A Study of Raspberry Pi Applications to Calligraphy Learning and Air-Conditioning Guidance Systems

2020, September

Samsul Huda

Graduate School of
Natural Science and Technology

(Doctor's Course)
OKAYAMA UNIVERSITY

Dissertation submitted to
Graduate School of Natural Science and Technology
of
Okayama University
for
partial fulfillment of the requirements
for the degree of
Doctor of Philosophy in Engineering.

Written under the supervision of

Professor Nobuo Funabiki

and co-supervised by
Professor Satoshi Denno
and
Professor Yasuyuki Nogami

OKAYAMA UNIVERSITY, September 2020.

To Whom It May Concern

We hereby certify that this is a typical copy of the original doctor thesis of
Samsul Huda

Signature of	Seal of
the Supervisor	
	Graduate School of
Prof. Nobuo Funabiki	Natural Science and Technology

Abstract

Recently, *Raspberry Pi* has been widely adopted for the computing platform in a lot of applications due to the features of the small size, the low cost, yet the rich functions, since it has been developed for educational purposes. *Raspberry Pi* has the basic hardware and software components as a single-board computer. As the hardware, it is equipped with the Ethernet port and the Wi-Fi adapter for network connections, and the USB port for the use of external devices such as a mouse, a keyboard, and a camera. It is also equipped with the *General Purpose Input Output (GPIO)* pins for use of various sensor and actuator devices. As the software, *Raspberry Pi* has preinstalled Linux-based *Raspbian* operating system and several popular programming languages including Python, C, C++, Java, Scratch, and Ruby. Hence, the popularity of *Raspberry Pi* is now increasing rapidly around the world for developing practical applications, particularly, for *IoT (Internet of Things)*.

In this thesis, we study two practical applications of *Raspberry Pi* in real worlds, namely, the *calligraphy learning assistant system (CLAS)* and the *air-conditioning guidance system (AC-Guide)*. Through the application studies, the efficient implementation, the hardware/software cost, the developing period, and the performance of *Raspberry Pi* application systems are investigated. For the implementations, Python is adopted for developing the application programs under *Raspbian* OS.

First, the application of *Raspberry Pi* to the *calligraphy learning assistant system (CLAS)* is studied. This system aims to assist self-learning of *calligraphy*. Using the *projection mapping* technology, *CLAS* directly projects a letter writing video of a teacher on the paper on which a learner will write letters. Then, the learner is able to learn the stroke order and writing speed in addition to the letter shape by tracing the writing of the teacher. Besides, the *letter portion practice function* is incorporated to allow a learner to repeat practicing hard portions of each letter. In the implementation of *CLAS*, we adopt a portable projector and open-source software for video processing and *projection mapping*. Through applications to international students from Japan, China, Myanmar, Indonesia, Bangladesh, and Kenya, we confirm the effectiveness in improving *calligraphy* skills for novice students.

Second, the application of *Raspberry Pi* to the *air-conditioning guidance system (AC-Guide)* is studied. This system aims to encourage the proper use of air-conditioning related devices/equipment in a room by sending alarm messages of turning them on/off. *AC-Guide* periodically samples the temperature and humidity in the room to calculate the *discomfort index (DI)*, and observes the AC usage using a web camera. The outdoor DI is also calculated by obtaining the weather data from an API called *openweathermapAPI*. Then, by referencing to the message decision table, the system outputs a voice alarm message and an email alarm message to the registered persons. In the implementation of *AC-Guide*, we adopt a temperature/humidity sensor with *GPIO*, a USB web camera, a USB speaker, and an open-source software for image processing. Through applications at two rooms in Okayama University, we confirm the correctness of the alarm messages.

These studies found that the implemented *Raspberry Pi* application systems require a small cost (less than 100,000 yen each) both for the hardware and software and the short developing period (less than one year by one graduate student), and yet, exhibit the acceptable performances for practical use.

In future studies, the *CLAS* will be improved by solving the brushwork cover-up problem by the teacher's hand, adding the brushwork instructions, adjusting the playback speed, and implementing the evaluation function of the stroke quality. The *AC-Guide* will be improved by considering the target temperature of the AC, using multiple sensors in a room, adding the coordinated guidance of multiple rooms, counting the number of residents in the room for more proper messages, sending turn-on/off messages to other devices such as a fan, a humidifier, and dehumidifier, also continue experiments in various rooms.

Acknowledgements

It is my great pleasure to express my heartiest thankfulness to those who gave me valuable time and supported me in making this dissertation possible. I believe that you are the greatest blessing in my life. Thanks all of you to make my dream successful.

I owe my utmost gratitude to my supervisor, Professor Nobuo Funabiki, who has supported me throughout my thesis with his patience and knowledge. I am greatly indebted to him, whose encouragement, advice, and supports from the beginning has enabled me to develop the understanding of this subject, not only in scientific matter but also in life. He gave me wonderful advice, comments, and guidance when writing papers and presenting them. I will not become who I am now without him.

I am deeply grateful to my co-supervisors, Professor Satoshi Denno and Professor Yasuyuki Nogami, for their continuous supports, guidance, and proofreading of this work.

Also, I wish to express that I was honored to get valuable suggestions from Associate Professor Minoru Kuribayashi during my research. Additionally, my highest appreciation is also directed to the course teachers during my Ph.D. study who delivered enlightenment and valuable knowledge.

I would like to acknowledge the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT) for financially supporting my Ph.D. study, Politeknik Elektronika Negeri Surabaya (PENS) which gave me the chance to be able to pursue this degree, also to my institution where I work, Universitas 17 Agustus 1945 Surabaya (UNTAG Surabaya), for giving me the support with the study leave permission.

Therefore, I would like to give the highest acknowledgment to a calligraphy teacher, Mrs. Naomi Funabiki, for her sincere help in producing the video contents and grading the student's letter results.

I would like to appreciate the fruitful discussions and cooperation with many people including Prof. Wen-Chung Kao, Dr. Amang Sudarsono, Dr. Sritrusta Sukaridhoto, Dr. Nobuya Ishihara, Dr. Khin Khin Zaw, Dr. Kyaw Soe Lwin, Dr. Sumon Kumar Debnath, Dr. Md. Manowarul Islam, Dr. Rahardhita Widyatra Sudibyo, Ms. Mousumi Saha, Mr. Kwenga Ismael Munene, Mr. Hendy Briantoro, Mr. Md. Mahbubur Rohman, Mr. Hein Htet, Ms. Pradini Puspitaningayu, and Ms. Irin Tri Anggraini. I would like to convey my respect to all members of FUNABIKI Lab for their support during the period of this study.

Finally, I am eternally grateful to my beloved parents, brothers, sister and all of my friends who always encourage and support me throughout my life. Your supports and understanding gave me the strength and inspiration to overcome any difficulties in my life.

Samsul Huda
Okayama University, Japan
September 2020

List of Publications

Journal Papers

1. **S. Huda**, N. Funabiki, M. Kuribayashi, and W.-C. Kao, “A proposal of calligraphy learning assistant system with letter portion practice function using projection mapping,” Int. J. of Web Inf. Sys., vol. 16, No. 2, pp. 137-149, Oct. 2019.

International Conference Papers

2. **S. Huda**, N. Funabiki, M. Kuribayashi, and W.-C. Kao, “A proposal of calligraphy learning assistant system using projection mapping,” Proc. of Int. Conf. on Scie. and Eng. (ICSE), pp. 10-15, December 2018.

3. **S. Huda**, N. Funabiki, M. Kuribayashi, and W.-C. Kao, “A calligraphy learning assistant system with letter portion practice function using projection mapping,” Proc. of IEEE Int. Conf. on Con. Elec. (ICCE), pp. 1-2, January 2020.

Other Papers

4. **S. Huda**, L. Xiqin, N. Funabiki, and M. Kuribayashi, “A Proposal of Calligraphy Learning Assistant System using Projection Mapping,” IEICE Tech. Rep., vol. 118, no. 211, MVE2018-23, pp. 53-58, Sep. 2018.

5. **S. Huda**, N. Funabiki, M. Kuribayashi, and H. H. S. Kyaw, “An Evaluation of Calligraphy Learning Assistant System Using Projection Mapping,” IEICE General Conf., BS-4-32, pp. S82-S83, March. 2019.

6. **S. Huda**, N. Funabiki, and M. Kuribayashi, “An Implementation of Letter Portion Practice Function in Calligraphy Learning Assistant System Using Projection Mapping,” For. of Inf. Tech. (FIT), Vol. 3, pp. 351-352, Sep. 2019.

7. **S. Huda**, N. Funabiki, M. Kuribayashi, R. W. Sudibyo, and N. Ishihara, “A Proposal of Air Conditioner Overuse Alarm System Using Raspberry Pi,” IEICE General Conf., BS-1-16, pp. S28-S29, March 2020.

8. **S. Huda**, N. Funabiki, M. Kuribayashi, R. W. Sudibyo, and N. Ishihara, “An Improvement of Air-Conditioning Guidance System with Outdoor Discomfort Index for Avoiding False Messages,” IEICE Tech. Rep., NS2020-27, pp. 31-36, June 2020.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Background

Nowadays, *IoT (Internet of Things)* has been rapidly growing with great advancements of inexpensive computing devices and networking technologies [1]. This makes it possible to access and control the physical devices from distant places by exchanging data over the Internet. Many people in industries, educations, and governments around the world take this opportunity to develop IoT applications for their use. A variety of IoT applications have appeared in different domains, including transportation, energy, home, healthcare, agriculture, logistics or industry [1–5, 40, 41].

Raspberry Pi is one of the most common computing devices in both researches and applications of IoT. *Raspberry Pi* has been popular as the computing platform in a lot of applications due to the features of the small size, the low cost, yet the rich functions, although it has been developed for an educational purposes. It was invented as educational device to inspire children to study computer science.

Raspberry Pi has the basic hardware and software components as a single-board computer [9]. For the hardware, it is equipped with the Ethernet port and the Wi-Fi adapter for network connections, and the USB port for the use of external devices such as a mouse, a keyboard, and a camera. It is also equipped with the *General Purpose Input Output (GPIO)* pins for use of various sensor and actuator devices. For the software, it has preinstalled Linux-based *Raspbian* operating system, and several popular programming languages, including Python, C, C++, Java, Scratch, and Ruby. Hence, the popularity of *Raspberry Pi* is now increasing rapidly around the world in developing practical applications, particularly, for *IoT* [6]. As a result, the study of practical IoT application systems using *Raspberry Pi* is significant.

1.2 Contributions

In this thesis, we study two practical application systems using *Raspberry Pi* in real worlds, namely, the *calligraphy learning assistant system (CLAS)* and the *air-conditioning guidance system (AC-Guide)*. Through the application studies, we investigate the efficient implementation, the hardware/software cost, the developing period, and the performance of *Raspberry Pi* application systems. In these systems, we implement application programs using *Python* on *Raspbian* OS.

First, we study the application of *Raspberry Pi* to the *calligraphy learning assistant system* [10–15]. This system is named *CLAS*, and aims to assist self-learning of *calligraphy*.

Using the *projection mapping* technology, *CLAS* directly projects a letter writing video by a teacher on the paper on which a learner will write letters. Then, the learner is able to learn the stroke order and the writing speed in addition to the letter shape by tracing the writing of the teacher. Besides, the *letter portion practice function* is incorporated to allow a learner to repeat practicing hard portions of each letter.

In the implementation of *CLAS*, we adopt a portable projector and open-source software for video processing and *projection mapping*. Through applications to international students from Japan, China, Myanmar, Indonesia, Bangladesh, and Kenya, we confirm the effectiveness in improving *calligraphy* skills for novice students.

Second, we study the application of *Raspberry Pi* to the *air-conditioning guidance system* [16, 17]. This system is named *AC-Guide*, and aims to encourage the proper use of air-conditioning related devices/equipment in a room by sending alarm messages of turning on/off them.

AC-Guide periodically samples the temperature and humidity in the room to calculate the *discomfort index (DI)* there, and observes the AC use using a web camera. In addition to this *indoor DI*, the *outdoor DI* is calculated by obtaining the weather data from an API called *openweathermapAPI*. Then, by referencing to the message decision table, the system outputs a voice alarm message and an email alarm message to the registered persons.

In the implementation of *AC-Guide*, we adopt a temperature/humidity sensor with *GPIO*, a USB Web camera, and a USB speaker, and open-source software for image processing. Through applications to two rooms in Okayama University, we confirm the correctness of the alarm messages.

Through these studies, we conclude that for a *Raspberry Pi* application system, the required cost for the hardware and software is low (less than 100,000 yen for the total), the developing period is short (less than one year by one graduate student), and the performance is acceptable for the practical use.

1.3 Contents of Thesis

The remaining part of this thesis is organized as follows.

In Chapter 2, we discuss the background technologies related to this thesis, including *Raspberry Pi 3 Model B+*, *GPIO*, *Raspbian OS*, *Python*, *Python Libraries*, and *SSH*, *WinSCP* to transfer files between *Raspberry Pi* and the computer, and the developing process of Raspberry applications.

In Chapter 3, we present the study of the application of *Raspberry Pi* to the *calligraphy learning assistant system* with the implementations and the evaluations.

In Chapter 4, we present the study of the application of *Raspberry Pi* to the *air-conditioning guidance system* with the implementations and the evaluations.

In Chapter 5, we review relevant works in literature.

Finally, in Chapter 6, we conclude this thesis with some future works.

Chapter 2

Background Technologies

In this chapter, we review the background technologies for the implemented *Raspberry Pi* application systems in this dissertation.

2.1 Importance of *Raspberry Pi*

Raspberry Pi is a one-board computing system that was initially intended for educational purposes to inspire school children to study computer science [9,18,19]. It allows having experiments with hardware configurations. *Raspberry Pi* has been developed by *Raspberry Pi Foundation* in the United Kingdom. Due to the features of the small size, the low cost, and the rich functions, *Raspberry Pi* has been used by many people around the world for industries, educations, or governments, as the computing platform of various application systems [1–5,40,41].

2.1.1 Raspberry Pi 3 B+ Model

Raspberry Pi 3 model B+ is the final revision of the third generation of *Raspberry Pi*. It was released in March 2018 [20]. Figure 2.1 shows the *Raspberry Pi 3 B+* board. It has the basic hardware and software components as a single-board computer [9]. For the hardware, it is equipped with the Ethernet port and the Wi-Fi adapter for network connections, the USB port for the use of external devices such as a mouse, a keyboard, and a camera, and the *General Purpose Input Output (GPIO)* pins for use of various sensor and actuator devices. Table 2.1 shows the details of *Raspberry Pi 3 model B+* specifications [20].

For the software, *Raspberry Pi 3 model B+* has preinstalled Linux-based *Raspbian* operating system, and popular programming languages, including *Python*, *C*, *C++*, *Java*, *Scratch*, and *Ruby*. Hence, the popularity of *Raspberry Pi* is now increasing rapidly around the world as the computing platform for practical applications, particularly, *IoT (Internet of Things)* [6].

2.1.2 General Purpose Input Output (GPIO)

Raspberry Pi 3 model B+ offers a 40-pin *GPIO (General Purpose Input/Output)* connector, for interfacing with external sensors and actuators [20]. The sensors can connect directly to the GPIO pins using jumper wires or with a breadboard. The GPIO pins can be used for driving LEDs, controlling external devices, sensing digital inputs, and waking up the device.

Figure 2.2 shows the GPIO pins layout. The GPIO pins on the *Raspberry Pi* are divided into the following three groups [22]:

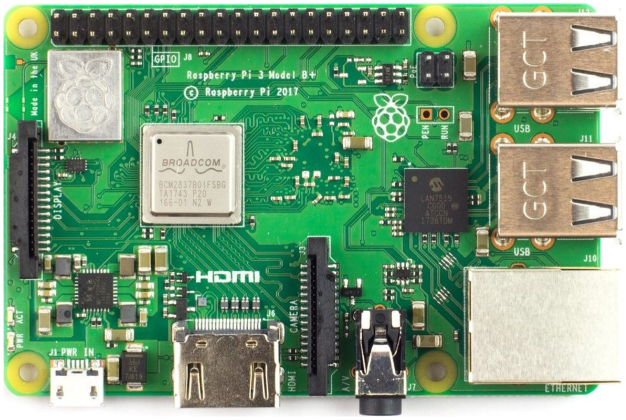


Figure 2.1: *Raspberry Pi 3 model B+* board.

Table 2.1: *Raspberry Pi 3 model B+* technical specifications.

Components	Details
Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory	1GB LPDDR2 SDRAM
Connectivity	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
	Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)
	4 × USB 2.0 ports
Access	Extended 40-pin GPIO header
Video sound	1 × full size HDMI
	MIPI DSI display port
	MIPI CSI camera port
	4 pole stereo output and composite video port
Multimedia	H.264, MPEG-4 decode (1080p30); H.264 encode(1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support	Micro SD format for loading operating system and data storage
Input power	5V/2.5A DC via micro USB connector
	5V DC via GPIO header

- *Power:*
  The pins labeled with *5.0v* supply 5 volts of power, and the pins with *3V3* supply 3.3 volts of power. There are two *5.0v* pins and two *3V3* pins.

- *GND:*
  The pins represent the ground pins. There are eight ground pins.

- *Input/Output:*
  The pins labeled with GPIO can be used for input or output. For example, GPIO 2 is on pin 3. Some of them are designed for specific purposes such as PWM (pulse-width modulation), SPI (Serial Peripheral Interface), I2C, and Serial.

It is important to connect the sensor/actuator pins to the corresponding correct GPIO pins. Connecting the wrong pins may damage the sensor or actuator permanently.

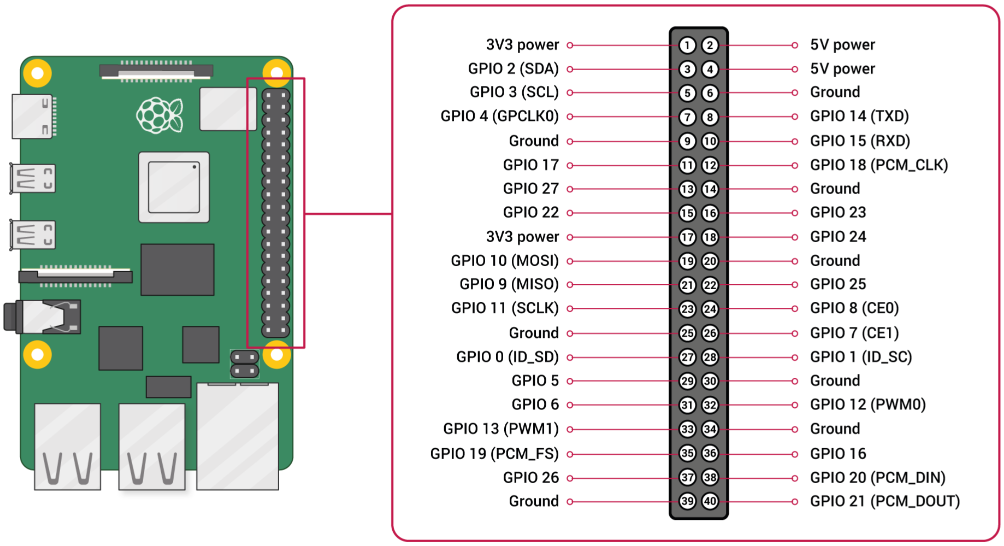


Figure 2.2: GPIO pins layout.

2.1.3 Raspbian OS

Raspbian OS is the official operating system (OS) for the *Raspberry Pi* board [23, 25]. *Raspbian OS* is based on the *Linux Debian OS*, which is optimized for the *Raspberry Pi* hardware. *Raspbian OS* is booted off from a micro-SD card where the entire operating system runs off the card.

For running *Raspberry Pi*, the micro SD card that has the preinstalled *Raspbian OS* has to be inserted in the micro SD card slot. This micro SD card serves two purposes. It provides the OS for the computer, and acts as the memory storage device. Also, *Raspberry Pi* can connect to an external hard disk or flash drive for more storage.

2.2 Python

Python is one of the most popular and powerful programming languages among system developers, due to its clear syntax and easy coding even for beginners. Basically, *Python* is the language that can be used for developing anything and everything. Like shell scripts, *Python* can automate the manual tasks. In general, *Python version 2* and *Python version 3* have been pre-installed on *Raspbian OS* [23].

Python offers following benefits [24–26]:

- *Readable and maintainable code:*
  The syntax rules allow to implement concepts without writing additional codes. Unlike other programming languages, it emphasizes code readability and allows to use English keywords instead of punctuation. Thus, *Python* can be used to build custom applications without writing additional codes.

- *Compatibility with major platforms and systems:*
  Python supports many operating systems. It allows to run the same code on multiple platforms without recompilations.

- *Robust standard library:*
  Python has a large collection of libraries, which can speed up the development process. The standard *Python* libraries allow to use a wide range of modules according to precise needs.

- *Many open source frameworks and tools:*
  Python, being an open source programming language, helps to reduce the software development cost significantly. *Python* gives the access to use several open source frameworks, libraries, and development tools.

2.3 Python Libraries for *Raspberry Pi* Application Systems

In this section, we give the overview of *Python* libraries used for developing and implementing *Raspberry Pi* applications.

2.3.1 DHT Library

In this study, we use the *DHT library* to interact with the *DHT22* temperature/humidity sensor. This library allows to easily retrieve the temperature and humidity from the sensor with a few lines of *Python* code [27]. It can be installed to *Raspberry Pi* using the following command:

```
$ pip3 install Adafruit_DHT
```

2.3.2 OpenCV

Open Source Computer Vision (OpenCV) is the open-source image processing toolbox that has been used for computer vision applications [28]. *OpenCV* is a library composed of image processing functions. It can be used to process images, videos, and even live streams. To install the latest version of *OpenCV* on *Raspberry Pi*, the following commands should be used:

```
$ sudo apt-get install libhdf5-dev libhdf5-serial-dev libhdf5-100
$ sudo apt-get install libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5
$ sudo apt-get install libatlas-base-dev
$ sudo apt-get install libjasper-dev
$ pip3 install opencv-python
$ pip3 install opencv-contrib-python
```

2.3.3 GMAIL SMTP

Python comes with the built-in smtplib module for sending emails using the *Simple Mail Transfer Protocol (SMTP)* [23]. This module makes *Python* possible to send confirmation emails to many users or receive email reminders. To send emails through *Gmail SMTP*, we use the built-in smtplib module without installing any additional packages.

In a new system implementation, it is highly recommended to set up a new Gmail account [29]. The Gmail account's security settings must be set properly to allow accesses from *Python* code. Besides, any chance that the system user might accidentally expose his/her login details must be avoided.

To set up a Gmail address for sending emails, the following procedures are necessary:

- Create a new Google account.
- Login to Google account.
- Configure the security setting of google account in `https://myaccount.google.com/security`.
- Turn ON access under "Less secure app access" section.
- Then, we can access the Gmail account from *Python* code.

2.3.4 Flask

Flask is a micro-framework written in *Python* that makes it easy to set up a simple Web application server running with features that can be useful in the system development process [30]. *Flask* is a very lightweight Web server.

Flask comes with the bare minimum required for the application. It allows users to add any package or functionality without being held down by the ones that come with a framework. With *Flask*, it is possible to present the sensor data to remote clients. To install *Flask* on *Raspberry Pi*, the following command should be used:

```
$ sudo apt-get install python3-flask
```

2.3.5 Schedule Library

Schedule Library is used to schedule a task at a particular time on every day or a particular day of a week. It allows the specified *Python* functions to run periodically at pre-determined intervals using a simple and human-friendly syntax [31]. Basically, *Schedule Library* compares the system time with the scheduled time set by user. Once the scheduled time and the system time become equal, the schedule job function is called. It can be installed manually using the following command [32]:

```
$ pip3 install schedule
```

2.4 SSH

SSH is an abbreviation of *Secure Shell.* SSH is a cryptographic network protocol to securely initiate a shell session on a remote host [33, 34]. SSH consists of the *SSH client* program and the *SSH server* program. It can establish a secure channel between them over an insecure network. *Raspbian* contains the *SSH server*, but is disabled by default [35]. It can be enabled manually from the desktop, or alternatively by using *systemctl* command, to start this service:

```
$ sudo systemctl enable ssh
$ sudo systemctl start ssh
```

Raspberry Pi can be connected as the *SSH server* from another machine at the same network using *SSH client* there.

2.5 WinSCP

To transfer files to or from a server using the *SSH File Transfer Protocol (SFTP)*, an SSH or SFTP client is necessary at the client side. *WinSCP* is a free SFTP, SCP, and FTP client utility for Windows OS, which can be freely download from [36]. It can be used to show, copy, and manage files and folders on *Raspberry Pi* as a remote device from windows PC. In this thesis, *WinSCP* is used for transferring Python script files and CSV data files between *Raspberry Pi* and hosts.

To operate *WinSCP*, the following requirements must be satisfied [37]:

- Both *Raspberry Pi* and a Windows PC need to be connected on the same network.
- The IP addresses of *Raspberry Pi* and the Windows PC should be configured, either DHCP or static IP.
- *Raspberry Pi* needs to have SSH enabled.

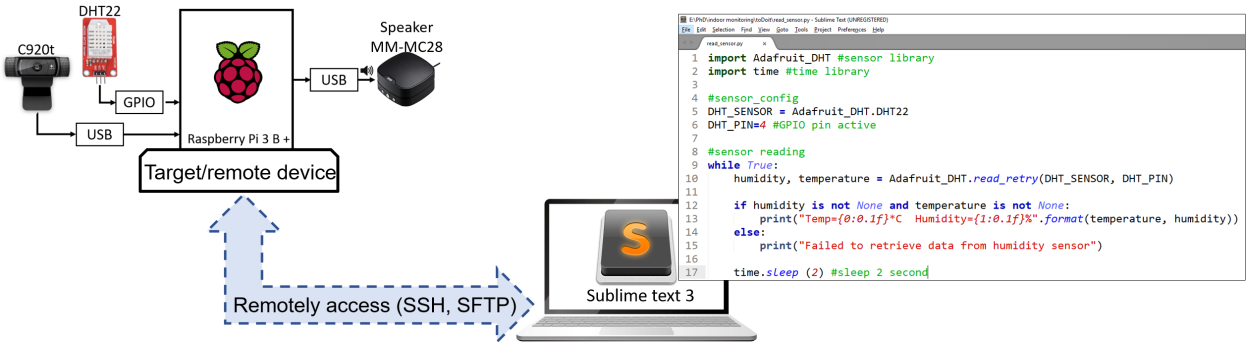
2.6 Developing *Raspberry Pi* Applications

By using a simple editor in *Raspberry Pi* such as *vim* or *nano* and the Python interpreter, we can develop and run *Python* programs for *Raspberry Pi* applications. Here, the peripherals such as a keyboard, a mouse, and a monitor are needed to develop them on *Raspberry Pi*. Hence, it becomes more complicated when *Raspberry Pi* is connected with sensors and other devices. Furthermore, it is easier to use a text editor that includes *Python* syntax highlighting.

Using an IDE (Integrated Development Environment) to develop *Python* program is more simple, because IDE integrates several tools and contains many features useful in software developments including the interpreter and the text editor. However, it consumes large memory.

A dedicated code editor is simple but has syntax highlighting and code formatting capabilities. Compared to an IDE, a dedicated code editor is usually smaller and quicker, but may often less features [38].

Sublime Text is a popular code editor that supports many languages including *Python*. It is fast and highly customizable and well supported. Besides the syntax highlighting (code coloring), it provides indentation guides [39]. Indentation is very important in *Python*. The body of a block is defined by its indentation, where indentation is an integral part of *Python* syntax.



```
import Adafruit_DHT #sensor library
import time #time library

#sensor_config
DHT_SENSOR = Adafruit_DHT.DHT22
DHT_PIN=4 #GPIO pin active

#sensor reading
while True:
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)

    if humidity is not None and temperature is not None:
        print("Temp={0:0.1f}*C  Humidity={1:0.1f}%".format(temperature, humidity))
    else:
        print("Failed to retrieve data from humidity sensor")

    time.sleep (2) #sleep 2 second
```

Figure 2.3: Development process of *Raspberry Pi* application.

In this thesis, we develop the *Python* programs on a PC remotely, not on *Raspberry Pi*. Therefore, the peripherals such as a keyboard, a mouse, and a monitor are not needed for *Raspberry Pi*. *Raspberry Pi* is connected with a sensor, a Web camera, and a USB speaker. They are placed in the designated place. Figure 2.3 shows the development process of a *Raspberry Pi* application. First, we write a Python script using *Sublime Text* in a Windows PC. Then, we transfer the script to *Raspberry Pi* through SFTP. Finally, we run the script on *Raspberry Pi* over SSH.

2.7 Summary

In this chapter, we reviewed the background technologies for implementations of *Raspberry Pi* application systems in this dissertation. They include *Raspberry Pi 3 Model B+*, *GPIO*, *Raspbian OS*, *Python*, *Python Libraries*, and *SSH*, *WinSCP* to transfer files between *Raspberry Pi* and the computer, and the developing process of Raspberry applications. In the next chapter, we will present the application of *Raspberry Pi* to *Calligraphy Learning Assistant System (CLAS)*.

Chapter 3

Application to Calligraphy Learning Assistant System (CLAS)

In this chapter, we present the study on the application of *Raspberry Pi* to the *calligraphy learning assistant system* that assists the self-learning of *calligraphy*.

3.1 Introduction

For several decades, *calligraphy* has been a popular activity worldwide since it is beneficial in relaxing and improving concentration and creativity. As well, it is addressed that people will boost their logic while practicing *calligraphy* [42,43].

In general, a teacher will write the letters on a paper with a calligraphy brush, and then, explain the stroke order and brushwork rules to a student. The teacher may hold the brush to show the proper strength between the brush and the paper [44]. If the brush is strongly pressed on the paper, the line will become wide and not smooth. That is, a student will practice writing letters on papers while observing the letters written by the teacher on the left side. Then, the teacher will correct the letters. Figure 3.1 illustrates this conventional calligraphy learning method.

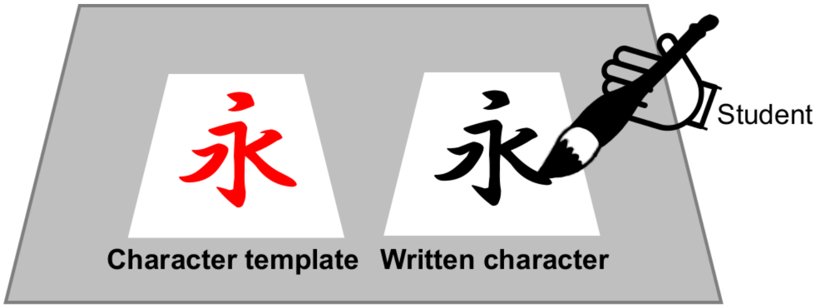


Figure 3.1: Conventional calligraphy learning method.

However, this method requires a teacher to be present all the time. Besides, the details of the brushworks for letter writing, such as the stroke order and the writing speed, are not involved in the static letters on the paper.

In this chapter, to assist the self-learning of *calligraphy*, we propose a *Calligraphy Learning Assistant System (CLAS)* using *projection mapping*. CLAS allows a learner to practice calligraphy writing with the dynamic brushwork by him/herself. By following the letter writing video of a teacher that is directly projected onto the paper, a student can learn the stroke order as well as the writing speed. Thus, the content of CLAS is the video-recording of letter writing by a teacher. The *letter portion practice function* is incorporated in CLAS, to allow a learner to repeat practicing hard portions of a letter. The repeated practices of hard portions are very useful for improving the whole letter writing.

3.2 Overview of Projection Mapping Technology

In this section, we overview the projection mapping technology.

Projection mapping is a technology of projecting videos or images on various types of objects to offer realistic presentations. A projection mapping system basically consists of a *computer* with the dedicated projection mapping software and a *projector*. As shown in Figure 3.2, the software maps each mesh of a visual content onto one surface mesh of the object, called *meshing*, and processes each content mesh using image conversion techniques so that it can naturally appear on the corresponding surface mesh using a *projector*. The ordinary projector can be used in projection mapping [53]. *Projection mapping* has been used in various applications, such as music visualizations, product presentations, or educational purposes [45–47].

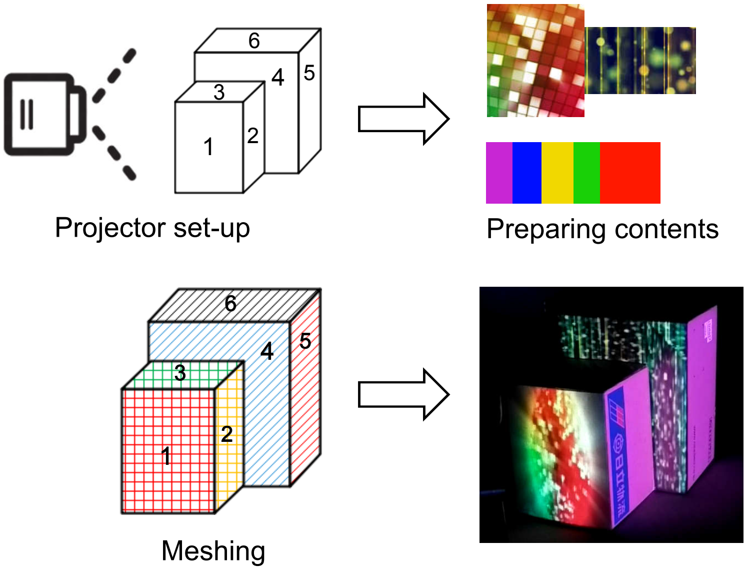


Figure 3.2: Content generation for projection mapping.

3.3 System Configuration

Figure 3.3 illustrates the system configuration of CLAS[11, 14, 15]. *Raspberry Pi* is used as the computing device to process the visual content or video for projection mapping using the projection mapping software. *Raspberry Pi* owns the processing capability of a high-definition video. Then, *openFrameworks* is adopted as an open source creative coding toolkit, and *OfxPiMapper* [48, 49] is installed as an open source projection mapping software, which acts as an *add-on* of *openFrameworks*.

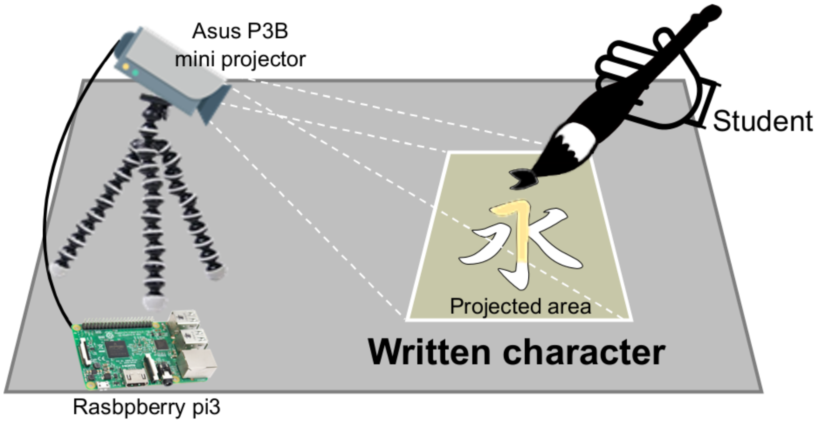


Figure 3.3: System overview.

ASUS P3B is adopted as a portable projector to project the video onto the surface of the paper on a desk. It supports until $800 lumens$ light with $12,000 mAh$ power built-in. In the experiments, this projector is set on the left side of the paper to project the video onto the paper surface.

3.4 CLAS Utilization

CLAS can be utilized through the four steps: 1) content recording, 2) content processing, 3) projection setup, and 4) calligraphy practicing. More details are as follows:

3.4.1 Content Recording

First, the video of letter writing by a teacher is video-recorded for the content. In our implementation, a smart-phone is used as the recording device and is mounted in the proper position on the desk before recording. A calligraphy teacher who has calligraphy experiences of over 40 years helped the video-recording in this study.

3.4.2 Content Processing

Then, the CLAS video content is generated from the video. To assist a novice learner efficiently practicing calligraphy writing, each content consists of both a movie and an image. The *movie* offers the dynamic brushwork movement of each letter writing by the teacher. Using *FFmpeg* [50], the brush and letter parts in the movie are highlighted, and the other parts such as the teacher's hand are diluted, to improve the visibility of letter writing.

The *image* represents the contour of each letter, which is used to help a learner write the correct shape. The letter contour is obtained by scanning the complete letter of the teacher on the paper using a scanner, and extracting the contour from the scanned letter using an open source software *inkscape* [51].

Finally, *FFmpeg* is used to produce one content by blending the movie and the image, as revealed in Figure 3.4.

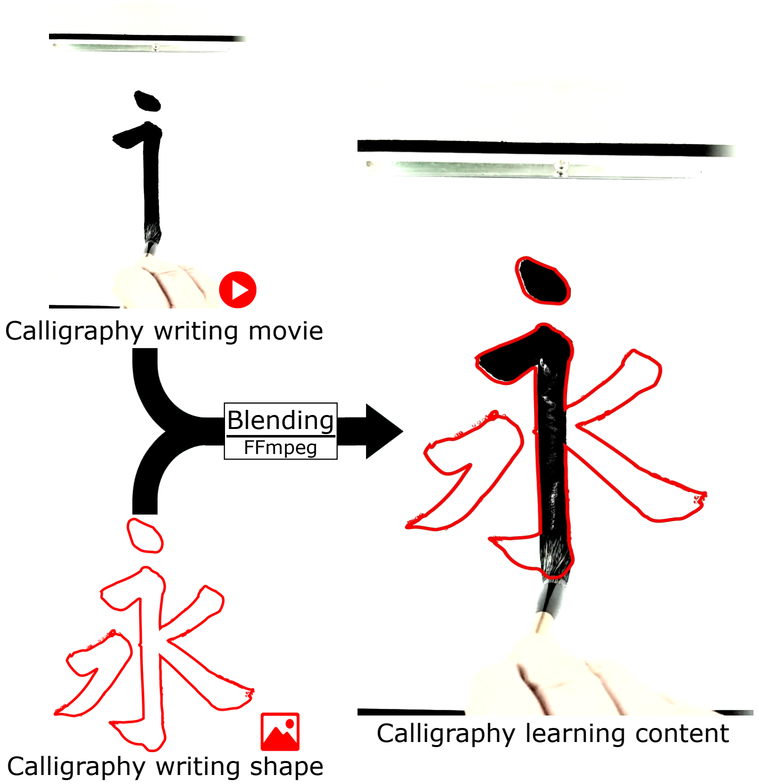


Figure 3.4: CLAS content production.

3.4.3 Projection Setup

In CLAS, the square paper recorded in the video content must be projected correctly onto the real square paper on the desk using a projector that is installed on the desk at the left oblique projection of the paper. The projection parameters in the software must be adjusted properly. This adjustment should be done once when the projector and the paper are set up on the desk.

3.4.4 Calligraphy Practicing

At CLAS, a learner will practice calligraphy writing using a real brush, ink, and paper on the desk, by tracing the letter writing in the content. By watching and tracing the brushwork of the teacher and the letter contour in the content, it is expected for a learner to understand and master the stroke order, the writing speed, and the writing pressure of the teacher.

3.5 Letter Portion Practice Function

In this section, we present the *letter portion practice function* in CLAS.

3.5.1 Letter Portions

A Japanese letter including a *kanji* usually consists of several portions or strokes. For example, “永” consists of five portions shown in Figure 3.5. “永” has been often used in calligraphy learning, because it has the eight basic brush strokes called *Eiji-happo* [52]. They include *Ten* (dot), *Yokoga* (horizontal stroke), *Tatega* (vertical stroke), *Hane* (upflick from a horizontal or vertical stroke), *Migihane* (rightward upflick), *Hidaribarai* (leftward downstroke), *Hidarihane* (leftward downflick), and *Migibarai* (rightward downstroke). For the proposed function, we prepare a video of writing each letter portion only several times by a teacher. Then, a learner could repeat practicing the portion by imitating the video.

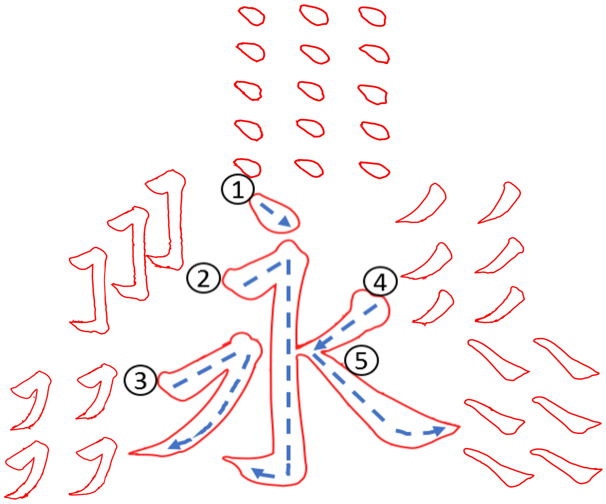


Figure 3.5: Five letter portions of “永”.

3.5.2 Implementation

The user interface in Figure 3.6 is implemented using *LibreOffice*, to allow a learner to select one portion or the whole letter writing for practice. A mouse and a keyboard are necessary for this user interface. By clicking a button, the corresponding video is automatically projected on the paper. In the implementation, the *macro function* and the *shell script* are used.

Figure 3.6: User interface for selecting the learning practice modes.

Figure 3.7 demonstrates the letter portion practice by a student using the proposed function. A student can repeat practicing the difficult portion by following the writing of a teacher that is directly projected on the paper. However, the current system does not have the function to evaluate the student writing and feedback the result on real time.

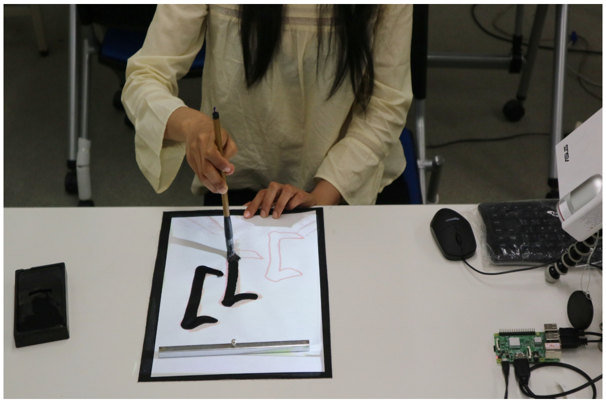


Figure 3.7: Letter portion practice using proposal.

3.6 Evaluations

In this section, we evaluate the *calligraphy learning assistant system* through applications to students in Okayama University.

3.6.1 Evaluation Indexes

To evaluate calligraphy writing results of students quantitatively, the *difference rate* and the *teacher grade* are used in this study.

3.6.1.1 Difference Rate

The *difference rate* indicates the rate of the number of the pixels whose binary values are different between the teacher's calligraphy image and the student's calligraphy image against the total number of pixels in the image. It represents the similarities of the letters from the teacher and the student respectively. The *difference rate* can be calculated manually by the following procedure:

1. The calligraphy result paper by a teacher or a student is converted into the digital RGB image using a scanner.
2. The RGB image is converted into the gray-scale image using *inkscape*.
3. The gray-scale image is converted into the binary image using *inkscape*.
4. The binary image is resized to the image of 512×512 pixels using *inkscape*.
5. The number of different pixels between the teacher's image and the student's image is calculated.
6. The difference rate is calculated by dividing the number of different pixels with the total number of pixels using *ImageMagick* [54].

3.6.1.2 Teacher Grade

The *teacher grade* signifies the five point score of the calligraphy result on the paper that is given by a teacher subjectively in terms of the shape and the smoothness. 1 means the worst and 5 represents the best.

3.6.2 Comparison with Conventional Method

In the first evaluation, we verify the effectiveness of CLAS in writing higher quality calligraphy than the conventional method.

3.6.2.1 Application Process

In this evaluation, we asked 14 Japanese and non-Japanese students in our group, to write “永” as a common character for calligraphy by using CLAS and the conventional method. Then, we compared the two indices of the calligraphy results at three stages. The Japanese students have learned *calligraphy* in elementary schools, while the non-Japanese students have no experience at all.

At the first stage, we asked the students to write “永” with the conventional method, and divided them into two groups, such that the average value of *difference rate* is similar between them.

Then, at the second stage, we asked the students in the first group to write the letter again by CLAS, while the students in the second group do so by the conventional method. In order to improve calligraphy skills, each student repeated letter writing five times.

Afterwards, at the third stage, we asked them to write the letter by the conventional method.

3.6.2.2 Application Result

Table 3.1 shows the minimum, maximum, and average of *difference rate* and *teacher grading* in each group. This table indicates that both the indices are similar between the two groups in the first and third stages, and the indices by CLAS are better than those of the conventional in the second stage.

Table 3.1: Two indices for two methods.

stage		CLAS		conventional	
		difference rate (%)	teacher grade	difference rate (%)	teacher grade
1st	min.	17.98	2	17.09	2
	max.	23.00	3	23.70	4
	ave.	19.85	2.57	19.63	2.71
2nd	min.	14.56	2	17.33	2
	max.	23.15	5	25.60	5
	ave.	18.46	3.51	20.29	2.97
3rd	min.	19.28	2	16.69	3
	max.	23.72	4	24.18	4
	ave.	21.03	3	20.53	3.43

3.6.2.3 *T-test* Verification

To verify the observations, the *independent sample T-test* is applied to investigate the significant difference between the results of the two groups as shown in Table 3.2. Here, if *t-value<−t-table and p-value<0.05*, it is regarded that there is a significant difference between the two groups (yes). If *t-value<−t-table and p-value<0.1*, it is assumed that there is a notable difference tendency (maybe). Otherwise, there is no significant difference (no). If t-value is negative, the average value of the proposal is lower than that for the conventional.

Table 3.2: T-test result with α=0.05.

phase	T-test	difference rate	teacher grade
1	t-value	0.195543755	−0.40824829
	t-table	1.782287556	1.782287556
	p-value	0.424119645	0.345140929
	result	no	no
2	t-value	−2.007348298	1.551343504
	t-table	1.782287556	1.782287556
	p-value	0.033887595	0.073390691
	result	yes	maybe
3	t-value	0.414863684	−1.441153384
	t-table	1.782287556	1.782287556
	p-value	0.342781205	0.087561566
	result	no	maybe

Table 3.2 suggests that CLAS helps generating better calligraphy results at the second stage, while both methods are similar at the other stages. Thus, it is concluded that CLAS is effective in writing superior quality calligraphy. Through improving the satisfaction, CLAS can encourage novice students to continue learning calligraphy on their own. However, the results at the third stage imply that more practices by CLAS are necessary to improve calligraphy by the conventional method.

3.6.3 Effectiveness of Letter Portion Practice Function

In the second evaluation, we verify the effectiveness of the *letter portion practice function* in CLAS in improving whole letter writing.

3.6.3.1 Application Process

In this evaluation, we asked 12 non-Japanese students who have used CLAS before, to practice "永" using CLAS in three stages.

At the first stage, we asked them to write the whole of "永".

Then, at the second stage, we asked them to freely select two hard portions among the five ones of "永" in Figure 3.5, and to practice them using the *letter portion practice function* up to three times for each portion.

After that, at the last stage, we asked them again to write the whole letter using CLAS, and compared the two indices of the calligraphy results in the first and last stages.

3.6.3.2 Application Results

Table 3.3 shows the minimum, maximum, and average of *difference rate* and *teacher grade* of the letters before and after applying the letter portion practice function. It indicates that after the application, the average *difference rate* is reduced by 4.8% and the average *teacher grade* is increased by 31.6%. In general, most of the students obtained the highest grade after the application.

Table 3.3: Two indices for letter portion practice function.

	Learning times	difference rate (%)		teacher grade	
		before	after	before	after
Min.	1	18.09	16.27	2	3
Max.	3	20.67	19.10	4	5
Ave.	2.58	19.28	18.34	3.08	4.5

3.6.3.3 *T-test* Verification

Table 3.4 exhibits the *paired T-test* results. It suggests *t-value>t-table and p-value<0.05*, which means that there are significant differences between before and after the application in both indices. Therefore, the effectiveness of the letter portion practice function is confirmed even in the short time use.

Table 3.4: Paired *T-test* result with α=0.05.

T-test	difference rate	teacher grade
t-value	3.634374427	−7.340392083
t-table	2.20098516	2.20098516
p-value	0.003926501	0.000014659
result	yes	yes

3.6.3.4 Hard Letter Portions

Table 3.5 presents hard letter portions for the students. The portions 2 and 3 consist of multiple basic brush strokes. Thus, they need to write different brush strokes continuously, which is considered challenging for novice students. The portion 5 needs the slow brush work in *Migibarai*, which can be also difficult for them.

Table 3.5: Selected hard portions by students.

hard portions	# of students
2 & 3	4
2 & 5	4
3 & 5	4

Figure 3.8 provides the example calligraphy results by students before and after applying the proposed function. It is observed that the results after application are better than the ones before applying.

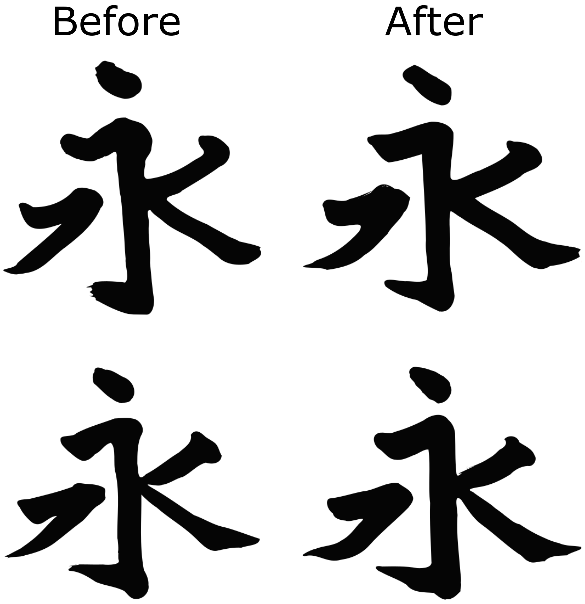


Figure 3.8: Example calligraphy results by students.

3.6.4 Application to Various Letters

In the last evaluation, we validate the effectiveness of CLAS in writing other letters than "永".

3.6.4.1 Application Process

In this evaluation, we prepared the model letters by the teacher and the CLAS contents for "一", "十", "山", and "岡大", and asked 13 Japanese and non-Japanese students in our group, to practice their calligraphy writing, first, by following the conventional method, and then, by using CLAS. In addition, we asked the students to answer the four questions in the questionnaire with five points.

3.6.4.2 Application Results

Table 3.6 shows the minimum, maximum, and average of *difference rate* and *teacher grading* by the students for each content. This table indicates that the both indices by CLAS are better than those of the conventional in any content. It also suggests that "一" is much easier than the others where the *difference rate* is much smaller.

3.6.4.3 Questionnaire Results

Table 3.7 offers the questionnaire results. For further details, 64% of the students answered that they could learn *calligraphy* more efficiently by CLAS than the conventional method. 65% reported that they could write excellent calligraphy, whereas 7% of them did not. 85% agreed that CLAS is simple to operate for learning *calligraphy*. 64% expressed that their motivations for learning calligraphy were increased after using CLAS, whereas 7% did not agree because they have been familiar with the conventional method.

However, some students may find it challenging since they cannot see the brushwork in the video occasionally that is covered up by the teacher's hand, as shown in Figure 3.9. The alleviation of this cover-up problem will be discussed in future works. Besides, several students commented

Table 3.6: Two indices for four contents.

letter		CLAS		conventional	
		difference rate (%)	teacher grade	difference rate (%)	teacher grade
一	min.	8.91	2	7.22	1
	max.	12.75	4	17.41	4
	ave.	10.89	3.23	12.89	2.54
十	min.	10.32	2	14.20	2
	max.	17.62	4	22.87	4
	ave.	15.55	3.38	19.21	2.62
山	min.	9.16	2	16.31	2
	max.	18.14	4	26.60	4
	ave.	15.90	3.08	22.19	2.85
岡大	min.	14.87	2	17.01	1
	max.	21.04	4	22.97	3
	ave.	18.01	2.85	20.17	2.46

Table 3.7: Questionnaire results on system usability.

		1	2	3	4	5	
learning speed	slow	0	0	5	6	3	quick
calligraphy shape	poor	0	1	4	5	4	beautiful
easy use	hard	0	0	2	9	3	easy
learning motivation	useless	0	1	4	6	3	useful

that adding the brushwork instructions and adjusting the play back speed can be effective to further improve this system, which will be also included in future studies.

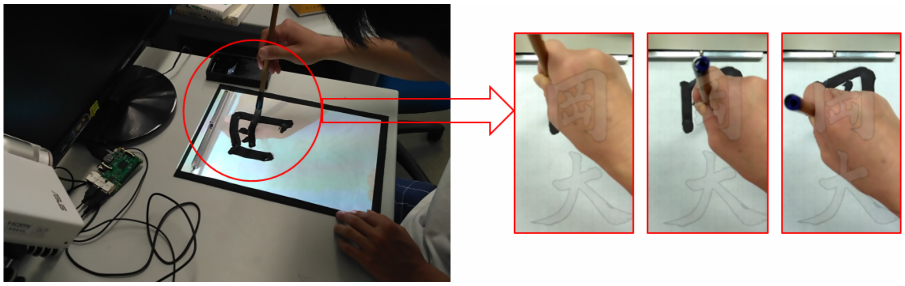


Figure 3.9: Hidden brushwork by teacher's hand.

3.7 System Hardware Cost

In this section, we discuss the hardware cost that was required to develop the proposed system.

Table 3.8 shows the hardware cost required to develop CLAS. The prices are referred from `https://www.amazon.co.jp/`. The total cost is relatively low, which is less than 100,000 yen. If a lower-specification model is selected for the projector, the cost can be reduced further. It is noted that all the software is free for CLAS.

Table 3.8: Hardware cost for CLAS.

Device	Price (Yen)
Raspberry Pi 3 B +	6490
Micro SD	990
Mini keyboard	1255
Mouse	462
Asus P3B projector	88600
Tripod	451
Total	98248

3.8 Summary

In this chapter, we presented the study of the application of *Raspberry Pi* to *calligraphy learning assistant system (CLAS)* using *projection mapping* for self-learning of *calligraphy*. The *letter portion practice function* was incorporated in CLAS to allow a learner to repeat the hard portions of each letter. The effectiveness of CLAS was verified through applications to students in Okayama University coming from various countries.

The future studies include the improvement of CLAS by tackling the brushwork cover-up problem by the teacher's hand, adding brushwork instructions, adjusting the playback speed, and implementing the evaluation method of the stroke quality. In the next chapter, we will present the application of *Raspberry Pi* to *air-conditioning guidance system (AC-Guide)*.

Chapter 4

Application to Air-Conditioning Guidance System (AC-Guide)

In this chapter, we present the study on the application of *Raspberry Pi* to the *air-conditioning guidance system* that encourages the proper use of air-conditioning.

4.1 Introduction

Nowadays, an *air conditioner (AC)* has been equipped at almost every house and office building around the world, to offer comfortable environments for humans and equipment in rooms. The quality of the indoor environment is critical to achieve healthy lives of tenants and proper functions of electronic devices inside rooms.

However, people often overuse the AC even at comfortable conditions. The AC overuse can lead to the *global warming* that has become the serious risk for sustainable societies. The *global warming* can lead to abnormal weather conditions and drastic shifts in climatic systems. The climate changes have affected human health and increased cases of diseases [55]. Therefore, the proper use of the AC is important to avoid unnecessary energy consumptions [56]

At the same time, the underuse of the AC can also cause troubles in current aging societies. If seniors stay in high temperature and humid environments for long time, they may suffer heatstroke, which can lead to the death [61]. According to the statics by the Japanese government, about $1,500$ people died because of heatstroke, where about 80% of them are 65 years old or older in 2019 [62]

In this chapter, we propose an *air-conditioning guidance system (AC-Guide)* to guide the proper use of the AC. Using *Raspberry Pi*, the *AC-Guide* periodically samples the temperature and humidity in the room to calculate the *discomfort index (DI)* there, and observes the AC use using a web camera. In addition to this *indoor DI*, the *outdoor DI* is calculated by obtaining the weather data from an API called *openweathermapAPI*. Then, by referencing to the message decision table, the system outputs a voice alarm message and an email alarm message to the registered persons.

4.2 System Overview

Figure 4.1 illustrates the system overview of the *AC-Guide*. Table 4.1 shows the hardware/software specifications.

A Web camera is attached to capture the LED on/off indicator in front of the AC controller attached on a wall. A light Web-server called *flask* is installed in *Raspberry Pi* so that the stored data

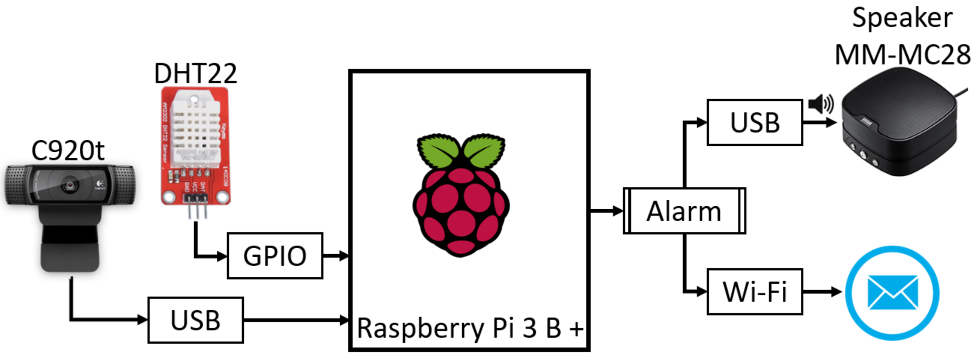


Figure 4.1: System overview.

Table 4.1: Hardware/software specifications of system.

item	specification
computer	Raspberry Pi 3 model b+
temperature/ humidity sensor	HiLetgo DHT22
web camera	Logicool C920t
speaker	Sanwa Supply MM-MC28
software	Python3, adafruit_dht, opencv3, omxplayer, flask

can be accessed from outside through the Internet. In our experiments, *Raspberry Pi* is connected to the Internet using Wi-Fi. The date and time are synchronized with the Internet for accurate records, because *Raspberry Pi* does not have the *Real Time Clock (RTC)* [57]. An email message is output from *Raspberry Pi* by using *Gmail SMTP*. Besides, a USB speaker is connected with *Raspberry Pi* to output a voice message.

4.3 System Implementations

The main functions of the system are periodically executed with the 10*min* interval. *Python* is used to implement the programs.

4.3.1 Periodical Functions

The system repeats the following procedure: 1) sampling the temperature and humidity data at the present time from the sensor using GPIO (General Purpose Input Output), 2) obtaining the weather data from an API called *openweathermapAPI*, 3) calculating indoor and outdoor DI, 4) taking photos of the AC control panel using a Web camera, 5) detecting the use of AC, and 6) outputting alarm messages with voices in five languages and with emails in English to the registered persons by referencing to the message decision table.

If DI is within the comfortable range and the AC is turned on, the system regards the *AC overuse*. The data is stored in the local storage of *Raspberry Pi* with CSV files that can be accessed from the Internet through *ssh*.

When the system runs at the first time after 23:50, it stops sampling data, closes the CSV file to save the data of the day, and sends the email message of notifying the information to the administrator of the system. Then, at 00:10, the system again starts sampling data for the next day.

4.3.1.1 Discomfort Index Calculation

The *discomfort index (DI)* is the factor to measure the human heat sensation under a room condition. In our studies, the DI is calculated by the following equation in [58, 59]:

$$DI = 0.81T + 0.01H \times (0.99T - 14.3) + 46.3 \tag{4.1}$$

where T and H represent the temperature (°C) and the relative humidity (%) of the room respectively. Table 4.2 shows the discomfort index values with the representative conditions.

Table 4.2: Discomfort index conditions.

DI	Discomfort Conditions
-55	cold
55-60	chilly
60-65	not feel anything
65-70	pleasant
70-75	not hot
75-80	slightly hot
80-85	hot and sweaty
85-	hot and irresistible

4.3.1.2 Weather Data API

The *outdoor DI* is calculated using the standard temperature and humidity in the open air. To obtain them, we use *openweathermap API* [60]. This API provides the standard weather data, including the temperature and humidity, at present time, at future, and also at past, in more than 20,000 cities across the world. In this work, we use the standard weather data at present time in Okayama city.

To access this API, the *API* key is necessary by creating an account on [60]. For the free plan, the maximum of 60 accesses to the API is allowed for one minute. The data is returned in the *JSON* format in default.

4.3.1.3 API Usage Procedure

The following procedure describes the steps to obtain the current weather data of Okayama city from *openweathermapAPI*:

1) Import the required modules by:
   `import requests, json.`
2) Authenticate to [60] with an API key by:
   $api\_key = ''The\_API\_Key''$
   $city\_name = ''okayama''$
   $base\_url = ''http : //api.openweathermap.org/data/$

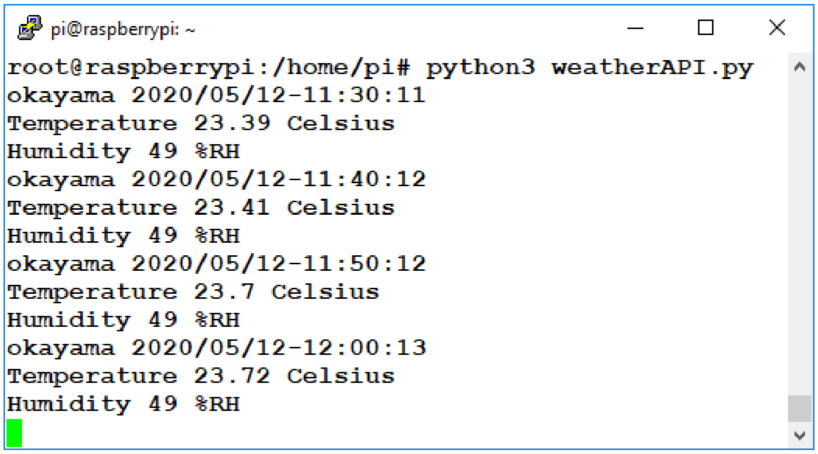
$2.5/weather?”$
$complete\_url = base\_url + ”appid = ” + api\_key + ”\&q = ” + city\_name.$

3) Use the requests module to send the API request by:
$response = requests.get(complete\_url).$

4) Convert the resulting JSON into a Python dictionary:
$x = response.json().$

5) Parse the data (temperature, humidity) from the query result by:
$y = x[”main”]$
$current\_temperature = y[”temp”]$
$current\_humidiy = y[”humidity”].$

Figure 4.2 shows the example of the outdoor weather data from the API. By obtaining the outdoor data, the outdoor DI is calculated using Eq. (4.1) that has been developed for the *indoor DI*.



```
root@raspberrypi:/home/pi# python3 weatherAPI.py
okayama 2020/05/12-11:30:11
Temperature 23.39 Celsius
Humidity 49 %RH
okayama 2020/05/12-11:40:12
Temperature 23.41 Celsius
Humidity 49 %RH
okayama 2020/05/12-11:50:12
Temperature 23.7 Celsius
Humidity 49 %RH
okayama 2020/05/12-12:00:13
Temperature 23.72 Celsius
Humidity 49 %RH
```

Figure 4.2: Weather data from API.

4.3.1.4 AC Use Detection

The use of the AC (AC turn-on/off state) is detected by checking the LED lamp at the control panel through the image processing using a Web camera, in Figure 4.3. Figure 4.4 shows the image processing procedure. The LED lamp is regarded as *on-state* when the intensity of the gray scale for the lamp image is larger than the given threshold. Otherwise, it is *off-state*. The current implementation of the *AC-Guide* does not use the set temperature of the AC, which will be in future studies.

4.3.1.5 Alarm Message Output

The system outputs a voice alarm message and an email alarm message to the registered persons by referencing to the message decision table. In a cold winter, the AC should be turned on to warm up the room. In a hot summer, the AC should be turned on to cool down the room. To keep the comfortable condition of the room is essential for seniors to avoid heatstroke or cold[61].

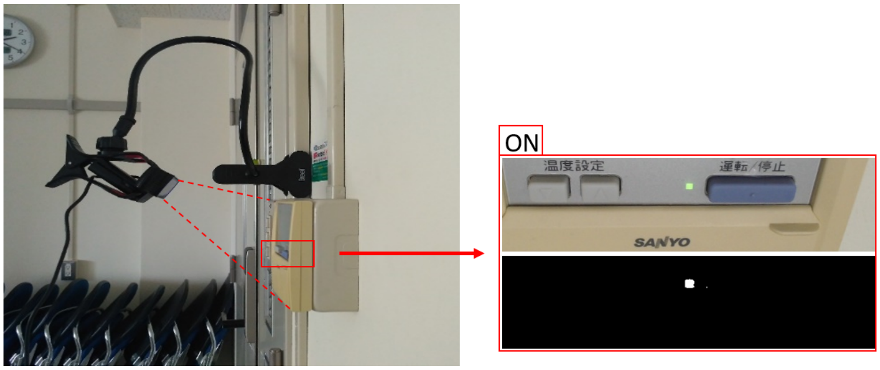


Figure 4.3: Web camera setup.

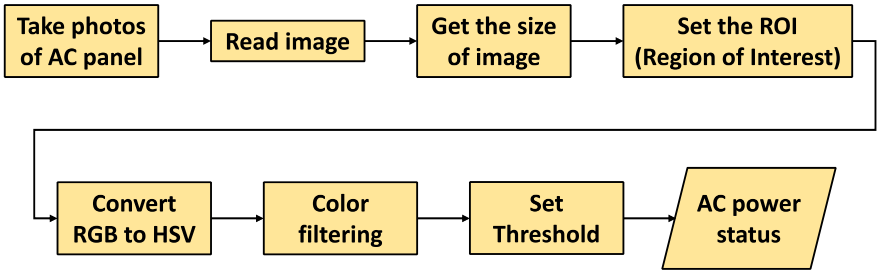


Figure 4.4: AC use detection flow.

When the DI stays in the uncomfortable range and the AC is turned off, the system outputs a voice/email alarm message of requesting turning on the AC. When the DI stays in the comfortable range and the AC is turned on, the system outputs a voice/email alarm message of requesting turning off the AC.

To prepare messages in different languages, the voice messages in Table 4.3 and 4.4 were recorded from international students in our laboratory. Currently, five languages, English, Japanese, Indonesian, Chinese, and Myanmar are supported.

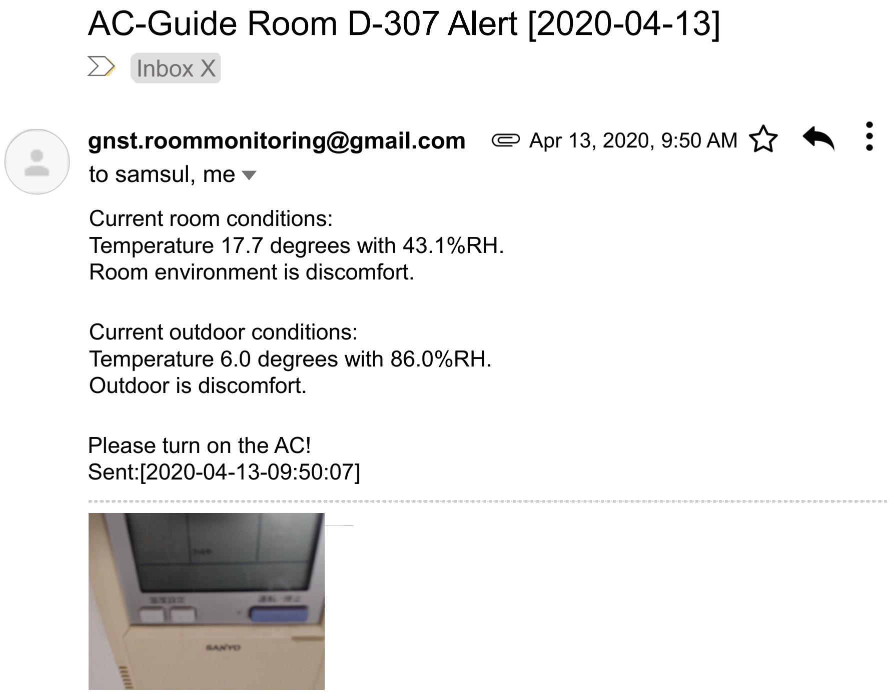
Table 4.3: Voice messages to turning on AC.

language	message
Japanese	エアコンをつけてください
English	Please turn on the air conditioner
Indonesia	Nyalakan pendingin ruangan
Chinese	Qing dakai kongtiao
Burmese	Kyaezuupyuywae aeyarcon ko phwintpaypar

Besides, the email message is sent to the registered persons who have the responsibility for the AC overuse. The message contains the date and time, the temperature and humidity of room, the indoor DI value, the outdoor weather data from the API, the calculated outdoor DI, the AC state, the photo of the panel, and the English text message, shown in Figure 4.5.

Table 4.4: Voice messages to turning off AC.

language	message
Japanese	エアコンを切ってください
English	Please turn off the air conditioner
Indonesia	Matikan pendingin ruangan
Chinese	Qing guanbi kongtiao
Burmese	Kyaezuupyuywae aeyarcon ko patepaypar



AC-Guide Room D-307 Alert [2020-04-13]

gnst.roommonitoring@gmail.com Apr 13, 2020, 9:50 AM
to samsul, me

Current room conditions:
Temperature 17.7 degrees with 43.1%RH.
Room environment is discomfort.

Current outdoor conditions:
Temperature 6.0 degrees with 86.0%RH.
Outdoor is discomfort.

Please turn on the AC!
Sent:[2020-04-13-09:50:07]

Figure 4.5: Example of alert email message.

4.4 Message Decision Table

Table 4.5 shows the table to decide the output messages. By the states of the three parameters, the indoor DI, the outdoor DI, and the AC use, it is decided to output the corresponding message or not. Table 4.6 describes the thresholds to decide the states of the three parameters. Table 4.7 describes the three alarm states: no alarm, alarm to turn off AC, and alarm to turn on AC.

4.5 Evaluations

In this section, we evaluate the effectiveness of the *AC-Guide* through simulations and applications in two rooms in No.2 Engineering Building at Okayama University.

Table 4.5: Alarm message decision table.

parameters			message
DI_in	DI_out	AC_use	
comfort	comfort	on	alarm (Turn-off AC)
comfort	comfort	off	no alarm
comfort	discomfort	on	no alarm
comfort	discomfort	off	no alarm
discomfort	comfort	on	no alarm
discomfort	comfort	off	alarm (Turn-on AC)
discomfort	discomfort	on	no alarm
discomfort	discomfort	off	alarm (Turn-on AC)

Table 4.6: Thresholds for state decision of three parameters.

input	conditions	value
DI_in	comfort	($63 \leq DI < 70$)
	discomfort	($DI < 63$ or $DI \geq 70$)
DI_out	comfort	($53 \leq DI < 70$)
	discomfort	($DI < 53$ or $DI \geq 70$)
AC_use	on	(detect.pixel ≥ 70)
	off	(detect.pixel < 70)

Table 4.7: Output for alarm control table

output	flag
no alarm	0
alarm (Turn off AC)	1
alarm (Turn on AC)	2

4.5.1 Simulation Results for False Message Avoidance

First, we evaluate the proposed system through simulating the data that caused the false turn-off messages if the *outdoor DI* was not considered. Figure 4.6 shows the simulation results for such cases by using the *outdoor DI* together to decide the output message. In the following, we will discuss each result one by one.

On Jan. 22, both the indoor and outdoor DIs were at discomfort. Even when the AC was turned on, no false turn-off message was output. On the other hand, when the AC was turned off, the turn-on messages were output.

On Jan. 23, when the AC was turned on, the outdoor DI was at discomfort, whereas the indoor DI was at comfort. Thus, no false turn-off message was output.

On Jan. 24, when the AC was turned on in the morning, the outdoor DI was at discomfort, whereas the indoor DI was at comfort. Thus, no false turn-off message was output. On the other hand, when the AC was turned on at 12:30-14:30, the turn-off messages were output, because both the indoor and outdoor DIs were at comfort.

On Feb. 10, both the indoor and outdoor DIs were at discomfort until 14:50. Thus, the turn-on messages were output. When the AC was turned on at 16:40, the outdoor DI was at discomfort, whereas the indoor DI was at comfort. Thus, no false turn-off message was output.

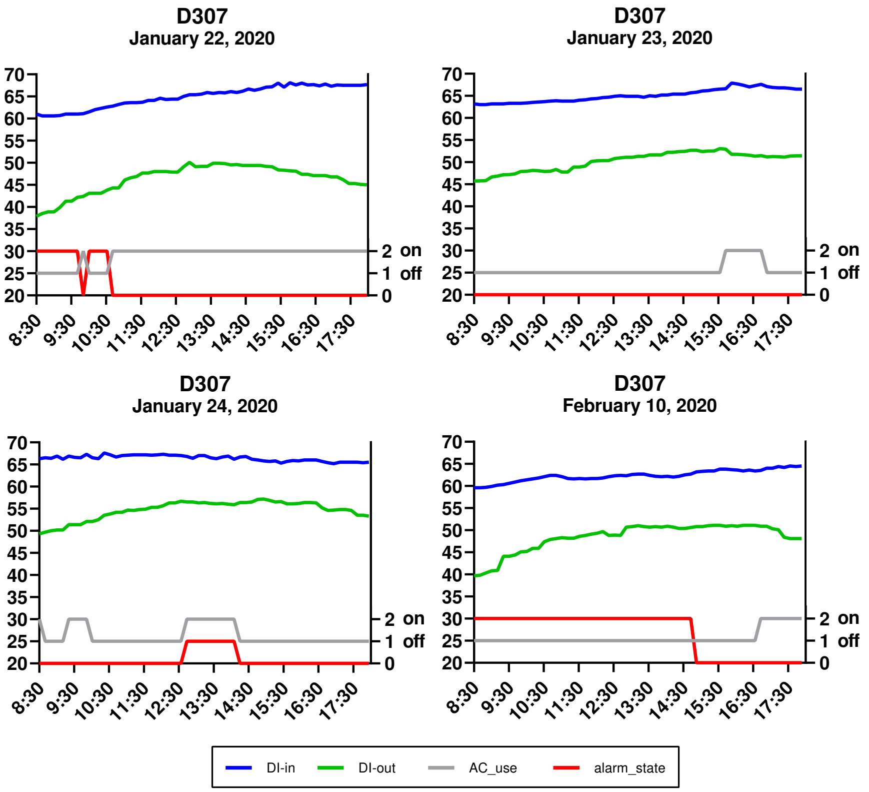


Figure 4.6: Evaluation results through simulations.

4.5.2 Application Results for AC Guidance

Next, we applied the proposed system in the two rooms, namely, D207 and D307, on April 13 and 14 in 2020 as the cold days, and on August 12 and 13 in 2020 as the hot days. During the day time, we turned on/off the AC by following the output messages of the system. Figure 4.7 and 4.8 show the results of them, respectively.

4.5.2.1 Application Results on Cold Day

On April 13, in D207, the turn-on messages are output until the AC is turned on at 10:50, when a student comes to the room. Then, the turn-on message is stopped. At 12:50, when the AC is turned off, the turn-on message comes back. Then, while the AC is turned on, it does not appear.

On the same day, in D307, the turn-on messages are output until the AC is turned on at 11:00. Then, the turn-on message is not output. After the AC is turned off at 12:40, the turn-on message is not output, because the indoor DI is at comfort. At 22:00, the turn-on messages are output again, because the indoor DI becomes at discomfort.

On April 14, in D207, the turn-on messages are output until the AC is turned on at 09:10, where a student comes to the room. At 12:10, the turn-off message is output, because both the indoor DI and the outdoor DI become at comfort. At that time, the AC is turned off, and no message is output until 22:00, because both DI are at comfort. At 22:00, the turn-on messages are output, because the indoor DI becomes at discomfort.

On the same day, in D307, the turn-on messages are output until the AC is turned on at 08:50.

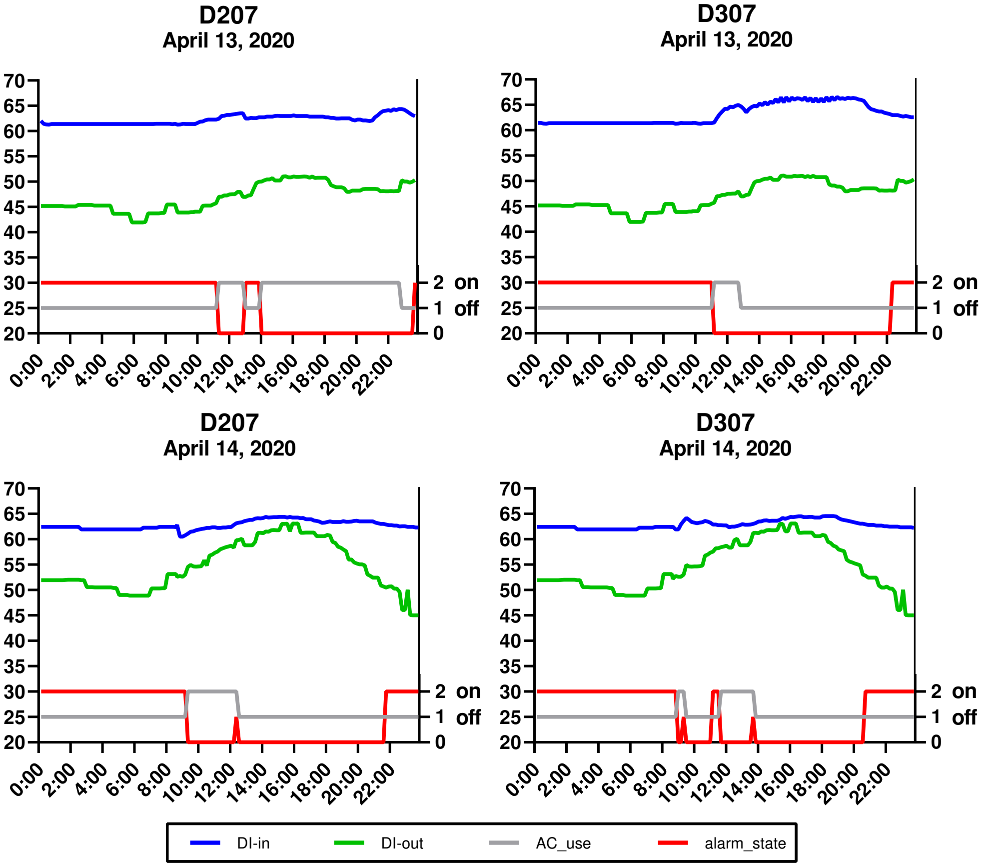


Figure 4.7: Application results on cold day.

After that, at 09:10, the turn-off message is output and the AC is turned off, because both the indoor DI and the outdoor DI become at comfort. The turn-on message is output and the AC is turned off at 11:10, because the indoor DI becomes at comfort. Then, no message is output. At 20:30, the turn-on message is output, because both DI become at discomfort.

4.5.2.2 Application Results in Hot Day

On August 12, in D207, the turn-on messages are output until the AC is turned on at 10:00, when a student comes to the room. Then, the indoor DI is quickly dropped, and the turn-on message is stopped. At 12:40, when the AC is turned off, the turn-on message comes back. Then, while the AC is turned on, it does not appear. After student leaves at 18:00, the turn-on messages are output.

On the same day, in D307, the turn-on messages are output until the AC is turned on at 11:00. Then, the indoor DI is quickly dropped, and the turn-on message is not output. After the AC is turned off at 12:00, the turn-on message comes back. Then, while the AC is turned on, it does not appear. After the student leaves at 17:50, the turn-on messages are output.

On August 13, in D207, the turn-on messages are output until the AC is turned on at 10:20, where a student comes to the room. At 13:40, the AC is turned off and the turn-on messages are output. Then, while the AC is turned on, it does not appear. After the student leaves at 18:40, the turn-on messages are output.

On the same day, in D307, the turn-on messages are output until the AC is turned on at 12:00.

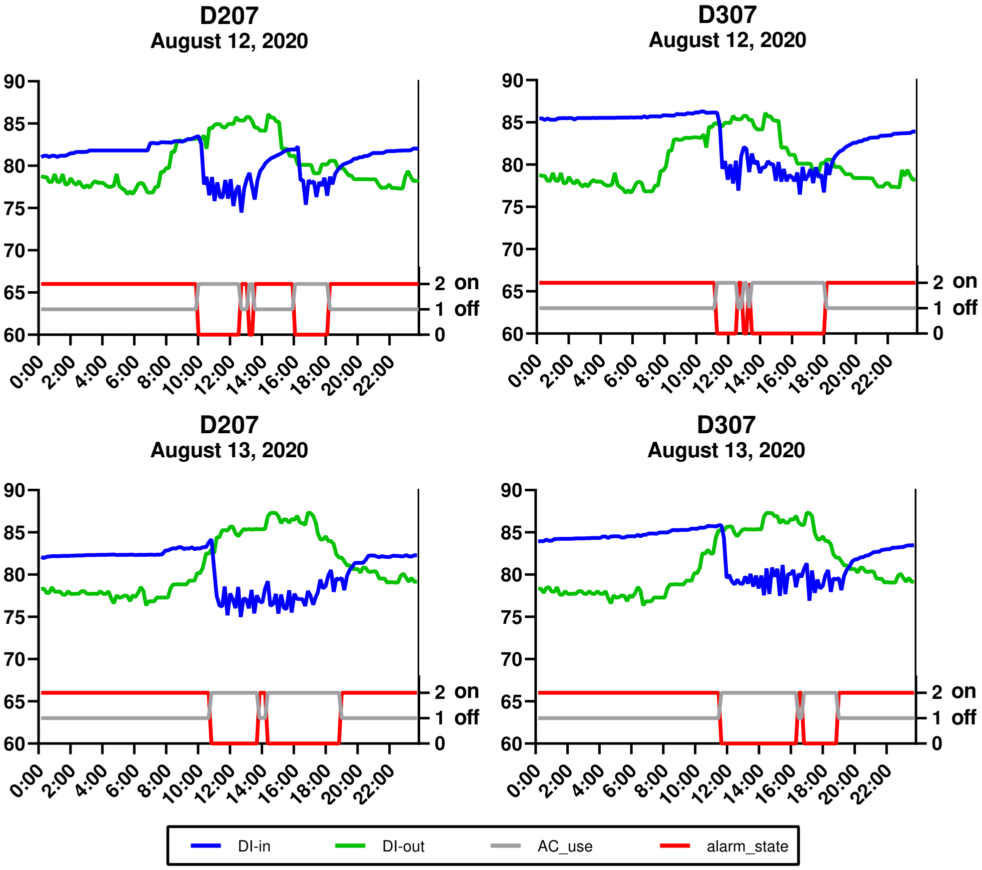


Figure 4.8: Application results on hot day.

After that, at 16:40, the AC is turned off and the turn-on message is output. Then, while the AC is turned on, it does not appear. After the student leaves at 18:50, the turn-on messages are output.

4.6 System Hardware Cost

In this section, we discuss the hardware cost that was required to develop the proposed system.

Table 4.8 shows the hardware cost required to develop *AC-Guide*. The prices are referred from `https://www.amazon.co.jp/`. The total cost is low, which is less than 40,000 yen. If lower-specification models are selected for the camera and the speaker, the cost can be reduced further. It is noted that all the software is free for *AC-Guide*.

4.7 Summary

In this chapter, we presented the study of the application of *Raspberry Pi* to the *air-conditioning guidance system (AC-Guide)* that guides the proper use of the AC. By referencing to the message decision table, the system properly sends both turn-on/off messages. The effectiveness of the *AC-Guide* was verified through applications in two rooms at Okayama University.

The future studies will include the improvement of the *AC-Guide* by considering the target

Table 4.8: Hardware cost for *AC-Guide*.

Device	Price (Yen)
Raspberry Pi 3 B +	6490
Micro SD	990
DHT22 Sensor	899
USB Webcam	11800
USB Speaker	16500
Total	36679

temperature of the AC, using multiple sensors in a room, adding the coordinated guidance of multiple rooms, counting the number of residents in the room for more proper messages, sending turn-on/off messages to other devices such as a fan, a humidifier, and dehumidifier, and continue experiments in various rooms. Besides, we will study the automatic control of the devices using remote controllers.

Chapter 5

Related Works in Literature

In this chapter, we briefly discuss related works to this study.

5.0.1 Related Works on CLAS

First, we will discuss them on CLAS. Several studies have been reported on assisting calligraphy writing practices with the computer aided technologies such as *augmented reality (AR)* and *virtual reality (VR)*, and on evaluating the quality of calligraphy.

[63] proposed a calligraphy writing system based on VR techniques. It adopts a force-reflection joystick as the input device, and develops a brush model that describes the brush skeleton and the geometrical-based deformation of the brush surface based on physics, to achieve realistic effects by simulating the haptic sensation similar to that of manipulating a real calligraphy brush. Through comparisons with a traditional calligraphy brush and a graphic tablet, the effectiveness is confirmed.

[64] developed a calligraphy learning system using a pen tablet on a PC. It focuses on the eight fundamental brush strokes in calligraphy, and compares each stroke using the coordinates of the starting and ending points of each stroke. However, the writing touch on a pen tablet is greatly different from that on a real paper with a brush.

[65] proposed a *Chinese calligraphic training system* based on VR. That is to say, a learner will not only acquire the brushwork rules, but also practice *calligraphy* as a mobile application in a virtual way, so that he/she may repeat practices without the consumption of papers and ink.

[66] presented a hierarchical evaluation approach of learning calligraphy. This method evaluates both the stroke sequence similarity and the character shape similarity between the characters written by a student and a teacher. As future works, we need to assess the quality of the stroke of a student.

[67] presented an AR system, which contains both the static writing path and the dynamic writing process of model characters. The learner needs to use a head mounted display to gain the visual information. However, the writing using a real brush and ink could be exceedingly different from that of the proposed system. It is noted that the brushwork cannot be substituted completely and accurately for a tablet or a touch screen of a smart-phone.

[68] introduced a *calligraphy-stroke learning support system* using a projector and a motion sensor called *Leap motion*. VR is used to show the static and animated models of characters on the paper. It provides visual information about the brush position obtained by a sensor. Unfortunately, it is not clear how to obtain the brushwork of a calligraphy expert.

[69] developed a VR-based calligraphy training environment to enable learners to acquire calligraphy skills intuitively. It adopts a Phantom device to allow a haptic interaction. The learner can pick up essential skills such as the appropriate tilting of the brush and the pen pressure. However, this system may create uncomfortable feeling to learners due to use of the extra device.

[70] proposed an interactive calligraphic guiding system to grade the score of written letters by using the image processing and fuzzy inference techniques. Three statistical features of locations, sizes, and widths of them are utilized to measure the score of calligraphic quality. That is, this tool can be used in our system to evaluate calligraphic quality, which will be included in future works.

[71] introduced a brush movement evaluation method for learners to understand the quality of brush movements without involvements of experts. It adopts a neural network technique to extract brush trajectories from a video stream, and compares them with the templates by experts to produce scores in the writing quality. In future studies, we will consider evaluating brush movements of learners.

[72] introduced a comprehensive evaluation method of Chinese calligraphy that involves global (whole character) and local (stroke) similarities. First, it identifies the candidate characters from the database according to the angular difference relations. Then, it matches the candidates with the most appropriate character. It considered various situations of positions, sizes, and tilt angles of characters without knowing what the character is.

[73] studied quantitative evaluations of calligraphy characters. The main evaluation focused on two parts, the straight section analysis and the curved section analysis. They are used to examine the distribution of the ink. Then, the system will recognize the contour line, and calculate the cut-point identification roundness index, the width index, the smooth index, and the ink ratio to extract the characteristics of the font shape numerically.

In our survey, most of the papers discussed calligraphy learning using virtual tools for practices, and did not discuss calligraphy learning using real *calligraphy tools* and a mini-computer as in the proposal.

5.0.2 Related Works on *AC-Guide*

Next, we will discuss related works on *AC-Guide*. Several studies have been reported on predicting the room temperature and developing an indoor monitoring system using *Arduino*.

In [74], Winter predicated that over the coming two decades, Asia will be the main driver of a 40% increase in global energy consumption, where around half of the energy consumption is associated with cooling or heating interior spaces by AC.

In [75], Chen et al. analyzed over 1.8 million measurements of AC power consumption and indoor/outdoor air temperatures in 129 houses in three cities in Australia from 2012 to 2014. They found that indoor temperatures for turn-on, turn-off, and operations of AC depend on average outdoor temperature for the previous seven days. They also suggested that more research will be required for better understanding thermal comfort and AC operation behaviors.

In [76], Yousif et al. studied to determine the thermal comfort or discomfort of people to the prevailing environmental conditions particularly temperature and relative humidity at Khartoum State in terms of Thom's discomfort index (DI). They found that less than 50% of the population experienced the sense of discomfort when the DI ranged from 22 to 24, and more than 50% suffered from discomfort when the DI ranged from 24 to 29.

In [77], Hintea et al. studied several machine learning methods for estimating cabin occupant equivalent temperature from a set of inexpensive cabin environmental sensors, including Multiple Linear Regression (MLR), Multiple Layer Perceptron (MLP), K-Nearest Neighbor (KNN), Multivariate Adaptive Regression Splines (MARS), Radial Basis Function Network (RBF), REP Tree, and Random Forest (RF), where RBF performed the worst and MLR outperformed all other techniques.

In [78], Kim et al. developed a definition of personal comfort models and proposed a unified modeling framework by establishing important concepts and methodologies based on prior thermal comfort research and machine learning best practice. They provided system architecture for the integration of personal comfort models in thermal controls, and described the potential role of standards in providing guidance to assure accurate and reliable performance of personal comfort models in real-world applications.

[79] developed a low-cost sensor network comprising of sensors in multiple locations in the room to monitor parameters that can help achieving the optimized control of the HVAC (Heating, Ventilation, and Air Conditioning) system in the room. It adopts *Arduino* as the microcontroller and several temperature and humidity sensors. Here, the difference of the decreasing rates at different locations shows the promising sign that the sensor nodes captured the spatial variations at various locations within the room.

[80] explored the thermal comfort of people in indoor environments through the *Internet of Things (IoT)* architecture. Using *Arduino Uno*, the system monitors the ambient temperature, the relative humidity, and the indoor wind speed. Those obtained data will then be used in the Fuzzy logic to control the load and provide people in the room with a relatively better thermal comfort experience under relatively better energy efficiency. It adopts the *Predicted Mean Vote (PMV)* as the thermal comfort indicator.

[81] developed a low-cost a low-cost wireless *Indoor Air Quality (IAQ)* system using *Arduino*. It allows to sense six air quality parameters, CO2, VOC (volatile organic compounds), temperature, humidity, CO, and Ozone. The sensor shield is used to integrate multiple sensors with their conditioning circuits and the shield is directly plugged into the standardized expansion headers on the *Arduino* board. The wireless communication with the mesh networking using the *Digi XBee* module is implemented for sensor data communications. Through comparisons with a professional-grade air quality measurement device, the effectiveness is confirmed.

[82] studied the automatic human comfort prediction using physiological signals directly collected from a building inhabitants. By using a number of sensors, including a thermal camera and several bio-sensors (galvanic skin response, hear rate tracker, respiration rate tracker), it records building's inhabitants under various thermal conditions (hot, cold, neutral), and consequently builds a multimodal model that can automatically detect the thermal discomfort. This model is expected to enable innovative adaptive control scenarios for the environment in real time, as well as a significant reductions in building energy usage.

[83] studied on characterizing the benefits in term of the summer comfort of the air movement created by a comfort fan in French climates. First, the conditions in which ventilation can provide a reliable level of comfort and can be an alternative to compressor based cooling systems, are investigated. By using experimental measurements of the air speed along with building simulations with the free temperature and humidity evolution, it was made possible to access inside climatic conditions when using a comfort fan. Then, it defines the general concepts of comfort fans. After that, it characterizes assessing the benefits of fans in terms of comfort and comparing it to air conditioners. It is found that occupants using comfort fans would face about 10% of discomfort hours during occupations and that comfort fans can suppress about 60% of discomfort by consuming four times less energy than a split system.

[84] introduced an intelligent approach that selects the most appropriate control among three existing strategies: state-feedback, PID, and ON/OFF controls, for the ventilation system control.

This approach considers the occupancy and the carbon dioxide concentration to select the most appropriate control. The whole concepts are implemented using a real-time platform, which includes IoT and Big data technologies together.

[85] developed the microclimate monitoring system in classrooms. The imbalance of microclimate parameters in classrooms can cause all kinds of health and work problems, affecting headaches, sleepiness, and tiredness resulting in lower concentration powers. This system measures microclimate parameters in classrooms including air temperature, humidity, and carbon dioxide level. It is noted that in a non-ventilated room the CO2 level was always above the recommended value (1,000 ppm) about 1,500 ppm, the temperature was slightly over the maximum recommended value (25 °C), and the relative humidity was slightly over the minimum recommended value (30 %).

[86] observed the variations of indoor temperature and humidity profiles of actual guest rooms equipped with *Occupancy-Based Climate Control (OBCC)* systems that were used to initiate a temperature set back to 15.6 °C in winter and to 26.7 °C in summer in the guest rooms. One year of one-minute temperatures and humidity data is characterized against the outdoor climate in heat losses and during the heating season due to the different dynamic heat balance conditions of the guest rooms. It is noted that potential discomfort appeared in the places with a stronger association between the outdoor and indoor temperatures. The guests who stayed in these rooms tended to set their thermostat at higher temperatures, which appeared to compensate the low balance-point temperatures of these rooms.

In our survey, most of the papers discussed indoor room monitoring systems to monitor the air quality level in the room, and did not discuss the guidance for the proper usage of AC to maintain a comfortable room and reduce energy usage as in the proposal.

Chapter 6

Conclusion

In this thesis, we presented our studies of two practical application systems in real world using *Raspberry Pi*, namely, the *calligraphy learning assistant system (CLAS)* and the *air-conditioning guidance system (AC-Guide)*.

Firstly, we reviewed the background technologies for implementing *Raspberry Pi* application systems in this dissertation. They cover *Raspberry Pi 3 Model B+*, *GPIO*, *Raspbian OS*, *Python*, *Python Libraries*, *SSH* and *WinSCP* to transfer files between *Raspberry Pi* and the computer, and the software development process of the *Raspberry Pi* application.

Secondly, we presented the study of the application of *Raspberry Pi* to the *calligraphy learning assistant system (CLAS)* for self-learning of *calligraphy*. Using the *projection mapping* technology, *CLAS* directly projects a letter writing video by a teacher on the paper on which a learner will write letters. Then, the learner is able to learn the stroke order and the writing speed in addition to the letter shape by tracing the writing of the teacher. Besides, the *letter portion practice function* is incorporated to allow a learner to repeat practicing hard portions of each letter. In the implementation of *CLAS*, we adopt a portable projector and open-source software for video processing and *projection mapping*. Through applications to international students from Japan, China, Myanmar, Indonesia, Bangladesh, and Kenya, we confirm the effectiveness in improving *calligraphy* skills for novice students.

Thirdly, we presented the study of the application of *Raspberry Pi* to the *air-conditioning guidance system (AC-Guide)*, to encourage the proper use of air-conditioning related devices/equipment by sending alarm messages of turning on/off them. *AC-Guide* periodically samples the temperature and humidity in the room to calculate the *discomfort index (DI)* there, and observes the AC use using a web camera. In addition to this *indoor DI*, the *outdoor DI* is calculated by obtaining the weather data from an API called *openweathermapAPI*. Then, by referencing to the message decision table, the system outputs a voice alarm message and an email alarm message to the registered persons. In the implementation of *AC-Guide*, we adopt a temperature/humidity sensor with *GPIO*, a USB Web camera, and a USB speaker, and open-source software for image processing. Through applications to two rooms in Okayama University, we confirm the correctness of the alarm messages.

These studies reveal that the hardware cost for the *Raspberry Pi* application system is relatively low, the developing period is less than one year only by one graduate student, and the performance of the system is acceptable for the practical use.

In future studies for *CLAS*, we will further improve the system by tackling the brushwork cover-up problem by the teacher's hand, adding the brushwork instructions, adjusting the play back speed, and implementing the evaluation function of the stroke quality. We will also increase the number of contents. Then, we will continue application experiments to various users.

For *AC-Guide*, we will improve the system by considering the target temperature of the AC, using multiple sensors in a room, adding the coordinated guidance of multiple rooms, counting the number of residents in the room for more proper messages, and sending turn-on/off messages to other devices such as a fan, a humidifier, and dehumidifier. Then, we will continue performance experiments in various rooms.

Bibliography

[1] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Comp. Netw., vol.54, no.15, pp. 2787-2805, Oct. 2010.

[2] T. Baranwal, Nitika, and P. K. Pateriya, “Development of IoT based smart security and monitoring devices for agriculture,” in Proc. Int. Conf. Cloud Syst. Big Data Eng., pp. 597-602. 2016.

[3] M. Maksimovic, V. Vujovic, and B. Perisic, “A custom Internet of Things healthcare system,” in Proc. Int. Conf. Inf. Syst. Tech., pp. 1-6. 2015.

[4] Q. H. Mahmoud and D. Qendri, “The Sensorian IoT platform,” in Proc. IEEE Ann. Cons. Comm. Netw. Conf., pp. 286-287. 2016.

[5] S. J. Hsiao, K. Y. Lian, and W. T. Sung, “Employing CrossPlatform Smart Home Control System with IOT Technology Based,” in Proc. Int. Symp. Comp. Cons. Cont., pp. 264-267. 2016.

[6] K. Hentschel, D. Jacob, J. Singer, and M. Chalmers, “Supersensors: Raspberry Pi Devices for Smart Campus Infrastructure,” in Proc. Int. Conf. Fut. Int. of Things and Cloud., pp. 58-62. 2016.

[7] Lukas, W. A. Tanumihardja, and E. Gunawan, “On the application of IoT: Monitoring of troughs water level using WSN,” in Proc. Int. Conf. Wir. Sens., pp. 58-62. 2015.

[8] S. Huda, N. Fahmi, A. Sudarsono, and M. U. H. A. Rasyid, “Secure Data Sensor Sharing on Ubiquitous Environmental Health Monitoring Application,” Jurn. Tekn., vol.78, no.6-3, pp. 119-122. 2016.

[9] Raspberry Pi, “The official website of the Raspberry Pi project,” `http://raspberry.org`. Accessed 1 May, 2020.

[10] S. Huda, L. Xiqin, N. Funabiki, and M. Kuribayashi, “A Proposal of Calligraphy Learning Assistant System using Projection Mapping,” IEICE Tech. Rep., vol. 118, no. 211, MVE2018-23, pp. 53-58, Sept. 2018.

[11] S. Huda, N. Funabiki, M. Kuribayashi, and W.-C. Kao, “A proposal of calligraphy learning assistant system using projection mapping,” in Proc. Int. Conf. Sci. Eng., pp. 10-15, Dec. 2018.

[12] S. Huda, N. Funabiki, M. Kuribayashi and H. H. S. Kyaw, “An Evaluation of Calligraphy Learning Assistant System using Projection Mapping,” in Proc. IEICE General Conf., K-022, pp. 82-83, Mar. 2019.

[13] S. Huda, N. Funabiki, and M. Kuribayashi, "An Implementation of Letter Portion Practice Function in Calligraphy Learning Assistant System using Projection Mapping," in Proc. FIT. Conf., K-022, pp. 351-352, Sept. 2019.

[14] S. Huda, N. Funabiki, M. Kuribayashi, and W.-C. Kao, "A Proposal of Calligraphy Learning Assistant System with Letter Portion Practice Function Using Projection Mapping," Accepted in Int. J. of Web Inf. Syst., Oct. 2019.

[15] S. Huda, N. Funabiki, M. Kuribayashi, and W.-C. Kao, "A Calligraphy Learning Assistant System with Letter Portion Practice Function Using Projection Mapping," in Proc. IEEE Int. Conf. Cons. Elec., pp. 1-2, Jan. 2020.

[16] S. Huda, N. Funabiki, M. Kuribayashi, R. W. Sudibyo, and N.Ishihara "A Proposal of Air Conditioner Overuse Alarm System Using Raspberry Pi," in Proc. IEICE General Conf., BS-1-16, pp. S28-S29, Mar. 2020.

[17] S. Huda, N. Funabiki, M. Kuribayashi, R. W. Sudibyo, and N.Ishihara "An Improvement of Air-Conditioning Guidance System with Outdoor Discomfort Index for Avoiding False Messages," IEICE Tech. Rep.,, June. 2020.

[18] Raspberry Pi, "Raspberry Pi in Education," `https://www.raspberrypi.org/education/`. Accessed 1 May, 2020.

[19] Raspberry Pi, "Raspberry Pi Foundation," `https://www.raspberrypi.org/about/`. Accessed 1 May, 2020.

[20] Raspberry Pi, "Raspberry Pi 3 Model B+," `https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/`. Accessed 1 May, 2020.

[21] Raspberry Pi, "GPIO documentations," `https://www.raspberrypi.org/documentation/usage/gpio/`. Accessed 1 May, 2020.

[22] Raspberry Pi, "GPIO documentations," `https://www.raspberrypi.org/documentation/usage/gpio/`. Accessed 1 May, 2020.

[23] Raspberry Pi, "Raspbian OS," `https://www.raspberrypi.org/downloads/raspberry-pi-os/`. Accessed 1 May, 2020.

[24] Python, "The official website of the Python," `https://www.python.org/`. Accessed 1 May, 2020.

[25] A. Kurniawan, "Raspbian OS Programming with the Raspberry Pi," Apress, 2019.

[26] G. Guillen, "Sensor Projects with Raspberry Pi," Apress, 2019.

[27] Adafruit, "DHT sensor library," `https://github.com/adafruit/DHT-sensor-library`. Accessed 1 May, 2020.

[28] Python Software Foundation, "OpenCV Library," `https://pypi.org/project/opencv-python/`. Accessed 1 May, 2020.

[29] Afternerd, “How to Send an Email With Python and smtplib,” `https://www.afternerd.com/blog/how-to-send-an-email-using-python-and-smtplib/`. Accessed 1 May, 2020.

[30] Hackaday, “Raspberry Pi : Python Flask web server,” `https://hackaday.io/project/4850-raspberry-pi-python-flask-web-server`. Accessed 1 May, 2020.

[31] Python Software Foundation, “Schedule Library,” `https://pypi.org/project/schedule/`. Accessed 1 May, 2020.

[32] D. Bader, “Schedule Library,” `https://schedule.readthedocs.io/en/stable/`. Accessed 1 May, 2020.

[33] D. J. Barrett, R. E. Silverman, and R. G. Byrnes, “SSH, the secure shell: the definitive guide,” O’Reilly Media Inc., 2005.

[34] T. Ylonen and C. Lonvick, “The secure shell (SSH) protocol architecture,” 2006.

[35] Raspberry Pi, “SSH (Secure Shell),” `https://www.raspberrypi.org/documentation/remote-access/ssh/`. Accessed 1 May, 2020.

[36] WinSCP, “WinSCP Application,” `https://winscp.net/eng/index.php/`. Accessed 1 May, 2020.

[37] Behind The Scenes, “Using WinSCP to connect to a Raspberry Pi,” `https://www.behind-the-scenes.co.za/using-winscp-to-connect-to-a-raspberry-pi/`. Accessed 1 May, 2020.

[38] Full Stack Python, “Text Editors and IDEs,” `https://www.fullstackpython.com/text-editors-ides.html/`. Accessed 1 May, 2020.

[39] Sublime, “Sublime Text,” `https://www.sublimetext.com/`. Accessed 1 May, 2020.

[40] Lukas, W. A. Tanumihardja, and E. Gunawan, “On the application of IoT: Monitoring of troughs water level using WSN,” in Proc. Int. Conf. Wir. Sens., pp. 58-62. 2015.

[41] S. Huda, N. Fahmi, A. Sudarsono, and M. U. H. A. Rasyid, “Secure Data Sensor Sharing on Ubiquitous Environmental Health Monitoring Application,” Jurn. Tekn., vol.78, no.6-3, pp. 119-122. 2016.

[42] T. Nakashima, “The synergy of positive and negative space in Japanese calligraphy,” J. Kinki Welfare Univ., vol. 8, pp. 113-119. 2007.

[43] A. Iezzi, “Contemporary Chinese calligraphy between tradition and innovation,” J. Liter. Art Stud., vol. 3, no. 3, pp. 158-179. 2013.

[44] T. Taylor, “Paper used in Japanese calligraphy,” AICCM Bulletin, vol. 30, pp. 51-58. 2007.

[45] A. Kanazawa, T. Asai, A. Minazuki, and H. Hayashi, “Construction of a MR space using projection mapping to promote hand hygiene,” in Proc. Int. Conf. Adv. Appli. Inform., pp. 247-252. 2014.

[46] L. Rodriguez, F. Quint, D. Gorecky, D. Romero, and H. R. Siller, “Developing a mixed reality assistance system based on projection mapping technology for manual operations at assembly workstations,” Procedia Comput. Sci., vol. 75, pp. 327-333. 2015.

[47] Y. Kushihashi and S. Mizumura, “Development of teaching material for robot mechanisms applying projection mapping technology,” J. Robot. Mechatron., vol. 29, no. 6, pp. 1014-1024. 2017.

[48] R. Krisjanis, “OfxPiMapper-Projection mapping tool for the Raspberry Pi,” MA thesis, 2015.

[49] R. Krisjanis, “OfxPimaper add-ons,” `https://github.com/kr15h/ofxPiMapper/`. Accessed 1 May, 2020.

[50] FFmpeg, “FFmpeg documentation,” `https://ffmpeg.org/documentation.html/`. Accessed 1 May, 2020.

[51] Inkscape, “Inkscape - draw freely,” `https://inkscape.org/en/`. Accessed 1 May, 2020.

[52] Japan visitor, “Shodo Japanese Calligraphy,” `https://www.japanvisitor.com/japanese-culture/language/japanese-shodo/`. Accessed 1 May, 2020.

[53] Projection Mapping Central, “Projection Mapping System,” `http://projection-mapping.org/what-is-projection-mapping/`. Accessed 1 May, 2020.

[54] ImageMagick Studio, “ImageMagick Application,” `https://imagemagick.org/`. Accessed 1 May, 2020.

[55] X. Wu, Y. Lu, S. Zhou, L. Chen, and B. Xu, “Impact of climate change on human infectious disease: empirical evidence and human adaptation,” Envi. Int., vol. 86, pp. 14-23. 2016.

[56] F. Wang, H. Yoshida, B. Li, N. Umemiya, S. Hashimoto, T. Matsuda, and H. Shinbayashi, “Evaluation and optimization of air-conditioner energy saving control considering indoor thermal comfort,” in Proc. Int. IBPSA Buil. Sim. Conf., pp. 88-95. 2009.

[57] V. Vujovic and M. Maksimovic, “Raspberry pi as a sensor web node for home automation,” Comp. and Elec. Engi., vol. 44, pp. 153-171. 2015.

[58] Discomfort index, “Discomfort Index Calculation,” `https://keisan.casio.jp/exec/system/1202883065/`. Accessed 1 May, 2020.

[59] M. Nakamura, S. Matsuo, S. Matsumoto, H. Sakamoto, and H. Igaki, “Application framework for effcient development of sensor as a service for home network system,” in Proc. Int. Conf. on Serv. Comp., pp. 576-583. 2011.

[60] Openweathermap, “Openweathermap API,” `https://openweathermap.org/`. Accessed 1 May, 2020.

[61] M. Asayama, “Guideline for the prevention of heat disorder in Japan,” Glob. Envi. Res., vol. 13, pp. 19-25. 2009.

[62] Ministry of Health, Labour and Welfare, `https://www.mhlw.go.jp/toukei/saikin/hw/jinkou/tokusyu/necchusho18/dl/nenrei.pdf`. Accessed 1 May, 2020.

[63] C. Liang, M. Hsieh, and K. Young, “A VR-based calligraphy writing system with force reflection,” Int. J. Autom. Smart Tech., vol. 1, no. 2, pp. 83-91. 2011.

[64] J. Shin, T. Okuyama, and K. Yun, “Sensory calligraphy learning system using Yongzi-Bafa,” in Proc. Int. Forum Strat. Tech., pp. 128-131. 2013.

[65] Y. Wu, Z. Yuan, D. Zhou, and Y. Cai, “A mobile Chinese calligraphic training system using virtual reality technology,” AASRI Procedia, vol. 5, pp. 200 - 208. 2013.

[66] M. Wang, Q. Fu, Z. Wu, X. Wang, and X. Zheng, “A hierarchical evaluation approach of learning Chinese calligraphy,” J. Comput. Inform. Syst., vol. 10, no. 18, pp. 8093-8107. 2014.

[67] R. Soontornvorn, S. Pongkarn, and H. Fujioka, “A development of AR calligraphy skill training system using dynamic font,” in Proc. Int. Conf. Broad. Wireless Comput., Commun. Appli., pp. 555-558. 2015.

[68] T. Matsumaru and M. Narita, “Calligraphy-stroke learning support system using projector and motion sensor,” Adv. Comput. Intell. Intell. Inform., vol. 21, no. 4, pp. 697-708. 2017.

[69] H. Nishino, K. Murayama, K. Shuto, T. Kagawa, and K. Utsumiya, “A calligraphy training system based on skill acquisition through haptization,” J. Ambient Intel. Human. Comp., vol. 2, no. 4, pp. 271-284. 2011.

[70] C. Han, C. Chou, and C. Wu, “An interactive grading and learning system for Chinese calligraphy,” Mach. Vision Appl., vol. 19, no. 1, pp. 43-55. 2008.

[71] P. Xu, L. Wang, Z. Guan, X. Zheng, X. Chen, Z. Thang, D. Fang, X. Gong, and Z. Wang, “Evaluating brush movements for Chinese calligraphy: a computer vision based approach,” in Proc. Int. Joint Conf. Artif. Intel., pp. 1050-1056. 2018.

[72] M. Wang, Q. Fu, X. Wang, Z. Wu, and M. Zhou, “Evaluation of Chinese calligraphy by using DBSC vectorization and ICP algorithm,” Math. Prob. Engin., vol. 2016, no. 4, pp. 1-11. 2016.

[73] Z. Wang, M. Liao, and Z. Maekawa, “A study on quantitative evaluation of calligraphy characters,” Comp. Tech. Appli., vol. 7, pp. 103-122. 2016.

[74] T. Winter, “An uncomfortable truth: air-conditioning and sustainability in Asia,” Environ. Plan., vol. 45, pp. 517-531, 2013.

[75] D. Chen, Z. Ren, and M. James, “What the indoor air temperatures in houses in three Australian cities tell us,” in Proc. Windsor Conf., April 2018.

[76] T. A. Yousif and H. M. M. Tahir, “Application of Thom’s thermal discomfort index in Khartoum state, Sudan,” J. Forest Prod. & Ind., vol. 2, no. 5, pp. 36-38, 2013.

[77] D. Hintea, J. Brusey, and E. Gaura, “A study on several machine learning methods for estimating cabin occupant equivalent temperature,” in Proc. Int. Conf. Inform. Cont., Auto. Robot. (ICINCO), vol. 1, pp. 629-634, July 2015.

[78] J. Kim, S. Schiavon, and G. Brager, “Personal comfort models - a new paradigm in thermal comfort for occupant-centric environmental control,” Build. Envir., vol. 132, pp. 114-124, March 2018.

[79] O. Bamodu, L. Xia, and L. Tang, “An indoor environment monitoring system using low-cost sensor network,” Ener. Proc., vol. 141, pp. 660-666, 2017.

[80] W. Sung, S. Hsiao, and J. Shih, “Construction of Indoor Thermal Comfort Environmental Monitoring System Based on the IoT Architecture,” J. of Sensors, vol. 2019, pp. 1-16, 2019.

[81] S. Abraham, and X. Li, “A Cost-effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications,” Proc. Comp. Science, vol. 34, pp. 165-171, 2014.

[82] M. Burzo, C. Wicaksono, M. Abouelenien, V. Perez-Rosas, R. Mihalcea, and Y. Tao, “Multimodal Sensing of Thermal Discomfort for Adaptive Energy Saving in Buildings,” in Proc. Int. Conf. IISBE Net Zero Built Envir., pp. 344-354, March 2014.

[83] L. Grignon-Masse, J. Adnot, and P. Riviere, “Environmental impact and discomfort relief of summer comfort appliances,” 2007.

[84] F. Lachhab, M. Bakhouya, R. Ouladsine, and M. Essaaidi, “Towards an Intelligent Approach for Ventilation Systems Control using IoT and Big Data Technologies,” Proc. Comp. Science, vol. 130, pp. 926-931, 2018.

[85] A. Kviesis, A. Klavina, and G. Vitols, “Development of classroom microclimate monitoring system,” Engin. for Rural Devel., vol. 16, pp. 719-724, 2017.

[86] H. Kim, and E. Oldham, “Characterizing Variations in the Indoor Temperature and Humidity of Guest Rooms with an Occupancy-Based Climate Control Technology,” Energies, vol. 13, no.7, pp. 1-21, 2020.