Proceedings of the 2005 International Conference on Active Media Technology (AMT2005)

May 19-21, 2005
Kagawa International Conference Hall,
Takamatsu, Kagawa, Japan

Editors
Hiroyuki Tarumi, Yuefeng Li, and Tetsuya Yoshida

Sponsored by
IEEE Systems, Man, and Cybernetics Society
Information Processing Society of Japan
Kagawa University

Proc. of Int'l Conf. on Active Media Technology (AMT 2005),
pp.423–428 (May 2005).

A Rule-Based RFID Tag System Using Ubiquitous Chips

Tomoki Yoshihisa
Osaka University, Japan
yosihisa@ist.osaka-u.ac.jp

Yasue Kishino
Osaka University, Japan
yasue@ist.osaka-u.ac.jp

Tsutomu Terada
Osaka University, Japan
tsutomu@cmc.osaka-u.ac.jp

Masahiko Tsukamoto
Kobe University, Japan
tuka@kobe-u.ac.jp

Ryohei Sagara
Osaka University, Japan
sagara.ryohei@ist.osaka-u.ac.jp

Teruki Sukenari
NEC Corporation, Japan
t-sukenari@cj.jp.nec.com

Daigo Taguchi
NEC Corporation, Japan
d-taguchi@bp.jp.nec.com

Shojiro Nishio
Osaka University, Japan
nishio@ist.osaka-u.ac.jp

Abstract

Because of the recent development of Radio Frequency Identification (RFID) technologies, various systems for RFID tags have been proposed. Since RFID tags only have a simple function, i.e., sending data, they can be available for various purposes. Accordingly, by customizing RFID tag systems, we can expand their applications. However, previous systems have been usually proposed for one special purpose only.

In this paper, we propose a rule-based RFID tag system using ubiquitous chips. Our previously proposed ubiquitous chips are rule-based I/O control devices. By applying rule-based principles, we can easily customize the RFID tag system and can construct flexible, scalable, and easy exploitable systems.

1. Introduction

Recently, there has been dramatic development of Radio Frequency Identification (RFID) technologies. RFID technologies deliver wireless communication with small tags, and various systems for these tags have been proposed. Since RFID tags only have a simple function, i.e., the sending of data, they can be used for various purposes. For example, suppose a case in which each employee has his/her own RFID tag and sets an RFID tag reader at the entrance of an important office, as shown in Figure 1. By identifying who passes the entrance, we can automatically check whether he/she has a license to enter the office. If the system can control the open-close motion of the entrance door, it can also be used for other purposes, e.g., shutting out outsiders, closing the door at night, and so on.

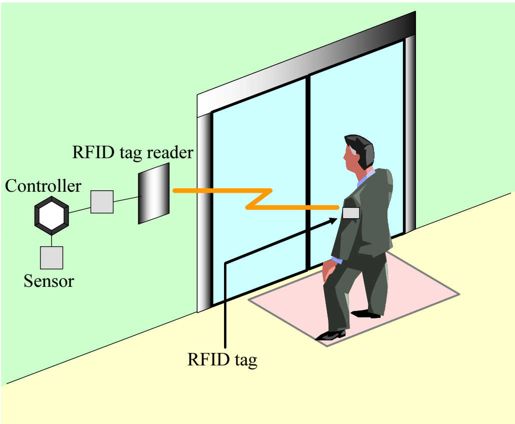


Figure 1. An RFID tag system

Since the applications of RFID tag systems are extensive, it is important for them to be customized. In the above case, the system must be customized so that it can control the door. If the system shuts out outsiders, it must detect outsiders. If the system closes the door at night, it must recognize that it is indeed night time. Hence, it must be customized so that it can control these sensors. Accordingly, by customizing RFID tag systems, we can expand their applications. Previous systems were usually proposed for one special purpose, and although some of those systems can change their attached sensors and other devices, they must return to square one to use new sensors and devices. This causes long development time. Moreover, in the case where component devices are embedded, it is difficult to customize them[12].

In this paper, we propose a rule-based RFID tag system. By applying rule-based principles, we can easily customize the RFID tag system and can construct flexible, scalable, and easy exploitable systems. The proposed system is built with ubiquitous chips[10]. Our previously proposed ubiquitous chips are rule-based I/O control devices, and since they are suitable for establishing a rule-based system, we use them here.

The remainder of this paper is organized as follows. In Section 2, we present several related works. In Section 3, we propose a rule-based system for RFID tags, and in Section 4, we describe a prototype of our proposed system. We present several applications of our proposed system in Section 5, and finally, we conclude the paper in Section 6.

2. Related Works

There are several studies focusing on RFID.

Issues with RFID usage in ubiquitous computing are discussed in [4]. A vision of ubiquitous computing is to provide various services with seamlessly integrated computers[11]. RFID tags have the potential to realize this vision. However, RFID tags have some problems such as tag collisions and tag detuning. That paper describes such issues and a primer of RFID technology. Privacy issues in RFID tags are discussed in [5].

A method to infer interactions with RFID tagged items is described in [3]. The method uses RFID readers to detect interactions such as the lifting up and pulling of items. Since the amount of energy transmitted from an RFID tag to the RFID reader depends on the distance and angle between them, the method can work.

A system that generates reminders by tagging items with RFID tags is proposed in [2]. In this system, all items users deem important are tagged and RFID tag readers are present at each of the locations such as home, car, company, and so on. Users carry their small personal server and their wristwatch user interface. The personal server contains some lists of items that are required for each location. By communicating with RFID tag readers, the personal server can recognize items with the user and can track the last known locations of tagged items. For example, if the user leaves home but forgets his/her cell phone, he/she will receive an alert from the wristwatch about the missing phone as he/she reaches the car. Since the personal server tracks the locations of the phone, the user can recognize the last locations of the phone. This system is convenient for reminders, but it cannot be customized easily.

On the other hand, some devices for ubiquitous computing environments such as ubiquitous chips are proposed. Smart-Its[1] is a small computing device consisting of two independent boards, a core board and a sensor board. Motes[6], MICA[8], and U-cube[7] are also small ubiquitous devices that are separated into two units. Although these devices can change attached sensors and other devices by changing the sensor unit, they must go back to square one to use new sensors and devices. This leads to long development times. Moreover, in the case where component devices are embedded, it is difficult to customize them.

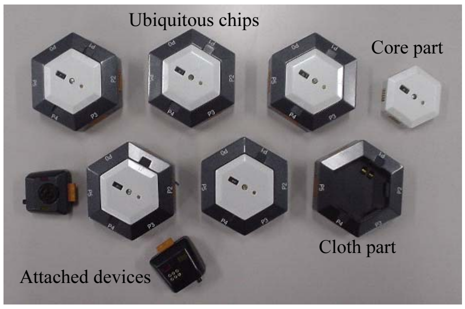


Figure 2. Ubiquitous chips and attached devices

3. Proposed System

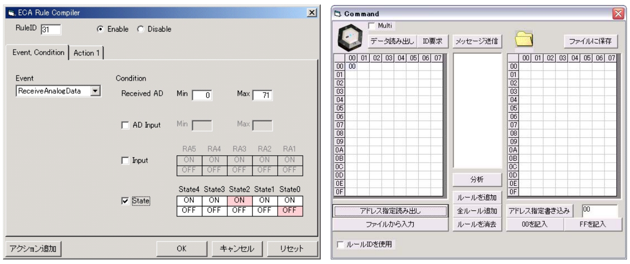
We propose a rule-based RFID tag system using ubiquitous chips. Since our previously proposed ubiquitous chips are rule-based devices, they are suitable for establishing the rule-based system. By attaching an RFID reader to ubiquitous chips, we construct the rule-based RFID tag system. The reason why we do not attach ubiquitous chips to RFID tags is that such an approach has the disadvantage of being incompatible with the billions of tags already in existence.

3.1. Ubiquitous Chip

This subsection provides brief description of ubiquitous chips. Detailed information is described in [10].

Figure 2 shows some ubiquitous chips, which are I/O control devices. Several devices such as switches, sensors, LED (Light Emitting Diode) can be attached to ubiquitous chips. The on/off states of an attached switch and inputs from a sensor are handled as “input signals,” while output operations such as turning on an LED are handled as “output signals.” By evaluating input signals and outputs signals to attached devices, ubiquitous chips control these devices.

3.1.1. Language Design Ubiquitous chips use ECA rules for the event-driven programming language. An ECA rule consists of the following three parts.



ECA rule editor ECA rule writer

Figure 3. ECA rule editor and writer

EVENT(E):	Occurring event
CONDITION(C):	Conditions for executing actions
ACTION(A):	Operations to be carried out

ECA rules have been used to describe the behaviors of active databases. An active database is a database system that carries out prescribed actions in response to a generated event inside/outside of the database. The main advantages of ECA rules are:

- as a consequence of their simplicity, we can program applications easily and intuitively.
- we can change a full/part of a program dynamically because behaviors are described as a set of ECA rules, and each rule is stored independently in the device.

For example:

E: Someone passes the entrance.
C: He/She has a license to enter the office.
A: Show the OK sign by turning on the LED.

If the system can control the open-close motion of the entrance door, by changing the rule, the system can be used for other purposes; it is not necessary to change ubiquitous chips. In the case of shutting out outsiders, the rule is changed to:

E: Someone passes the entrance.
C: He/She does not have a license to enter the office.
A: Close the door.

In this way, by employing ECA rules, the system can be easily customized without changing ubiquitous chips.

Moreover, we implemented an ECA rule editor and an ECA rule writer because it is difficult to describe ECA rules directly[9]. By using these tools, we can easily describe ECA rules and transport them to ubiquitous chips.

3.1.2. Communication Ubiquitous chips have serial ports to communicate with other ubiquitous chips and devices. The format for sending data to ubiquitous chips is shown in Figure 4. The data format consists of three parts: the header, the command, and the footer. For stable communication, we use the header and the footer, which indicate the start and the end of the data. The command is the context of the data.

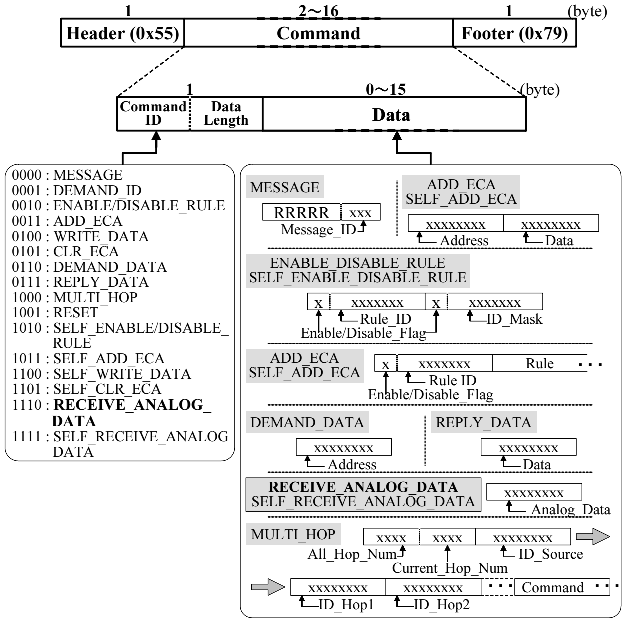


Figure 4. Format for sending data to ubiquitous chips

The command ID, which is the first four bits of the command, indicates the type of command, while the remaining four bits of the first byte is the data length. In the case of sending MESSAGE-1 to a ubiquitous chip, the command ID is 0, the data length is one byte, and the message ID is 1. In this case, the data becomes 0x55, 0x01, 0x01, 0x79 in binary coding. The message ID is only three bits long due to binary coding of the ECA rules.

By using RECEIVE_ANALOG_DATA command, we can send any data of one byte, and ubiquitous chips can evaluate the received data by ECA rules. For example, suppose the case where a ubiquitous chip has the following rule:

E: Receive a RECEIVE_ANALOG_DATA command.
C: The received data is less than 0x70.
A: Open the entrance door.

If the ubiquitous chip receives 0x55, 0xE1, 0x01, 0x79, the entrance door is opened.

3.2. Required Components

To establish an easily customizable RFID tag system using ubiquitous chips, the following components are required.

- To sense various contexts such as illumination, temperature, and humidity, sensing devices that can be handled by ubiquitous chips via input signals are required.

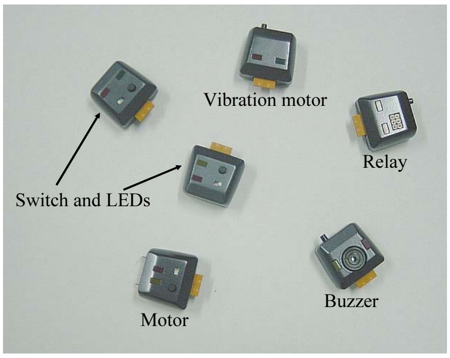


Figure 5. Input devices for ubiquitous chips

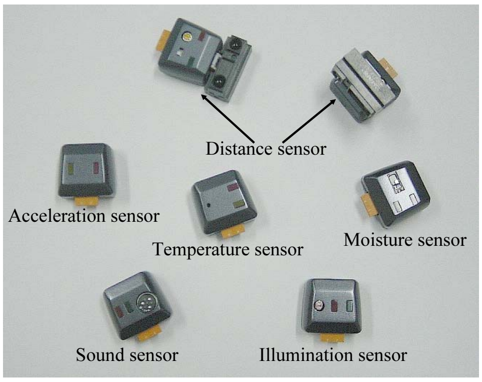


Figure 6. Output devices for ubiquitous chips

- To interact with the real world, output devices such as motors and buzzers are required.
- To read ID from RFID tags, RFID tag readers that can communicate with ubiquitous chips are required.

To address the first requirement, various input devices shown in Figure 5 are available. These input devices are attached to ubiquitous chips. Their electric power is supplied from ubiquitous chips. Since they are easily attached and removed, we can change input devices flexibly. For example, the signal of the illumination sensor turns off when it is dark and on when it is bright. Hence, by using the illumination sensor, ubiquitous chips can recognize whether it is night or day.

For the second requirement, some output devices shown in Figure 6 are available. These output devices can be changed just as flexibly as the input devices. For example, the motor device can be applied to open and close a door.

For the last requirement, due to the recent proliferation of RFID systems, we can use various RFID systems. The RFID readers of these systems usually have a serial port

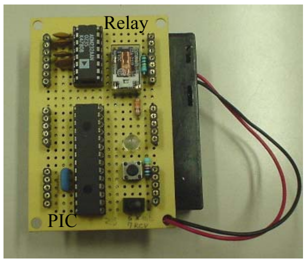


Figure 7. Conversion Module

and can communicate with computers via the serial port. However, since these RFID readers do not support the data format of ubiquitous chips, they cannot communicate with ubiquitous chips only by connecting to ubiquitous chips directly. Thus, to convert the data from the RFID reader into the ubiquitous chips' format, we have to develop a conversion module.

3.3. Conversion Module

Figure 7 shows a prototype of the conversion module. The module is controlled by PIC (Peripheral Interface Controller) by Microchip Corp. The module has two serial ports. By changing these serial ports by the relay, it can communicate with the RFID reader and the ubiquitous chip. The module receives the data from the RFID reader and converts the data into the ubiquitous chips' format. Since ubiquitous chips can evaluate the data sent by using the RECEIVE_ANALOG_DATA command, the module sends the ID to the ubiquitous chip using that command.

The function of the conversion module is only to convert the data; hence, if the RFID reader supports the ubiquitous chips' format, the module is not required.

4. Prototype

Figure 8 shows a prototype of our proposed system. We use an RFID reader and RFID tags made by Texas Instruments (RI-K2A-001A). The RFID reader has an antenna to read RFID tags, and it that it detects. The reader is connected to a ubiquitous chip through a conversion module. The conversion module converts the ID received from the RFID reader into the ubiquitous chips' format. In the prototype, since the system can distinguish all IDs by just the one byte of the ID, the conversion module sends that one byte of the ID to the ubiquitous chip with the RECEIVE_ANALOG_DATA command. For example, if the RFID finds an RFID tag for which the ID is 0x000123456789 and sends the ID to the conversion module, the conversion module sends 0x55, 0xE1, 0x89, 0x79 to the ubiquitous chip. However, by sending RE-

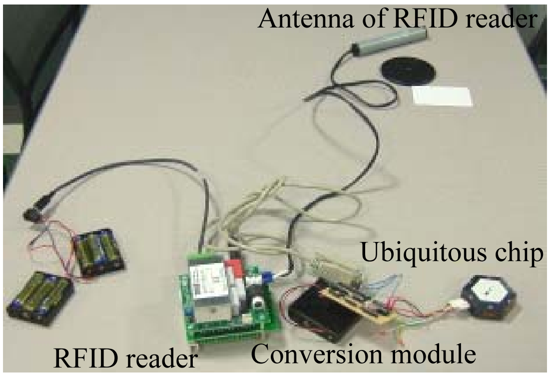


Figure 8. Prototype of our proposed system

RULE 1	
E:	RECEIVE_ANALOG_DATA
C:	0x00 ≤ RECEIVED_DATA ≤ 0x70
A:	O1 = ON, TIMER1(5sec)
RULE 2	
E:	RECEIVE_ANALOG_DATA
C:	0x71 ≤ RECEIVED_DATA ≤ 0xFF
A:	O1 = OFF
RULE 3	
E:	TIMER
C:	TIMER_ID = 1
A:	O1 = OFF

O1: LED

Table 1. ECA rules for proposed system(1)

CEIVE_ANALOG_DATA command several times, the conversion module can send an ID of more than one byte if need be.

5. Applications

First, we introduce a simple application of our proposed system. By customizing the system, we can add several functions to it. In this section, to explain behaviors of ubiquitous chips in detail, we show more detailed rules than those given in previous sections.

Suppose the case where all employees have their own RFID tag, and an RFID tag reader is set at the entrance of an important office as mentioned in Section 1. By identifying who passes the entrance, we can automatically check whether a person has a permit to enter the office. The ID of employees who do have a permit to enter the office is less than 0x71 and that of employees who do not

RULE 1	
E:	RECEIVE_ANALOG_DATA
C:	0x00 ≤ RECEIVED_DATA ≤ 0x70, I1 = ON
A:	O1 = ON, TIMER1(5sec)
RULE 2	
E:	RECEIVE_ANALOG_DATA
C:	0x71 ≤ RECEIVED_DATA ≤ 0xFF, I1 = ON
A:	O1 = OFF
RULE 3	
E:	RECEIVE_ANALOG_DATA
C:	0x00 ≤ RECEIVED_DATA ≤ 0x40, I1 = OFF
A:	O1 = ON, TIMER1(5sec)
RULE 4	
E:	RECEIVE_ANALOG_DATA
C:	0x41 ≤ RECEIVED_DATA ≤ 0xFF, I1 = OFF
A:	O1 = OFF
RULE 5	
E:	TIMER
C:	TIMER_ID = 1
A:	O1 = OFF

I1: Illumination sensor
O1: Motor

Table 2. ECA rules for proposed system(2)

is more or equal. Table 1 shows the ECA rules for this application. When the RFID reader detects an employee, it sends his/her ID to the ubiquitous chip using the RECEIVE_ANALOG_DATA command. If received data is less than 0x71, the actions of RULE 1 are executed. Since an LED is attached to Output1 (O1), the LED is brightened by turning O1 on. Also, TIMER 1, whose interval is 5 seconds, is set. When the timer expires, O1 is turned off. Therefore, the LED has been brightened for 5 seconds. If the received data is equal or more than 0x71, O1 is turned off. In this way, by watching the LED, we can check the license status.

In the case where the company extends the system so that the ubiquitous chip can control the open-close motion of the entrance door by adding a motor device, the system must be customized. In this case, we can control the open-close motion without changing rules only by replacing the LED with the motor.

Next, by customizing the system, we change the system in order to shut out employees at night whose ID is less than 0x41. To recognize whether it is night or not, illumination sensors are attached to the ubiquitous chip. In this case, the rule is modified as in Table 2. The illumination sensor is attached to Input 1 (I1) and the motor is attached to O1. Rules executed when the ubiquitous chip receives the RECEIVED_ANALOG_DATA command are different at daytime and nighttime. During daytime, i.e., when I1 is turned on, RULE 1 or RULE 2 is executed and employees whose ID is less than 0x71 can enter the office. During night, i.e., when I1 is turned off, RULE 3 or RULE 4 is executed and employees whose ID is less than 0x41 can enter the office.

This system can also be used for other applications. For example, checking the time to ring in and out by recording the time, and so on.

In this way, our proposed system can be customized easily.

6. Conclusion

Recently, various systems for RFID tags have been proposed. Since applications of RFID tag systems are extensive, it is important for them to be customized. However, previous systems were usually proposed for only one special purpose.

In this paper, we proposed a rule-based RFID tag system using ubiquitous chips. By applying rule-based principles, we can easily customize the RFID tag system. Our previously proposed ubiquitous chips are rule-based I/O control devices. Since they are suitable for establishing the rule-based platform, we used them. Moreover, we showed several application examples constructed by the system.

In the future, we plan to construct networking functions, such as routing, between ubiquitous chips. Moreover, in the current development environment, it is still rather difficult for users to customize the system and to develop larger systems composed of hundreds of ubiquitous chips. Therefore, we plan to provide new development tools for end-user programming and larger systems.

Acknowledgement

This research was supported in part by The 21st Century Center of Excellence Program "New Information Technologies for Building a Networked Symbiotic Environment" and in part by Grant-in-Aid for Scientific Research (B) numbered 15300033 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] M. Beigl and H. Gellersen, "Smart-Its: An Embedded Platform for Smart Objects", Smart Objects Conference (sOc) 2003.

[2] G. Borriello, W. Brunette, M. Hall, C. Hartung, and C. Tangney, "Reminding About Tagged Objects Using Passive RFIDs", In Proc. of the 6th Int. Conf. on Ubiquitous Computing (UbiComp 2004), pp. 36–53, 2004.

[3] K. P. Fishkin, B. Jiang, M. Philipose, and S. Roy, "I Sense a Disturbance in the Force: Unobtrusive Detection of Interactions with RFID-tagged Objects", In Proc. of the 6th Int. Conf. on Ubiquitous Computing (UbiComp 2004), pp. 268–282, 2004.

[4] C. Floerkemeier and M. Lampe, "Issues with RFID Usage in Ubiquitous Computing applications", In Proc. of the 2nd Int. Conf. on Pervasive Computing (PERVASIVE 2004), pp. 188–193, 2004.

[5] D. Henrick and P. Muller, "Trackling Security and Privacy Issues n Radio Frequency Identification Devices", In Proc. of the 2nd Int. Conf. on Pervasive Computing (PERVASIVE 2004), pp. 219–224, 2004.

[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors", In Proc. of the 9th Int. Conf. on Architectural Support for Programming Languages and Operating Systems, pp. 93–104, 2000.

[7] Y. Kawahara, M. Minami, S. Saruwatari, H. Morikawa, and T. Aoyama, "Challenges and Lessons Learned in Building a Practical Smart Space", In Proc. of the 1st Annual Int. Conf. Mobile and Ubiquitous Systems (MobiQuitous 2004), pp. 213–222, 2004.

[8] MICA, http://www.xbow.com/products/Wireless_Sensor_Networks.htm

[9] M. Resnick and S. Ocko, "LEGO/Logo: Learning Through and About Design", http://llk.media.mit.edu/papers/1991/11.html.s.

[10] T. Terada, M. Tsukamoto, K. Hayakawa, T. Yoshihisa, Y. Kishino, A. Kashitani, S. Nishio, "Ubiquitous Chip: a Rule-based I/O Control Device for Ubiquitous Computing", In Proc. of the 2nd Int. Conf. on Pervasive Computing (PERVASIVE 2004), pp. 238–253, 2004.

[11] M. Weiser, "The Computer for the Twenty-first Century", Scientific American, Vol. 265, No. 3, pp. 94–104, 1991.

[12] P. Wellner, E. Machay, R. Gold, M. Weiser, et al "Computer-Augmented Environments: Back To The Real World", Communications of the ACM, Vol. 36, No. 7, pp. 24–97, 1993.