International Journal of Smart Home
Vol. 8, No. 6 (2014), pp. 197-204
http://dx.doi.org/10.14257/ijsh.2014.8.6.19

Microcontroller Implementation of Rule-based Inference System for Smart Home

BonJae Koo, Young Soo Park and SungHyun Yang*

Department of Electronic Engineering, Kwangwoon University, Seoul, Republic of Korea
shyang@kw.ac.kr

Abstract

Recently, the development of Machine to Machine (M2M) communication has been largely accomplished in a variety of fields including smart home. In M2M communication, the role of sensor node is only limited to gather data and send them to upper application layers. In this research, the limited role of the sensor node in traditional M2M communication is improved in order for the sensor node to make inference based on the sensor data. If the sensor node can make inference, it is possible to provide basic context-aware services within sensor node level. Therefore, in this paper, implementation of rule-based inference system on microcontroller for smart home is proposed.

Keywords: *Machine to Machine, Rule-based inference, Microcontroller, Smart home*

1. Introduction

Context-awareness system is a system which adapts its behavior accordingly depends on the gathered context data [1]. The interest and the importance of context-awareness technology is growing larger as consumer electronics become smarter and people want more comfortable and intelligent services. In general, M2M communication including the sensor node is essentially needed to support the context-awareness technology. The role of the sensor node is only limited to gather the data and send them to upper application layers [2]. However, microcontroller in the sensor node can be programmed to have context-awareness system of its own. One of the methods of implementing context-awareness system is to build a rule-based inference system. A rule-based inference is a method to represent certain knowledge for a given field and solve problems by use of the knowledge [3]. A rule-based inference needs three basic components to be functional: working memory, rule base, and inference engine [4]. There are several ways for the implementation of context-aware system: using microprocessor, microcontroller, and FPGA. In this case, microcontroller implementation of rule-based inference system for smart home is presented. Arduino open-source electronics platform which includes Atmega328 is used for this particular design. The Atmega328 in Arduino platform has been programmed to have a function as an inference system including working memory, rule base, and inference module in it. A number of sensors (Temperature sensor, humidity sensor, pressure sensor, gas sensor, ambient light sensor, IR sensors) are directly connected to the microcontroller. Bluetooth communication module will be attached to the microcontroller to deliver the result of the inference.

[*] Corresponding Author

ISSN: 1975-4094 IJSH

This paper demonstrates how to build the rule-based inference system on microcontroller. The rest of the paper is organized as follows. In section 2, design of the system including the details of working memory, rule base, inference engine is discussed. In section 3, implementation result is represented and discussed. The result of the inference is shown in the serial monitor in Arduino software. Finally, the conclusion including plans for the future work is summarized in section 4.

2. The Design of the System

In this section, the details of the system will be explained. Section 2 is divided into three sub sections: Working memory section, rule base section and inference engine section. The overall design of the inference system is shown in the Figure 1.

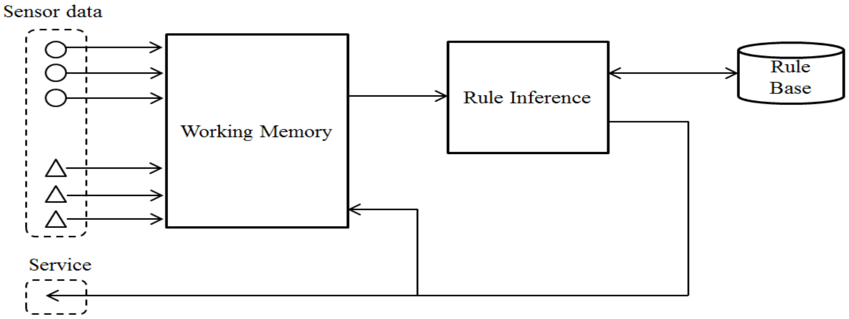
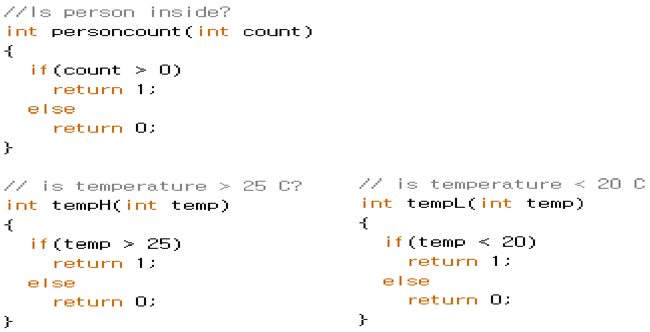


Figure 1. The Design of the Inference System

2.1 Working Memory

Working memory holds the set of data those are specifically needed to a problem being solved [5]. In this case, it principally means sensor data in the system as well as the result from the inference. Once the sensor data is detected, the data is processed and updated into proper Boolean values. Those Boolean values are saved in the array of the working memory in this system. The reason of having an array in this system is that we have to compare the new sensor data and the data in working memory. Once new sensor data is detected, it will be processed and compared with the data in the working memory. Size of the array, which is a size of the working memory, depends on the number of sensors it has on the system and the number of inference results which have an effect on different rules. Here is an example of how the data is processed.



```
//Is person inside?
int personcount(int count)
{
  if(count > 0)
    return 1;
  else
    return 0;
}
```

```
// is temperature > 25 C?
int tempH(int temp)
{
  if(temp > 25)
    return 1;
  else
    return 0;
}
```

```
// is temperature < 20 C
int tempL(int temp)
{
  if(temp < 20)
    return 1;
  else
    return 0;
}
```

Figure 2. Code for the Data Process

Person count returns 1 when the person is located inside home. The tempH returns 1 when the temperature is above 25 degree and tempL returns 1 when the temperature is below 20 degree. Those values are set for the service threshold.

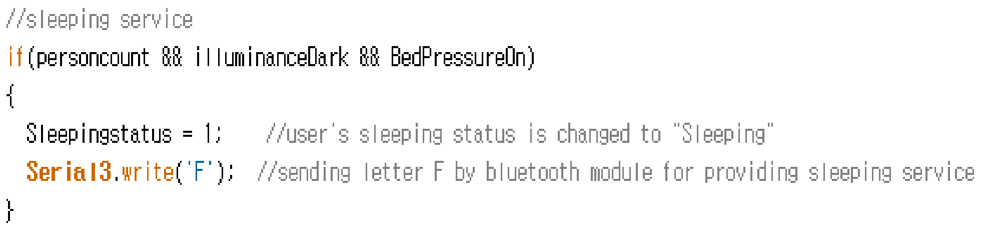
2.2. Rule Base

Rule consists of IF so called antecedent, promise or condition and THEN which is called consequent, conclusion or action [6]. Rule of inference system for this particular case can be defined as Figure 3.

```
Rule "GuardService"
if      "Person ?x "
              "LocatedIn ?x Outside"
              <and>
              "Mode Normal"
then          "Mode Guarding"
service       "GuardingService"
```

Figure 3. Rule for Crime Prevention in Home

In this example, GuardService rule is fired when the user is not located in the house and the previous status of the house is normal. Mode Guarding which comes after means the change of status of the house on the working memory. Providing service depends on the infrastructure of the house. In this case, GuardingService can be provided in a way of providing alarm when the motion is detected in house. In Figure 4, it is shown that the rule can be coded in the program.



```
//sleeping service
if(personcount && illuminanceDark && BedPressureOn)
{
  Sleepingstatus = 1;    //user's sleeping status is changed to "Sleeping"
  Serial3.write('F');  //sending letter F by bluetooth module for providing sleeping service
}
```

Figure 4. Code for Sleeping Service Rule

In this example, a rule for sleeping service is used. When the condition in the rule is entirely true, meaning the rule is fired, sleeping status is changed into 1 and saved in the working memory and inference engine uses this status to apply in different rules.

2.3 Inference Engine

As it is shown in Figure 5, the entire process of inference starts from the changes occurred in attributes. The changes trigger the inference engine to search the rule base until it finds the right antecedent of the rules [7]. There are two algorithms in rule based inference: forward chaining and backward chaining. Forward chaining algorithm is used in this system because of the nature of the system having continuous input changes [8]. As attributes in working memory changes, inference engine finds the proper rules to apply those attributes. If the proper rule is found and the condition of the rule is satisfied, the rule is fired and service is executed.

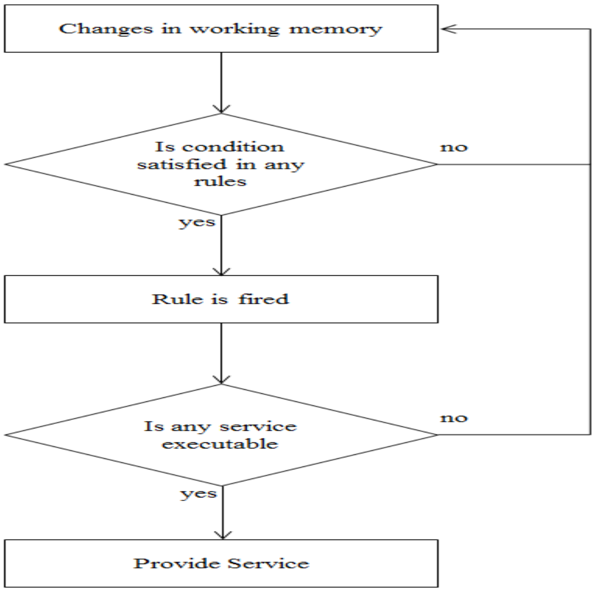

Changes in working memory
Is condition satisfied in any rules
no
yes
Rule is fired
Is any service executable
no
yes
Provide Service

Figure 5. Flowchart of how the Inference Engine Works

3. The Implementation of the System

In this section, the implementation of the proposed system is presented including the list of the rules and serial monitor output from Arduino software. Arduino open-source electronics platform is used for this particular design. In figure 6, Arduino Uno is connected to seven sensors including two IR sensor sets for both window and the door. Two IR sensor sets are used to make a counter for checking how many people are in the house and whether the person actually goes out or comes in.

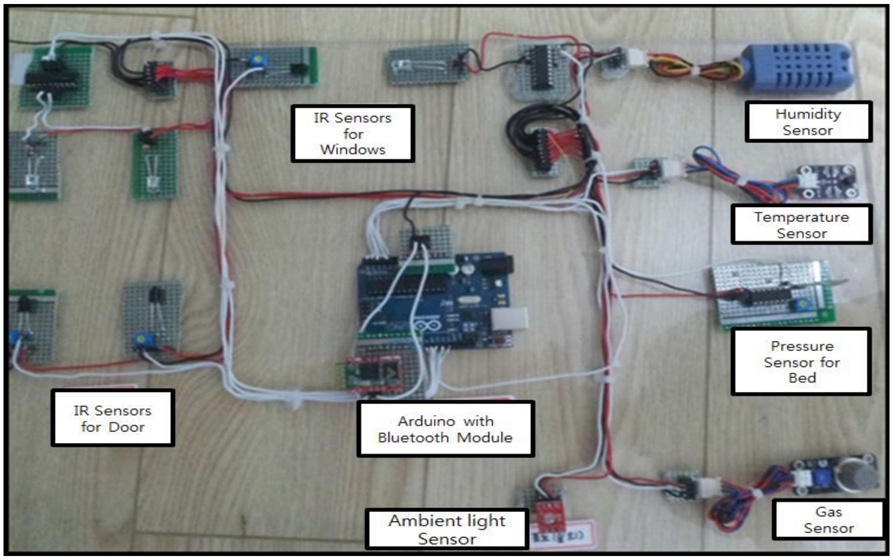


Figure 6. Experimental Setup for Implementation of the Inference Engine

3.1 Rules

Total 9 rules are defined in this system and the condition and the sensors used in each rule are shown in Table-I.

Table-I. Defined Rules with Condition and Sensors Used

Rule Name	Condition	Sensors used
Heating Service	Person_inside&&Temperature_Low	IR, Temperature
Air Conditioning Service	Person_inside&&Temperature_High	IR, Temperature
Humidifier Service	Person_inside&&Humidity_Low	IR, Humidity
Dehumidifier Service	Person_inside&&Humidity_High	IR, Humidity
Sleeping Service	Person_inside&&Illuminance_Dark &&BedPressure_On	IR, Ambient light, Pressure
Wakeup Service	Person_inside&&BedPressure_Off &&Sleepingstatus_True	IR, Pressure
Fire Alarm Service	Temperature_High&&Gas_On	Temperature, Gas
Crime Prevention Service	Person_outside&&Window_closed	IR
Normal Service	Person_inside&&ModeGuarding	IR

3.2 Test result

Test setup is made to using Atmega128 with Bluetooth module attached. LEDs, a buzzer, and a motor with motor driver are connected to Atmega128 and they represent different types of home appliances. Once the inference engine finds the proper service to be provided, the result of the inference engine is sent to the Atmega128 via Bluetooth communication. The inference procedure can also be shown in the serial monitor in the Arduino software.

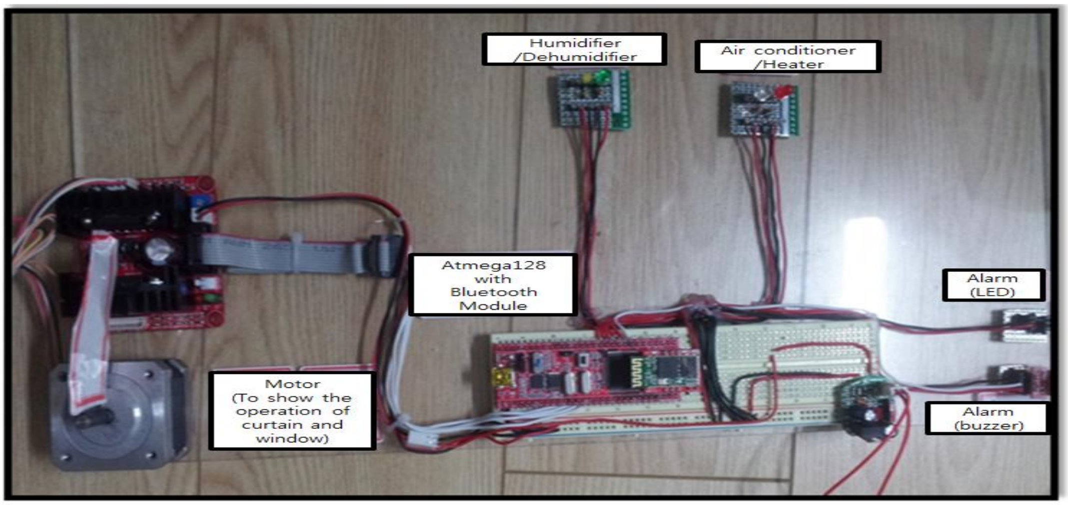
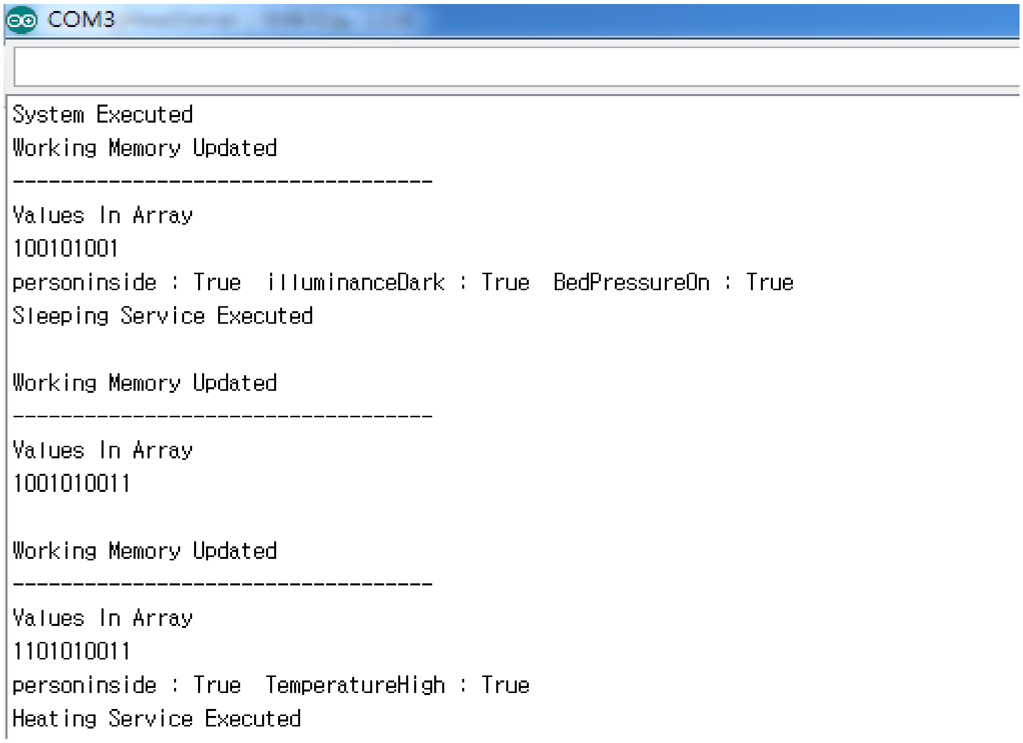

Humidifier /Dehumidifier
Air conditioner /Heater
Atmega128 with Bluetooth Module
Alarm (LED)
Motor (To show the operation of curtain and window)
Alarm (buzzer)

Figure 7. Test Setup with Atmega128

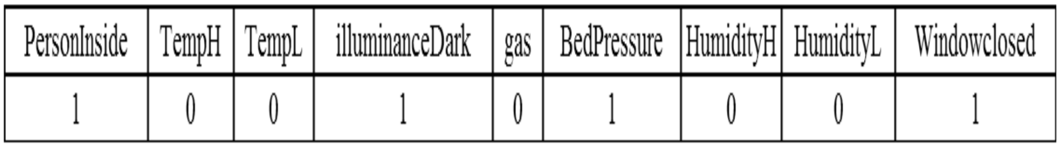


```
System Executed
Working Memory Updated
-----------------------------------
Values In Array
100101001
personinside : True  illuminanceDark : True  BedPressureOn : True
Sleeping Service Executed

Working Memory Updated
-----------------------------------
Values In Array
1001010011

Working Memory Updated
-----------------------------------
Values In Array
1101010011
personinside : True  TemperatureHigh : True
Heating Service Executed
```

Figure 8. Output of Inference Engine on Serial Monitor

Once system starts, the working memory is updated with current sensor value. In the output, Values in array shows the data in the working memory of the system. The meaning of the value is shown in the table-II. Sleeping service is executed because the condition on sleeping service rule and the sensor data are matched. When working memory is updated after the sleeping service rule is fired, the value 1 is added at the end of the array. It is showing that the person's current status is added in the working memory after the inference. The working memory after the sleeping service rule is executed is shown in table-III. When the services are executed the microcontroller sends the result to the Atmega128.

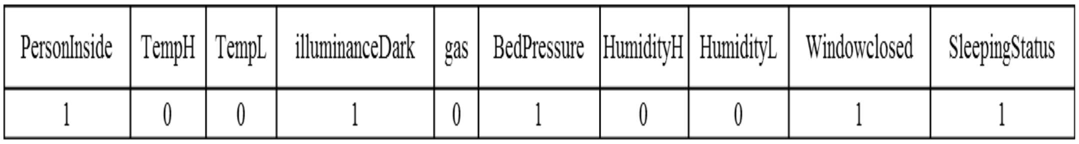
Table-II. Output Result of Initial State of Working Memory



PersonInside	TempH	TempL	illuminanceDark	gas	BedPressure	HumidityH	HumidityL	Windowclosed
1	0	0	1	0	1	0	0	1

When working memory is updated after the sleeping service rule is fired, the value 1 is added at the end of the array. It is showing that the person's current status is added in the working memory after the inference. The working memory after the sleeping service rule is executed is shown in table-III. When the services executed messages are shown in the serial monitor, the inference system sends the result to the Atmega128. When "Heating Service Executed" message is shown in serial monitor, one LED connected to Atmega128 is turned on and it indicates the heating service is successfully delivered and executed.

Table-III. Output Result of Working Memory after Sleeping Service is Executed



PersonInside	TempH	TempL	illuminanceDark	gas	BedPressure	HumidityH	HumidityL	Windowclosed	SleepingStatus
1	0	0	1	0	1	0	0	1	1

4. Conclusion

In this paper, a microcontroller based rule inference system for smart home is proposed and its implementation using Arduino platform is demonstrated. According to the successful implementation, the system is able to provide context-aware services based on rules set up in microcontroller and the sensor data. The possibility of providing context-awareness services based on M2M technology is shown. The context-awareness services can be provided without using upper application layers by using this system. The complexity and the variation in the number of the rules depend on the number of sensors connected to the system. This system can be configured in larger places by deploying multiple sensor nodes which bypass the sensor data to the microcontroller which has the ability to make inferences.

Even though the system demonstrated is successfully implemented and able to provide context-aware services, it is necessary to have user's feedback. As a future work, it is planned to have a feedback system and to provide services using the smartphone.. Moreover, once the connection between the system and smartphone is made, there are more services can be provided using sensors in smartphone.

Acknowledgements

This work was supported by the Industrial Strategic Technology Development Program (10041788, Development of Smart Home Service based on Advanced Context-Awareness) funded by the Ministry of Trade, Industry & Energy (MI, Korea).

References

[1] M. Baldauf and S. Dustdar, "A survey on context-aware systems."Int. J. Ad Hoc and Ubiquitous Computing, vol. 2, no. 4, **(2003),** pp. 263-277.

[2] J. R. Rosslin and T. H. Kim, "Review: context aware tools for smart home development", International Journal of Smart Home, vol. 4, no. 1, **(2010)**.

[3] N. L. Griffin, "A Rule-Based Inference Engine which is Optimal and VLSI Implementable", Proc. of the Tools for Artificial Intelligence, Architectures, Languages and Algorithms, **(1989)**, pp. 246-251.

[4] D. Merritt, "Building Expert Systems in Prolog", Springer Compass International 1st ed, **(1989)**, pp. 1-3.

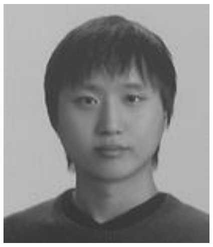
[5] M. Negnevitsky, "Artificial Intelligence: A Guide to Intelligent Systems", Addison Wesley, 2nd ed, **(2005)**, pp. 30-33.

[6] I. O. Folorunso, O. C. Abikoye, R. G. Jimoh and K. S. Raji, "A Rule-Based Expert System for Mineral Identification", Journal of Emerging Trends in Computing and Information Sciences, vol. 3, no. 2, **(2012)**.

[7] S. Russell and P. Norvig, "Artificial Intelligence: A modern approach", Prentice Hall, 3 ed, **(2009)**.

[8] P. Jackson, "Introduction to Expert Systems", Addison Wesley, **(1998)**, pp. 87-93.

Authors



BonJae Koo

He received his Bachelor of Engineering degree in Electrical Engineering from State University of New York at Stony Brook, New York, USA in 2011. Now he is pursuing his master's degree at Kwangwoon University, Seoul, and Republic of Korea. His main research interests are Microcontroller, Context-aware systems and Sensor Networks.

Young Soo Park

He received his B.S and M.S degree in Computer Science from the Institute of Information Technology and Graduate School of Kwangwoon University, Seoul, Republic of Korea, in 1994 and 1997. He completed his Ph.D. from Kwangwoon University in 2006. He is a Professor in Computer Engineering at the Institute of Information Technology of Kwangwoon University from 2011 to 2014. His main research interests are Software Engineering, Mobile Web Computing, and 3D Image Processing.

SungHyun Yang

He received his B.S. and M.S. degree in Electrical Engineering from Kwangwoon University, Seoul, Republic of Korea, in 1983 and 1987 respectively. He completed his Ph.D. from Kwangwoon University in 1993. He is a Professor in Electronic Engineering at Kwangwoon University, Seoul, Republic of Korea. He is a Director of the Ubiquitous Home Network Center, Kwangwoon University. He was a Research Scientist at Boston University from 1996 to 1998. He was Chairman of the Home Network Market Activation Section, Korean Association for Smart Home from 2007 to 2008. His main research interests are Digital logic, Embedded Systems, M2M, Next Generation Ubiquitous Home Networks, and Context-aware systems.