Networking*, Vol. 35, No. 4, (2001).

9. William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, and Jeremy Lilley, "The design and implementation of an intentional naming system," *Proc. ACM Symposium on Operating Systems Principles* (1999).

10. Todd D. Hodes, Randy H. Katz, "Composable Ad hoc Location-based Services for Heterogeneous Mobile Clients," *Wireless Networks*, Vol. 5, No. 5, pp. 411 – 427, (1999).

11. Krumm J. and Hinckley K., "The *NearMe* Wireless Proximity Server," *The 6th International Conference on Ubiquitous Computing (UbiComp)* (2004).

12. Chiara Petrioli, Stefano Basagni, and Raffaele Bruno, "A Performance Comparison of Scatternet Formation Protocols for Networks of Bluetooth Devices," *IEEE First International Conference on Pervasive Computing and Communications (PerCom)* (2003).

13. Poole I., "What exactly is . . . ZigBee?" *IEE Communications Engineer*, Vol. 2, No. 4, pp. 44 – 45 (2004).

14. ZigBee Alliance. http://www.zigbee.org/

15. Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest and Daniel W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," *Security in Pervasive Computing, Lecture Notes in Computer Science*, Vol. 2802, pp. 201 - 212 (2003).

16. Hinckley K., "Bumping Objects Together as a Semantically Rich Way of Forming Connections between Ubiquitous Devices," *The 5th International Conference on Ubiquitous Computing (UbiComp)* formal video program (2003).

17. Hush D. R. and Wood C., "Analysis of Tree Algorithms for RFID Arbitration," *Proceedings of the IEEE International Symposium on Information Theory*, pp. 107 (1998).

18. Y. Taguchi, S. Saruwatari, H. Morikawa, T. Aoyama, M. Hasegawa, and M. Inoue, "U[1]-chip: Wireless Communication Module for Fast Service Discovery," *The 6th International Conference on Ubiquitous Computing (UbiComp)* poster (2004).

21. The Official Bluetooth® Wireless Info Site. http://www.bluetooth.com/

22. H8S/2215 Group Hardware Manual: Renesas 16-Bit Single-Chip Microcomputer - H8S Family/H8S/2200 Series. http://www.renesas.com/

23. CC1020 Data Sheet, edition 1.6. http://www.chipcon.com/