British Journal of Applied Science & Technology
15(5): 1-9, 2016, Article no.BJAST.25211
ISSN: 2231-0843, NLM ID: 101664541

SCIENCEDOMAIN *international*
www.sciencedomain.org

Representation of Raspberry PI Practice in Z Notation

Wen Jinjie[1*], Guo Yang[1] and Zhao Zhengxu[2]

[1]School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang, 050043, China.
[2]Shijiazhuang Tiedao University, Shijiazhuang, 050043, China.

Authors' contributions

This work was carried out in collaboration between all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2016/25211
Editor(s):
(1) Wei Wu, Applied Mathematics Department, Dalian University of Technology, China.
Reviewers:
(1) Chen Gao, Tufts University, USA.
(2) Kexin Zhao, University of Florida, USA.
(3) Lirong wang, State University of New York, USA.
Complete Peer review History: http://sciencedomain.org/review-history/14214

Original Research Article

Received 23rd February 2016
Accepted 2nd April 2016
Published 16th April 2016

ABSTRACT

The maker is the pioneer of open source, and the open hardware is the essential tools for the makers. As the typical representative of the open hardware, Raspberry Pi has been widely applied in open source since 2012. Z notation is a formal specification language based on the set theory and the first order predicate logic. So the Z notation can improve the reliability and robustness of the computer system using strict mathematical theory. This paper accomplishes a technical explanation of the concept of open hardware and the organization of the Raspberry pi via utilizing the Z notation. The Z notation of Raspberry pi is more precise and more systematic compared with the other informal specification. This research has great significance for the large-scale popularization of Raspberry pi and open hardware development.

Keywords: Raspberry pi; maker; Z notation; open hardware; formal.

**Corresponding author: E-mail: 474600137@qq.com;*

1. INTRODUCTION

The maker is a group that transforms the challengingoriginality into reality, especially in electronics, machinery, 3D printing and other fields. As an essential ingredient of the open culture, the open hardware constructs computer or electronic products in the mode same as the open software. Its aim is to encourage hardware source code (i.e., product design documents) can be shared in the association composed by hardware designers, producers and consumers. The open hardware is not only the development of the hardware design method, but also is the development of the innovative concept. This concept will contribute to promote the progress of computer technology in the long run.

The Raspberry pi is a micro-computer motherboard based on Advanced RISC Machines architecture [1]. The Raspberry pi not only has comprehensive functions, but also has the compatibility for the other electronic hardware. The conception of Raspberry pi design is to propose a novel tool for the makers to transform many ideas into reality. The Z notation is based upon set theory and mathematical logic. It owns the capability of accomplishing technical description logical relationships between set and set insides complicated system. As a result, the Z notation-based specification of Raspberry pi organizations can develop the knowledge of hardware interfaces within the Raspberry pi, and consequently advance the implementation of open hardware.

2. BACKGROUND

The Raspberry pi was launched by the raspberry pi non-profit foundation to promote cheap computer science education in 2012, the founder of raspberry pi named Eben Upton. The Raspberry pi is a series of credit card–sized single-board computers using ARM architecture and the system of Raspberry pi is based on Linux [1]. The users can connect the raspberries pi with TV, monitor, keyboard, mouse and other peripheral devices. The Raspberry pi has powerful functions including word processing, spreadsheet, media center and even games. What's more the Raspberry pi can also play 1080p HD video and has a strong system resources and interface resources.

Since the first Raspberry pi was launched in 2012, Raspberry pi became more and more popular among the global users and the circulation is more than expected. The statistical circulation of Raspberry pi from 2012 to 2015 is given by the figure below.

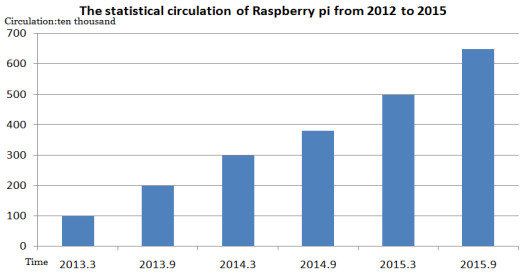

Fig. 1.1. The statistical circulation of Raspberry pi from 2012 to 2015

Because of the low price and the small volume, so the emergence of the Raspberries pi makes all creations having the potential to become a reality. At the latest World Makers Forum held in queens New York the artist named Sam Blanchard and the scientist from Virginia Tech named Kirk Cameron cooperated and composited a computer that consists of 256 pieces of Raspberry pi to display "The visual form of physical data " in September 2015. The Raspberry pi has a broad application prospect such as distributed computer cluster, monitoring systems, mobile base stations, remote monitoring, intelligent robot, etc.

Currently, as one of the most popular formal description language the Z notation is a formal language using "mathematical script" or "mathematical symbols" to describe the computer system [2-5]. It not only can be applied to describe the computer hardware system and especially suitable for the computer software system. The Z language describes "what to do" but not "How to do" and only gives a functional description of the target software system. In fact, Z language is just a set of mathematical symbols. The "program" written by Z language is an abstract design of computer software or hardware system [6]. So the content written by Z language is neither the computer program nor the code can be compiled on a computer. The content written by Z language can't run on a computer but understand and analyze for users. Through this content the users can comprehend the module, data type, procedure, function, object and class of the computer system and do some analysis, validations and improvements for the behavior of the computer system, structure and logic [7].

The mathematical foundation of Z language is predicate logic and set theory. The predicate logic is a kind of formal system applied to philosophy, linguistics, mathematics, computer science; it also has other names such as first-order logic, first-order predicate calculus, low order predicate or quantitative theory, etc. The set theory is a mathematical theory researching the whole made up of abstract objects, this whole contains some basic math concepts such as assemble, element, relationship, etc [8]. The predicate logic and set theory build mathematical objects formally with undefined terms and the elements in the set, thus forming the axiomatic mathematical foundations.

The Z language builds the mathematical model for the status characteristics and behavior characteristics of the software system using some known abstract features through the predicate logic and set theory, i.e. defines the state of the software system and the operation from one state to another state clearly [9]. The Z language describes the operations through the relationship between the pre-condition and the post-condition of state transition, the input variables and the output variables. There are four kinds of schemas; they are Initialization schema, State schema, Operator schema and Error schema. The Z schema consists of three parts are Schema name, Declarations and Predicate, which is to say that the schema is equal to the Declarations adds the Predicate.

A basic Z schema is shown as below:

SchemaName
Declarations
Predicate

The Z notation, then, is a mathematical language with a powerful structuring mechanism. In combination with natural language, it can be used to produce formal specifications. We may reason about these specifications using the proof techniques of mathematical logic. We may also refine a specification, yielding another description that is closer to executable code [10,11].

In conclusion, Z language describes the computer system with mathematical logic and theory model. At last it generates an abstract design. Usually we called the abstract design Z notation.

Z language has the following advantages compared with the informal languages:

First of all, the Z language is not the computer programming tools or programming system, it is a kind of design specification including a number of abstract mathematical theories such as sets, sequences, bags, relationships, functions, classes, objects etc, so the z language is a kind of mathematical language.

Secondly, the Z notation is not entirely similar with the ASCII texts or strings; it also includes the formal mathematical symbols and calculus of graphics. The content of Z notation cannot be complied and run, but can be used to do logical reasoning and perform mathematical calculations.

In addition, the Z notation not only contains the mathematical symbols but also uses natural language to define variables and comments. So the Z notation is easy to reading, writing and parsing.

Finally, the Z notation not only is rigorous, clear and no ambiguous but also can be verified using formal method software such as Z/EVES etc.

3. RASPBERRY PI NOTATION

When we describe any objective things using Z language, there are different kinds of describing grains we can choose. The higher we stand point of view, the higher level of abstract. With the decrease of the abstract degree, the Z notation will be more and more accurate. The first step in using Z language is initialization. The initialization is to initialize the system variables, which is give variables initial values. Because there is no operation of system state variables before initialization, so the initialization has no pre-state.

The structure of the raspberry pi system can be divided into two parts of hardware system and software system.

3.1 Hardware System

As a whole, the hardware system consists of host computer (central processing unit, internal memory), input interface, output interface, peripherals, storages and network communication equipment [12,13].

3.2 Software System

The software system mainly includes the operating system. In addition, the users can

install a variety of applications according to individual demands.

When we describe the Raspberries pi using Z language, we can define "host" as the set of computer mainframe and its type is "HOST"; "input" as the set of computer input devices and its type is "INTERFACE"; "output" as the set of computer output devices and its type is "INTERFACE". They are both the set with at least one and countable number of elements, so we can decorate them by "$\mathbb{F}_1$". The "host" includes CPU, GPU and memory, so the three objects can be abstracted as global variables in the Raspberries pi system. The "output" includes GPIO, RCA_VIDEO and AUDIO, so the three objects can be abstracted as output variables in the Raspberries pi system. The "input" includes power, USB, SD and network, so the four objects can be abstracted as input variables in the Raspberries pi system. According to the Z language syntax, we describe output variables with "!" and describe input variables with "?".The Z notation of the whole Raspberries pi system is as follows:

InitRaspberry Pi

host：$\mathbb{F}_1$ *HOST*
output：$\mathbb{F}_1$*INTERFACE*
input：$\mathbb{F}_1$*INTERFACE*
system：*SOFTWARE*

---

host =CPU∪GPU∪Memory
output=*GPIO∪RCA_VIDEO∪AUDIO*
input=*p*ower∪USB∪*SD*∪network
system=*Linux*|*Raspbian*|*RaspBMC*|*Pidora*|*XBMC*

We will involve two schemas in the description below. The first is "ΔSys≙[Sys, Sys′]", it includes pre-state and post-state. The second schema is "ΞSys≙[ΔSys|θSys=θSys′]", it includes pre-state and post-state, but the two states are equal.

There appeared a number of different versions with the development of Raspberries pi. The Raspberry pi 1 includes Model A, Model A+, Model B and Model B+, because of the different versions with different attributes, so we can map into a binary relationship from the versions to the attributes [14-16]. The different versions with different attributes are summarized in Table 4.1.

The host module of Raspberry pi includes CPU, GPU and Memory.

1. CPU

The central processing unit is the brain of the Raspberry pi. It can do mathematical operations, compare stuff and save stuff. The average processor of Raspberry pi 1 contains about 700,000,000 transistors.

2. GPU

The Graphic Processing Unit of the Raspberry pi is in charge of putting a picture on your screen. Its operational capability can reach 1G pixel per second and conform to the OpenGL ES 2.0 standard so can play 1080p Full HD video.

3. Memory

The Raspberry pi hardware has evolved through several versions that feature variations in memory capacity, and peripheral device support. The memory capacity of Model A and Model A+ is 256 MB, the memory capacity of Model B and Model B+ is 512 MB.

Table 4.1. The different versions with different attributes table

Version		Model A	Model A+	Model B	Model B+
Host	CPU	700MHz			
	GPU	OpenGL ES 2.0,1080p /1G pixel/s			
	Memory	256 MB	256 MB	512 MB	512 MB
Output	GPIO	26 pins	40 pins	26 pins	40 pins
	VIDEO	640×350—1920×1200			
	AUDIO	3.5 mm phone jack			
Input	Power	300 mA	200 mA	700 mA	600 mA
	USB	1	1	2	4
	SD Card	2G			
	Network	Wireless		Wired	

So the Z notation of the host of Raspberry pi is as follows:

Host

Ξ*InitRaspberry pi*
A,*A*+,*B*,*B*+:*version*
CPU：$\mathbb{N}_1$
Memory：*version*↔$\mathbb{N}_1$
GPU:：$\mathbb{F}_1$*host*

CPU=700*MHz*
dom *Memory* ={*A*,*A*+,*B*,*B*+}
ran *A*_ *Memory*=ran*A*+_ *Memory*=256*MB*
ran *B*_ *Memory*=ran*B*+_ *Memory*=512*MB*
GPU= OpenGL ES 2.0∧(1080p∨1 *G*pixel/*s*)

The output module of Raspberry pi includes GPIO, RCA_VIDEO and AUDIO.

1. GPIO

There are 26 pins in the general purpose input-output (GPIO) connector of Model A and B. There are 40 pins in the General purpose input-output (GPIO) connector of Model A+ and B+.

2. RCA_VIDEO

The video controller is capable of standard modern TV resolutions, such as HD and Full HD, and higher or lower monitor resolutions and older standard CRT TV resolutions. As shipped (i.e. without custom overclocking) it is capable of the following: 640×350 ; 640×480; 800×600; 1024×768; 1280×720; 1280×768; 1280×800; 1280×1024; 1366×768t; 1400×1050; 1600×1200; 1680×1050; 1920×1080; 1920×1200.

3. AUDIO

The output of audio analogs via a 3.5 mm phone jack in the Raspberry pi system.So the Z notation of the output module of Raspberry pi is as follows:

Output

Ξ*InitRaspberry pi*
GPIO: *version*↔$\mathbb{N}_1$
RCA_VIDEO:$\mathbb{N}\times\mathbb{N}$

♯(ran *A*_ *GPIO*)=26∨♯(ran *A*+_ *GPIO*)=40∨
♯(ran *B*_ *GPIO*)=26∨♯(ran *B*+_ *GPIO*)=40
640×350≤RCA_VIDEO≤1920×1200
AUDIO=3.5mm

The input module of Raspberry pi includes Power, USB, SD card and Network.

1. Power

The unit uses a Micro USB connection to power itself (only the power pins are connected - so it will not transfer data over this connection). A standard modern phone charger with a micro USB connector will do. But the different versions need different electric current.

2. USB

In order to connect additional devices to the Raspberry pi, so the Raspberry pi contains several USB interfaces that will allow multiple devices to be used. The Model A and Model A+ have only one USB interface, the model B has two USB interfaces and the model B+ has four USB interfaces.

3. SD Card

As the Raspberry pi has no internal storage or built-in operating system it requires an SD-Card that is set up to boot the Raspberry pi. The capacity of the SD card must be greater than 2GB.

4. Network

Though the model A and A+ do not have an Ethernet port, they can be connected to a network using a Wi-Fi adapter. On the model B and B+ the Ethernet port is provided by a built-in USB Ethernet adapter.

The Z notation of input module is shown as follows.

Input

Ξ*InitRaspberry pi*
Power: *version*↔$\mathbb{N}_1$
USB: *version*↔$\mathbb{N}_1$
SD_storage:$\mathbb{N}_1$
Network：*version*↔*type*

ran *A*_*Power*=300*mA*∨ran *A*+_*Power*=200*mA*∨
ran *B*_*Power*=700*mA*∨ran *B*+_*Power*=600*mA*
♯(ran*A*_*USB*)=1∨♯(ran*A*+_*USB*)=1∨♯(ran*B*_*USB*)=2∨
♯(ran*B*+_*USB*)=4
SD_storage≥2*G*
dom *Network* ={*A*,*A*+,*B*,*B*+}
ran *A*_ *Network*=ran *A*+_*Network*=Wireless∨
ran *B*_ *Network*=ran *B*+_*Network*=Wire*d*

4. AN APPLICATION CASE—PM2.5 INDICATOR

In this section, we will display the PM2.5 index of any city based on the Raspberry pi and prompt the condition of air using the combination of the LED and the buzzer alarm. We can obtain the PM2.5 index through the meteorological data released by the website, so can meet the demand of the PM2.5 indicator. The PM2.5 indicator consists of the Raspberry pi, the LED light, the buzzer alarm and the wireless receiver. The wireless receiver is used to receive the index of PM2.5. As the core of the PM2.5 indicator, Raspberry pi manages the logical processing of the index received through the wireless receiver. The LED indicates the PM2.5 index, and the buzzer alarm will raise the alarm when the PM2.5 index goes beyond the warning line [17]. The hardware structure of the PM2.5 indicator is shown in the figure below:

Fig. 4.1. The hardware structure of the PM2.5 indicator

Firstly, we can exploit the API of professional institution to obtain the date about the PM2.5.We use the API of the website (http://www.heweather.com/documents/api and http://www.heweather.com/documents/cn-city-list) in the following example. We will get the "cityid" and the "key" after registration. The personal "key" is "99ddea4ef47842d88f3b1878fd94be7f" and the "cityid" of Shijiazhuang is "CN101090101"

Registration

email?:*EMAIL*
city?:CITY
id!:*ID*
key!:*KEY*
emailkey: *EMAIL*↣*KEY*
cityid: *CITY*↣*ID*
*password*1?,*password*2?:*PASSWORD*
report!:*RESPONSE*

---

email?≠∅∧*city?*≠∅
*password*1?=*password*2?
key!=ran ((*email*?)◁*emailkey* ())
id!=ran ((*city*?)◁*cityid* ())
report!=Success

The Python program to get the PM2.5 index is shown below.

```
weather_url = 'https://api.heweather.com/x3/weather?cityid=CN101090101&
key=99ddea4ef47842d88f3b1878fd94be7f
 def get_pm25():
    global weather_url
    req = urllib2.Request(weather_url)
    resp = urllib2.urlopen(req)
    content = resp.read()
    if(content):
       weatherJSON = json.JSONDecoder().decode(content)
        #print(content)
        try:
            if weatherJSON['HeWeather data service 3.0'][0]['status'] == "ok":
                if weatherJSON['HeWeather data service 3.0'][0].has_key('aqi'):
                    print(weatherJSON['HeWeather data service 3.0'][0]['aqi']['city']['pm25'])
                    return int(weatherJSON['HeWeather data service 3.0'][0]['aqi']['city']['pm25'])
                else:
                    return -1
            else:
                return -1
```

The following code controls the LED and the buzzer alarm with the function “SAKS ()”. The function SAKS.ledrow.items[7].on() indicates the eighth LED light is on, the function SAKS.ledrow.off () indicates all the LED lights are out.

```
if __name__ == "__main__":
    while True:
        pm25 = get_pm25()
        if pm25 == -1:
            time.sleep(30)
            continue
        if pm25 >= 250:
            SAKS.ledrow.off()
            SAKS.ledrow.items[7].on()
            SAKS.buzzer.beepAction(0.05,0.05,3)
        if pm25 < 250:
            SAKS.ledrow.off()
            SAKS.ledrow.items[7].on()
        if pm25 < 115:
            SAKS.ledrow.off()
            SAKS.ledrow.items[6].on()
        if pm25 < 75:
            SAKS.ledrow.off()
            SAKS.ledrow.items[4].on()
        if pm25 < 35:
            SAKS.ledrow.off()
            SAKS.ledrow.items[4].on()
            SAKS.ledrow.items[5].on()
         #print (("%4d" % pm25).replace(' ','#'))
        SAKS.digital_display.show(("%4d"%pm25).replace(' ','#'))
        time.sleep(1800)
```

The Z notation to describe the principle of the PM2.5 indicator is shown below.

Get_pm25

weather_url : URL
content!:*CONTENT*
requestopen: *weather*↔*content*!
status: *BOOL*
pm25:N
report!:*RESPONSE*

((ran *requestopen*())≠∅∧*status*=*true*∧*pm*25=*n*∧
report!=*success*)∨
((ran *requestopen*())=∅∨*status*=*false*)∧
report!=*fail*)

LED_control

pm25,sleeptime, *itemon:*$\mathbb{N}_1$
itemoff, *beepaction: bool*

(*pm*25=-1∧*sleeptime*=30)∨
(((*pm25*>250∧*itemoff*=*true*∧*itemon*=7∧*beepaction*=*true*)∨
(115<*pm25*<250∧*itemoff*=*true*∧*itemon*=*7)*∨
(75<*pm25*<115∧*itemoff*=*true*∧*itemon*=6*)*∨
(35<*pm25*<75∧*itemoff*=*true*∧*itemon*=4*)*∨
(pm25<35∧*itemoff*=*true*∧*itemon*=4∧*itemon*=5*)*)∨
sleeptime=1800)

5. CONCLUSION AND FUTURE WORK

This paper finished the concept of open hardware, the organization of Raspberry pi, and the Z notation of the Raspberry pi. These research achievements advanced and developed. In the other paper work, these achievements will be technically explained with illustrative diagrams.

In the future, this research will focus on the under mentioned topics:

1. Can the Z language be used to describe the structure of the maker system? The maker is an organic component of the open source culture. It is a more complicated system compared with the Raspberry pi, so there will be more difficulties within associate descriptions.
2. How to avoid incorrect specifications? Compared with the informal specifications, Z notion has the capability of finishing formal, technical and precise specifications. The notion is an original reference for the state description and behavior description of objects in the system. However, Z notion does contain errors or contradictions insides specification applications. These problems will make formal design fail to be put into practice. Therefore, this research needs to do formal reasoning and validation to ensure the implementation of Z notation.

ACKNOWLEDGEMENTS

This paper is based upon our experience in using Z notation and the Raspberry pi at the Complex Networks and Visualization Laboratory, Shijiazhuang Tiedao University. We thank all the members of the Complex Networks and Visualization Laboratory.

We would like to thank the project “High-level personnel from institutions of higher learning in Hebei province (Item number GCC2014010)”. This research has been funded by this project.

The content of the PM2.5 indicator is based on the Raspberry Pi Laboratory. We would like to thank the Raspberry Pi Laboratory.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Available: https://www.raspberrypi.org/ 2016.
2. Jim Woodcock, Jim Davies. Using Z specification, refinement and proof. Prentice Hall; 1994.
3. Spivey JM. The Z notation: A reference manual. Programming Research Group. Prentice Hall; 1989.
4. Hengjun Zhao, Mengfei Yang, Naijun Zhan. Formal. Verification of a descent guidance control program of a Lunar Lander. Formal Methods: 19th International Symposium, Singapore; 2014.
5. Li Junshan. Formal. Verification of Lunar rover control software using UPPAAL. Formal Methods: 19th International Symposium, Singapore; 2014.
6. Leo Freitas, Jim Woodcock, Yichi Zhang. Verifying the CICS file control API with Z/Eves: An experiment in the verified software repository. Science of Computer Programming. 2009;74:197-218.
7. Zhao Zhengxu, Wen Jinjie, Zhao Weihua. Z notation and methods of use. Beijing: Science Press; 2015.
8. Wen Jinjie, Zhao Zhengxu. Z formalization of OpenGL graphics specification. Journal of the Hebei Academy of Sciences. 2014; 31(2):41-48.
9. Wen Jinjie, Zhao Zhengxu, Xu Qian. A test case generation strategy based on z formalization specification. Application Research of Computers. 2015;12:375-341.
10. Zhao Xiaofeng, Zhao Zhengxu. Specification information and virtual manufacturing environment Z described in Shandong: Shandong University; 2010.
11. Zhao Xiaofeng, Zhao Zhengxu. Study of environmental norms in virtual machining simulation technology Z-based system simulation. Journal of System Simulation. 2009;21(22):7143-7146.
12. Available:http://www.iso.org/iso/catalogue_detail?csnumber=21573 2002.

13. Available:http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=46112 2007.
14. Bowen JP. Z glossary. Information and Software Technology. 1995;37(5):333-334.
15. Smith G. The object-z specification language. Springer; 2000.
16. Harrison MA. Introduction to formal Language Theory. Addison-Wesley; 1978.
17. Available: http://shumeipai.nxez.com/ 2016.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/14214