University of Sheffield

Multi-functional Smart Home Monitor Using a Raspberry Pi and an Android Phone

Longfu Sun

Supervisor: Fabio Ciravegna

A report submitted in fulfilment of the requirements
for the degree of MSc in Advanced Computer Science

in the

Department of Computer Science

September 6, 2018

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Longfu Sun

Signature:

Date:

Abstract

As people's safety awareness increases, increasing number of people, especially those who often travel, began to consider installing the home monitoring system. Traditionally, the function of the home monitor is too simple, which is nothing more than camera, warning, etc. It is not only expensive, but also the monitoring scope is limited. Actually, traditional home monitor has also exposed its problem of high price and poor information confidentiality.

This project plans to use low-cost device to design a mobile smart home monitoring device to solve the above problems. It not only has the basic functions of home monitoring, but also can be moved in many ways to expand the scope of monitoring. Increasing number of more comprehensive monitoring functions have been introduced, including harmful gas detection, temperature and humidity feedback,etc. Apart from the Raspberry Pi, this project introduces new hardware devices that allow the scalability of functions and provide better circuit design solutions. Use network programming technology to connect the device to an Android phone and provide user interface for remote control.

The related work is mainly the implementation of Raspberry Pi hardware cart function, Android app, and TCP/IP socket programming. Socket programming is also considered to be the focus and difficulty of this project. After analysis, the conclusion is that the difficulty of this project is mainly reflected in the comprehensiveness of its related technologies. There is no too much frontier technology, but students are required to have a strong programming ability. For embedded systems, a variety of programming language, network programming, electronic circuits must have a deep understanding.

Acknowledgements

First of all, I need to thank my supervisor Professor Fabio Ciravegna, whose valuable advice and creative ideas have provided my project with important help. Meanwhile, I need to thank my family for their support and encouragement, thank my Mother and my Father for helping me become who I am. Thanks to my friends, all of them has interesting soul. I need to thank my girlfriend. Without her, I could not overcome the academic pressure of this year. Finally, allow me to thank the Diamond Building, which I have come to almost every day of this year, for the study space and coffee it offers.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Human beings are always thinking about what the next level of technology will be, Artificial Intelligence (AI)? Cloud Computing? Big data or Internet of Things(IoT). Sumshanth interpreted the Internet of Thing is an “amazing new paradigm” in the field of computer science and the Internet[16]. He states that this technology involved to creates the interconnection via networking technology embedded in every objects, so devices are able to send and receive data.

Raspberry Pi, meanwhile, uses a powerful processor, the user-friendly Linux operating system. Combined with its cheap price and the characteristics of it that encouraging education, it was widely praised by the public after its release and was quickly popularized [18]. Users found that they could do almost anything by this hand-sized motherboard, such as programming, make a game box, router, TV media centre or use it to take videos. How does its work, and what kind of effect can be made, whether people is able to applied the characteristics of Raspberry Pi to the domain of IoT? Another facet worth considering is that of Android. Android is not just ahead of IOS as the most popular mobile operating system, actually, from as early as April 2017, StatCounter, which is an independent web analytics company, has issued a report stating that Google’s Android has surpassed Microsoft’s Windows operating system as the most popular operating system throughout the world[17].

1.2 Aim

With this technology in mind, I will attempt the following: Using the networking programming technology, mainly Socket programming, an Android mobile phone and a Raspberry Pi embedded micro-controller motherboard will be connected to build a foundation of internet data interaction equipment. Applying the existing technology, I will then try to implement a multi-functional smart home monitor. The final output of this project is two programs and a series of embedded hardware control scripts.

Students will state the whole implement process from the hardware circuit design, the assembly of the Raspberry Pi and GPIO hardwares to the Android app implementation. The end result is a highly implemented Raspberry Pi and a independent android app. The advantage of the system do not only include the principle of low-cost and strong security but also the functional comprehensive of the project is also stronger than similar products' obviously, which is includes four kinds of mobile modes to different working requirements (infared sensing probe tracking movements, ultrasonic sensor obstacle avoidance, human control movement, presetting path movement), detection of indoor temperature and humidity, harmful gas monitor, real-time video shooting by two camera(PiCam and USB camera), 2-dimensional rotationof camera holder, PIR anti-theft induction and email warning, MySQL database storage and selection. The system reduces the network delay and enhances the stability of the system. In order to verify the practicability of the system, detailed functional testing and comprehensive system testing were carried out.

Chapter 2

Literature Survey

Within this chapter, in order to make the best and most suitable selection of technology and devices, the field, applied technology, hardware devices and their alternatives will be described and compared in detail.

2.1 The Internet of Things

The Internet of things(IoT) can be defined from the broad sense and the narrow sense. In the broad sense, the Internet of things is a vision for the future development of network technology, to implement that people are able to use any network and any objects for information communication and transformation at any time and place. In the narrow sense, no matter whether there is a connection of Internet, as long as there exists a LAN that is implemented through sensors connection, it belongs to the range of Internet of things[1]. According to this two level of definition, people can issue the different expression of IoT. For example, Guo Bin[14] believe the Internet of things is a complementary network of the Ubiquitous Internet, because of the network is not limited in the human being's information communication. He states that, in essence, the information interactive entity of the IoT is various everyday things. However, there are also some people think the IoT can be understood as the future Internet, in the future everything has comprehensive perception ability and intelligent processing capacity.

Whether the IoT is a complementary network or a full replacement is not the main point of studies. Both of them is seems reasonable, because the so-called "comprehensive perception" is indeed main developing technology in the field of the Internet of things, which can be refined into the perception, acquisition, measurement and processing of data, including the RFID, the QR code, GPS, sensor network, Graphics-speech processing, etc[1].

2.1.1 IoT elements

Thus, what elements should be contained in a complete IoT system? In [15], Jayavardhana Gubbi divides various IoT component into three layers. The first layer consist of hardware, such as sensors, actuators and embedded system, a second layer is middlewares, which works for data transmission and analyze. Finally, some interpreter and visualization tools.

RFID

RFID(radio frequency identification) is an automatic information recognition technology that uses radio waves to reading and writing data[1]. The features of this technology include short-distance radio frequency identification(shorter than 100 meters) and the strong confidentiality and unforgeability. RFID has been widely applied in many fields, such as second-generation Chinese citizens ID card, the E-passport and mobile payment[2]. In fact, on the basis of RFID, there is another radio wave communication technology, which is NFC(near-field communication). Technically, NFC belongs to a subset of RFID, and their application is different, the RFID could reading and writing on one-way(tag are either active and passive). The relationship of RFID reader and tags can be one-to-many, but NFC's are fixed one-to-one[2]. Meanwhile, this relationship is similar to the relationship between TCP/IP and socket, NFC a subset of RFID which is encapsulated calculation logic.

WSN

WSN(wireless sensors network) can be defined as a network of many inexpensive sensors that work for sense external data and connect through wireless communication[3]. A intergrated WSN should include a sensor nodes collection, actuators nodes, a gateway to the external Internet as well as clients. According to different data consolidation forms(routing relationships between nodes), WSN can also be divided into tree-structure, star-structure as well as chain-structure[4]. No matter the structure of WSN, It is obvious that the data obtained from the end of the node can be transmitted by other nodes. So, how to reduce the interference and to improve the transmission efficiency is worth considering.

Thus, the WSN is a kind of inexpensive Internet of things technology, nodes can be placed by some cheap sensors, MCU as well as embedded system, etc. Meanwhile, nodes are part of data aggregation, which have the function of saving the resource. Finally, if there is a WSN that is being built, the scale of the system and the topology has a high degree of freedom. The location of the nodes can be placed and moved at will. The entire system is more flexible and reliable than other networks, which is also called as self-organizing[4].

IP addressing schema

The WSN system supports that a huge number of sensors and other types of machines that are connected to the network properly. Further, another problem appears, whether increasing number of machines mean bigger requirem -ents of IP address? For now, the answer should be yes, in fact, this problem has been improved by the strict approach of assigning the IPv4 IP address over the past 30 years. If IPv6's IP distribution will be widespread before the rapid development of IoT, people will not have to worry about its(due to IPv6 32 bit model, people have 340 undecillion[5]). However, if the IPv4 12 bit model still be used, the IP addressing schema is a considered problem.

2.2 Smart Home

Today, no one would question the influence of Invention of computer and the Internet, Similarly, the introduction of the Internet of things is interpreted as a third wave of technology. Swati Kashyap predicted that, in 2025, there will be 1 trillion devices are connected through a variety of technology[6]. Due to the IoT application, many industries are transforming, which includes intelligent household, agricultural information, wearable devices, self-driving cars, physical distribution, etc. Among them, the smart home can not be ignored.

2.2.1 What is Smart Home

The National Association of Home Builders(NAHB) has put forward the concept of home automation in 1975[9]. According to the technical level of that time, smart home is undoubtedly an unimaginable theory, no enough technology and theory to support this idea. However, with the popularity of Internet of things, not only the smart home become one of the most important application direction of IoT, but also people constantly put forward advanced and low-cost methodology. The smart home is mainly based on residential, implementing the intelligence and automation of information communication, security, convenience and other fields, and this application involves embedded system, automatic control, integrated wiring, network, computer vision, speech processing and so on. Combining with the understanding of the Internet of thing, the key point in home home is to solve the problem of how to connect various objects, appliances, processors, servers and terminal.

2.2.2 Smart home solutions

An early solution was put forward by a Scottish company, X10[8]. This technology enables domestic appliances or products to interact with each other, because it can convert electrical signals into commands. Obviously, this technique requires appliances to be connected by the wire, and because there may devices in the circuit, the interference is unavoidable. The reliability of the information transmission is severely limited. Gradually, wired product is take placed by radio wave technology. Such as the ZigBee, the Wi-Fi and the Bluetooth.

ZigBee

The ZigBee protocol is a low power LAN protocol based on IEEE802.15.4 standard, which is a type of mesh networking[10]. It is means that in a ZigBee network, every node can send and receive signals, and has a strong self-organizing ability. Meanwhile, The self-repairing ability of grid network architecture can also make Zigbee network stable and reliable. By contrast, the disadvantage is that ZigBee can not connect to the external network directly, a ZigBee network must have three kinds of equipments: terminal node, regulator and routers[10]. All of them can exchange, which means the function of an equipment is completely determined by the internal protocol stack. The role of the router is to complete the transition of the ZigBee network to the external Internet, routers do not have any use value but must exist, which actually increases the cost. Moreover, ZigBee's self-organizing ability is not very meaningful because a smart home Internet system does not require hundreds nodes.

Wi-Fi

At present, it seems that the wireless connection of each node in smart home based on Wi-Fi is more in line with people's habits, and people have adapted to enjoy conveniences brought by web server. Wi-Fi is a kind of built-in network protocol for the smartphones and the PC. It is very easy to be handled by people that acquiring and transmitting the information by a smartphone-side application or browsers. From 1998, a company, which is called MobileSTAR, began to officially provide commercial-grade WLAN services. By 2015, the number of wireless hotspots based on Wi-Fi technology all round the world has reached 70 million[11]. The development of Wi-Fi has become a new type of wireless network ecosystem and its functionality is not limited to the transmission of standard Internet data. According to its different work scenarios and different needs, including the IoT. In [12]Jatin Parekh think that, if the Wi-Fi technology is used for implementing the smart home, There will be more forms of networking available, and network forms of organization will be simplified. At present, if a smart home system is uses the ZigBee protocol, each node in this system must be has a ZigBee-supported chip, Zigbee technology obviously does not reach the popularity level of Wi-Fi. When designing the smart home system, the type of optional parts will be restricted. By contrast, Wi-Fi coverage is now almost ubiquitous.

Bluetooth

Unlike Wi-Fi, Bluetooth uses the IEEE 802.15 protocol standard and has evolved into four categories in three versions. Speed from the beginning of 1Mbps to 54Mbps[11]. So , for transmission rate, its ability has close to that of Wi-Fi. However, The most attractive thing about Bluetooth technology should be its strong robustness, low power consumption(BLE), and product differentiation[13]. Low power is the main reason why the Bluetooth is the best solution for some wireless devices, such as wireless keyboard, wireless mouse, etc. once open the Bluetooth equipment, it will continue to search, and try to keep connection with other devices, while guarantee the robustness. This kind of maintenance of system stability inevitably aggravates power consumption. Therefore, if Bluetooth wants to develop further, it must do better in terms of power consumption. In the smart home application, Bluetooth should be in the same logical position as Wi-Fi, which is used to connect the sensor's controller to the back end smartphone or PC. The Bluetooth communication frequency is about 2.4 GHZ like wifi, which belongs to ultra-high frequency. As we all know, this frequency is characterized by strong penetration, but limited distance.

Mixed

In fact, apart from what has been mentioned above, there are many other kinds of technologies and products that can be chosen for implement the wireless connection in smart home field, such as the Z-wave, Insteon, the Google Thread, Apple homeKit, etc. The intelligent smart home industry is still in a primary exploration stage, no technology can lead overall capabilities in others. So, the mixed system may also be an acceptable choice. For example, in the smart home sensor layer, people can introduce the ZigBee technology, using its strong networking organization capabilities. More sensors can be connected together, then, the retrieved data will be transmitted to a middle node for processing by the routing technology. The processed data is accessed through a gateway and interacts with a smartphone via Wi-Fi or Bluetooth. For the purpose of this project, the Raspberry Pi is an acceptable microprocessor, because the Raspberry Pi has built-in Wi-Fi module and the Bluetooth module, and the connection will become relatively simple. The operating system of Raspberry Pi allows developers to connect the Raspberry Pi to an external network just like a wireless internet connection of PC via some Linux commands. By contrast, if using some kind of low integrated level single-chip, such as the Arduino or 51 series microcontrollers, corresponding chips must be required, such as the cc2539ZigBee development module, the HC-05 Bluetooth module and ESP8266Wi-Fi module.

SMTP

SMTP(Simple Mail Transfer Protocol) is a Internet transfer protocol, which is not closely related to the IoT in the traditional cognitive, but an protocol that is able to transfer the email from source address to the destination address. However, after reading previous papers, student found that this protocol was widely applied in the field of IoT, especially the smart home alarm or other functions like send sensor data to clients that are easier for users to view. Similar to the HTTP protocol, the Python programming language also has a built-in library of SMTP(smtplib). Another viable option is gammu Python library, this library works for send 3G SMS messages to mobile phone[52]. However, in comparison, sending email by SMTP is a better solution for reminding users that the IoT system is processing some data, or alerting. Because the smtplib library is not only able to send plain text email, but also hyperlinked or attached message[53]. Meanwhile, sending SMS message by embedded system is more complex, the additional 3G module is required, which is also not in line with the low-cost principle.

There are two ways to implement sending mail by Python, if SMTP-enabled service are installed in localhost, the IoT system can be directly treated as the host of SMTP, if not, introducing the SMTP service from other providers, such as Gmail or outlook.

2.2.3 Low Cost

People pay special attention to two characteristics of a smart home system, one is low cost and another is security. Some people question: a smart home is nothing more than a series of embedded chips, why strictly pay attention to the energy consumption of single-chips and sensors modular? The answer has been be mentioned above, in the future, there will a greater number of nodes that connected to the network, and the dependence on energy will be even greater. For this reason, due to unnecessary cost of wiring and the low-level network organizing ability, The wired Internet of things has been phased out by the wireless Internet.

The principle of low-cost can be reflected in two aspect, which are the low-cost of building a smart home and the efficiency of using a smart home system. In other words, reducing cost or improve efficiency on the premise of ensuring the service quality of smart home system. What has been mentioned above that if the ZigBee is applied, there must have an extra node to act as a processor, which do not have any using value for users[10]. The existence of this node undoubtedly aggravated the consumption of the system. Correspondingly, there is no such problem with Wi-Fi and Bluetooth. However, according to [49], in order to guarantee the transmission rate. Wi-Fi consumes about twice as much energy as ZigBee. So, ZigBee has no ability to replace Wi-Fi and is further promoted.

In addition, the concept of low-cost should be embodied in the topology structure of WSN. Common topologies includes mesh, tree and star. For every node of tree structure, The transmission path and transmission speed of data are restricted by the tree structure(leaf node data must go through their parent nodes)[41]. This transmission method wastes more energy and time. In contrast, the transmission efficiency of tree structure is the lowest. Finally, the low-cost principle should also be reflected in program design. For example, many smart home systems have a toxic gas detection function. If the program that working on monitoring the gas is always running in the back-end of system, this is undoubtedly a waste of resources. A method that can be adopted is set a timer, for example, run this program automatically every one or two minutes. After the test is completed, the resources are released and the program sleeps.

2.3 Micro-controller Motherboard

No matter there is a simple STC51 series microcontroller chip, the Arudino embedded development board, or the Raspberry Pi, this project is looking for a suitable hardware product as the Core and the control unit of smart home system. At the moment, it does not seem necessary to use the most rudimentary and backward 51 series single-chips, but in this section, serveral popular embedded micro-controller motherboard will be described in detail, to choose the best solution according to their equipped function modules and the project's modular programming.

2.3.1 Arduino

Addimulam(2015) said that, as a kind of famous micro-controller motherboard, Arduino is better suited to executing one program once over and over again, while Raspberry Pi is more suitable for run multiple programs[16]. Due to the fact that this project involves into multiple extension module controls and connections, Raspberry Pi may be more appropriate. In addition, the Raspberry Pi's factory image file already contains the IDLE of C++ and Python, developers are allowed to operate and write the software directly, after rewriting the image file to the SIM card by ordinary computer and connecting the keyboard and screen. Taking into account the developer's programming experience, they can even use the command line below:

$$Sudo \quad apt-get \quad install \quad nameOfIDE \tag{2.1}$$

to download some integrated development environments, such as Spyder or WinPython, which could significantly reduce the programming difficulty and error rate. On the contrary, if developer want to write Python program in Arduino board. The configuration process is troublesome. Developers have to write Python program in other computer, and install PySerial, which is a

Python API module, to read and write serial data in the Arduino board[17].

2.3.2 BeagleBone Black

Another possible choice is BeagleBone Black (BB Black), which is another popular low-cost, community-supported development platform. In fact, compar -ed to Raspberry Pi, the performance of BB Black is better in many aspects. For example, BB Black has a higher GPIO capability, which means that it could carry more external output/input devices. In addition, for prototype board hardware development, BB Black follows the principles of open source[18], the benefit of which is that it is easy for beginners to learn and modify the programming. 1GHZ processor running speed also means faster processing speed. Finally, the its unique Bonescript can be compiled in Cloud9 programming environment[19]. In contrast, the Raspberry Pi is more capable of handling multimedia data. Mr Michael Leonard analyzed the various indicators of this two kind of microcomputers. In his blog, the compared data has been list as Figure 2.1. According to this comparison, Raspberry Pi is shown to have an RPi camera connector and 2 USB hosts. With that in mind, for cameras and other external devices, there are more options available to users.

	BeagleBone Black	Raspberry Pi
Base Price	45	35
Processor	1GHz TI Sitara AM3359 ARM Cortex A8	700 MHz ARM1176JZFS
RAM	512 MB DDR3L @ 400 MHz	512 MB SDRAM @ 400 MHz
Storage	2 GB on-board eMMC, MicroSD	SD
Video Connections	1 Micro-HDMI	1 HDMI, 1 Composite
Supported Resolutions	1280×1024 (5:4), 1024×768 (4:3), 1280×720 (16:9), 1440×900 (16:10) all at 16 bit	Extensive from 640×350 up to 1920×1200, this includes 1080p
Audio	Stereo over HDMI	Stereo over HDMI, Stereo from 3.5 mm jack
Operating Systems	Angstrom (Default), Ubuntu, Android, ArchLinux, Gentoo, Minix, RISC OS, others…	Raspbian (Recommended), Ubuntu, Android, ArchLinux, FreeBSD, Fedora, RISC OS, others…
Power Draw	210-460 mA @ 5V under varying conditions	150-350 mA @ 5V under varying conditions
GPIO Capability	65 Pins	8 Pins
Peripherals	1 USB Host, 1 Mini-USB Client, 1 10/100 Mbps Ethernet	2 USB Hosts, 1 Micro-USB Power, 1 10/100 Mbps Ethernet, RPi camera connector

Figure 2.1: the Raspberrry Pi constrasts with BeagleBone Black [42]

2.3.3 Raspberry Pi 3

The lastest model is Raspberry Pi 3 Model B+, which was issued in July of 2017. This model has 40 GPIO pins header(general purpose input output). People hold the idea that, compared with the previous model, the most significant progress of this model is Wi-Fi modular and Bluetooth 4.1 have been integrated on board. In this case, it is not only a simple replacement of desktop system,but also the a key point to in the construction of an internet of things(IoT) system[20]. The real figure of Raspberry Pi has been shown in Figure 2.2:

Raspberry Pi has as strong application in regards to education, and the official recommended programming language is Python. Meanwhile, because of its basis in education, the makers of Raspberry Pi advocate for kids study

"Scratch" programming language using this microcomputer. For GPIO progra -mming, the most popular tools are C++ "writingPi" library and Python "RPi.GPIO" library[21]. All of the control of the GPIO in this project is implement by Python, actually the name of Pi in the Raspberry Pi is inspired by the 'Python' word[22], which is also indicate the importance of Python to the Raspberry Pi.

Python controls the GPIO of Raspberry Pi is implemented by "RPi.GPIO" library, which has been pre-installed in latest Raspberry Pi before published. The basic functions of the library have been listed in table 2.1. People just need to launch Python IDE in Raspberry Pi and using "import RPi.GPIO" command to import the related packet. The functions in this package are very easy to be leant and applied even by beginners.

Figure 2.2: Raspberry Pi 3 Model B+[48]

Table 2.1: Basic Function of RPi.GPIO Library

GPIO.setmode(GPIO.BCM/GPIO.BOARD)	Activating the Broadcom-chip specific pin numbers.BOARD: using the physical pin numbers BCM: using "Broadcom SOC channel"
GPIO.setup(channel,GPIO.IN/GPIO.OUT)	Set pins as "input" or "output"
GPIO.cleanup	Release resources after use.
GPIO.output(channel,GPIO.LOW/GPIO.HIGH)	Set the output level of pins, before driving some external device
GPIO.input()	When data need to be read from external device, set pins as input
GPIO.PWM(channel,HZ)	Raspberry Pi support PWM output. In this project, this output model will be use to control the the servo motor and DC motor.

2.4 Other Hardware Components

In essence, the Raspberry PI and the BeagleBone Black are ARM-based micro motherboards, which works for calculate data or process events. However, as with other ordinary computers, if there is no input-output devices, people can not use the Raspberry Pi for anything. In following section, the alternative hardware devices will be introduced to address this issue, including the develop -ment board, sensors, camera, and motors.

2.4.1 Development Board

The development board is an additional chips, which can be mounted to expand the computer's capabilities. This set-up is also called "add-on" board or an “Expansion board”. The development board can be considered as an embedded breadboard, in which is installed various drivers and serval expansive module. In line with what was mentioned above, the Raspberry Pi does not have expected middlewares, whom is very useful for hardware development, such as the motor driver module or Analog-to-Digital converter module. In this case, many hardware devices can not compatible with the Raspberry Pi directly, especially some sensors that output analogue signals. In addition, the Raspberry Pi only has 40 GPIO pin. Ignoring several GND and 5V VCC ports, the ports that can be used for input-output for data transmission is very few. However, more pins are required to link the motherboard and other components(each hardware piece has 2 or 4 pins). In order to solve the mentioned two requirement, many hardware manufacturers design PCBs (print circuit boards) for their own micro-controller motherboards . Some of their product are very famous, such as the sense HAT, the Explorer HAT and the RaspiRobot Board. One development board that is very suitable for this project is BST-4WD, which was design by chinese company Yahboom.

The BST-4WD development board provides interfaces and drivers for various modules, including the driver for of four-wheeled DC motors, tracking sensors, infrared obstacle avoidance sensors, gray level sensors, LED searchlights, servo motors, ultrasonic sensors and so on. The pins on these modules cannot be changed. For example, the four-wheeled DC motor will use six pins, which are GPIO 20, 23, 19, 26, 16 and 13. When the DC motors need be used, the Python program needs to declare the adjustment of these six pins. Although these pins are fixed, it does not mean that each GPIO port can only complete one kind of specific task. For example, in this project, the MQ-2 is not be allocated to a specific module, and we find that the MQ-2 and servo motors have similar requirements as for their number of pin(only an I/O pin, one GND and one VCC). So, we can connect MQ-2 to Raspberry Pi by servo motor module,which will not have any negative influence on the function of the servo. Therefore, in a limited range, we can expand the function of the BST-4WD development board. In addition, the other strength of the BST-4WD is that this development board is not only adapted to the Raspberry Pi, but also provides interfaces for the Arduino and STC 51 series micro-controllers[23]. The application of the development board will obviously alleviate the difficulty of the circuit design and shorten the time needed for the project. The whole project will not require too many jumper wires or additional motor driver and converser to be used around the product. Meanwhile, the incidence of peripheral circuit faults will also be reduced, lowering the chances of a short circuit, open circuit or circuit wear. the BST-4WD development board has been shown in Figure 2.3.

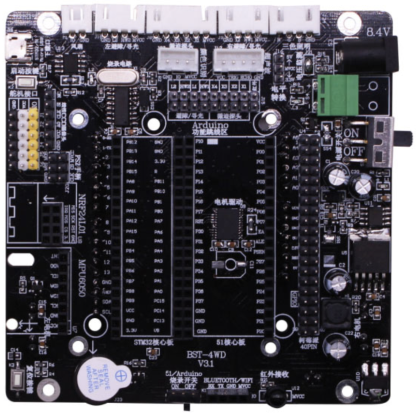


Figure 2.3: BST-4WD development board[43]

2.4.2 Sensors

Temperature and Humidity Sensor

DHT-11 can simultaneously detect the temperature and humidity of the surr -ound environment. The connection between the DHT-11 sensor and the Raspberry Pi is very simple, and only one data port is needed. However, the data structure of the Raspberry Pi when it receives data is not same as which is obtained by the other sensor. The DHT-11 sensor uses a single bus data format, which means that both input data and output data are trasmitted using one pin. The output data package consists of 5 Byte (40 Bit). Accurately speaking, this 40 Bit includes 8 bit integer humidity value(byte 4), 8 Bit decimal humidity value, 8 Bit integer temperature value, 8 Bit decimal temperature value and 8 Bit checksum (byte 0)[23].

Our ultimate goal is to identify temperature and humidity values from this sensor and use them as two independent values, rather than having only an unreadable data package. If any external library method is not used, the solution is that, after each reading of the data, we would need to extract the 8 bit data from each data package once until the end of data. This method can solve the problem, but the external library method is a more reliable solution.

The introduction of the Adafruit library method makes it more convenient to connect some specific electronic components, such as some display devices(LED matrix and LCD display), sensors, and PWM controls[25]. For this project, firstly we need install the git server into the Raspberry Pi, and then clone the Adafruit library with following command:

$$git\ clone\ http://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code.git \quad (2.2)$$

as a result, we can apply the related library method in Python programming, and read the value of humidity and temperature by function: .retry (sensorNumber, GPIONumber)

HC-SR04 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor and a servo motor will be used to implement the function of obstacle avoidance movement. The servo motor is used to help the HC-ST04 sensor rotate roughly 180 degrees, making it possible to Change the angle to check whether there are any obstacles in the other direction, after finding the obstacle on one way.

The HC-SR04 is the key device of the above-mentioned function. The working principle of the ultrasonic sensor is mainly related to the sending and receiving of square wave signals, The HC-SR04 has four pins, apart from common VCC pin and GND pin, there is a trigger pin, called “Trig”, and a receiver pin, called “Echo”. After the sensor is powered, “Trig” sends 8 square waves(40KHZ) automatically. The Raspberry Pi Python programming sets a timestamp for this moment, and “Echo” will be waiting for the reflector. When the reflector is received by “Echo”, a second timestamp will be set. Now two timestamps make a difference value(T), T is the time that system keep high level. Using the beloved format, the distance between the obstacle and HC-SP04 (D) can be calculated using T and ultrasonic spreading velocity in air (S=340m/s)[26]

$$D = T * S/2 \tag{2.3}$$

Tracking Sensor

The tracking sensor is a kind of light sensitive sensor, able to detect if light is reflected or absorbed in the area in front of it[27]. This sensor will be used to allow the product to move following a line.

The Raspberry Pi will enable it to identify the bending path of the line and take some corresponding measures allowing it to change moving direction. In order to implement these functions, roughly three to five sensor are required, increasing the number of tracking sensors generally leads to a better solution. Meanwhile, a black line will serve as a path. The effect of its use that not only that the number of taken pins is reduced(four sensor only need one VCC and one GND total), but it then becomes also able to fit into the pin number which is allocated by the development board. People have to make sure that the four sensing heads are located in a straight line, perpendicular to the Raspberry Pi car. Meanwhile, we also used an black insulating tape instead of a curved closed track(black line).

The control of tracking sensor is that: the Raspberry Pi processes a reading operation for its tracking sensors, When the sensors detect the insulating tape (when they see no light detected), port level is low; and when there is no insulating tape be detected, the port level is high. Further, we can perform a logic operation on the output level of four tracking sensor.

- if Right 2 and Left 2 is low(false) level, the Raspberry Pi car is moving on the center of tracking, remian go straight
- if Right 2 is low(false) level, while Left 2 is high(true), the car is on the left, turn right slightly
- if Left 2 is low(false) level,while Right 2 is high(true), the car is on the right, turn left slightly
- if Left 1 is low(false) level, the car is serverely shifted to the right, making a great turn left
- if Right 2 is low(false) level, the car is serverely shifted to the left, making a great turn right

Meanwhile, there are a series of special circumstances. Doing the following operation can improving the correctness of the functions.

- if Right 1, Right 2 and Left 2 are low(false) level, the car may be passing a 90° right bend, turn right.
- if Right 2, Left 2 and Left 1 are low(false) level, the car may be passing a 45° left bend, turn left
- if all sensors's head are low(false) level, The car may parallel to the path,turn left or right 90 degree to move forward in the original direction, or return.
- if all sensor's head are high(true) level, The car is completely derailment.stop at original place, which can be regarded as original stage.

Harmful Gas Sensor

MQ-X series sensors are frenquently used for testing harmful gases, including MQ-135,MQ-307a, etc. The basest one is MQ-2, which is able to detect methane, butane, LPG and smoke in air[28]. The working principle of the MQ-2 sensor is complex, which involves into the reaction of tin-oxide under heating conditions and the change in semiconductor conductivity[29]. But, we only just need to focus on how to using this sensor.

After the development of this project, the expected application effect of MQ-2 will be: when starting the product, the MQ-2 module will run automatically(it needs 10 second of warm-up), the Raspberry Pi will monitor the environmental data which is detected by MQ-2. If one of methane, butane, LPG or smoke is found in the air, the normal running will be interrupted, and the Python program will send an alarm signal to the Android client. The client jumps to the warning view-page immediately and the Android phone will start vibrating.

PIR

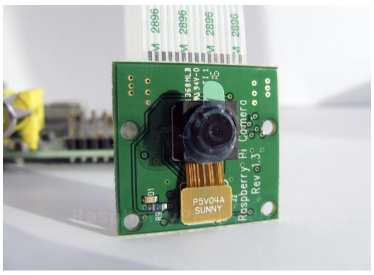
The human motion is regarded as a kinds of electronic switch and has been effectively used in various fields of IoT, the model of the PIR generally used for embedded development is HC-SR501 infrared pyroelectric sensor, the most obvious feature of this sensor is a smooth white glass cover, but actually it is a Fresnel lens. This component has two function, the first one is amplify the infrared signal, the infrared signal that heat-released via human body can be refracted to sensor. The second function is to sense the variable thermoluminesc
-ence infrared signals in various areas of the monitoring range, so that PIR generate electric signals and improve the sensitivity[54].

The main sense principle of the PIR can be divided into three parts, first of all, there is a high frequency transmitter, make the electrical circle around to the high frequency magnetic. Further, the living creature is close to the magnetic field, the eddy current will be generated on the surface of creature, which will trigger the reverse magnetic field. Thirdly, the generated reverse magnetic field will offset the high frequency magnetic field. If the offset occurs, the transmitter will stop to vibrate, the PIR sensor output a high/low level by monitoring whether the transmitter is vibrating. Therefore, the essence of PIR is the doppler effect.

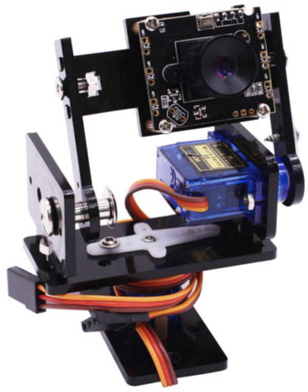
2.4.3 Camera and Motion

For the camera, there are two possible selections. The first one is the PiCamera, which is provided and supported by the official of the Raspberry Pi. This camera can be used directly after configuring the ".config" file. Secondly, we could use some kind of USB camera. The advantage of using a USB camera is that the user can choose various different cameras according to their requirement, such as the High-Definition camera and the waterproof camera. The disadvantage is that not all USB cameras are supported and compatible with the Raspberry Pi. two selection of camera is shown in Figure 2.4.

However, the key to using the camera module is figure out how to process a wireless network transmission for video streaming, rather than worry to much about the selection of camera. The planned method is that to build a network webcam server using Motion. Motion is a lightweight application done in Linux to help the user control the camera. There is an HTTP server inside of the Motion, so the user could access the video streaming from remote web browser by "http://[IP address of Raspberry Pi]:8081", 8081 is Motion's default occupied port[30]. In order to use the application, no programming script will be required, but configuring the .configure file and the daemon(in order to ensure that this application is always working in the system background) will be necessary.



(a) PiCam



(b) USB Camera

Figure 2.4: Two popluar Raspberry Pi Camera [44][45]

2.4.4 Motor

DC Motor

The DC(Direct Current) motor and the AC(Alternating Current) motor is most common device that converts electrical energy into mechanical energy and complete a electro-mechanical movement[31]. In order for the appliances to move flexibly, four DC motors are expected to be use in this project. Meanwhile, if people try to control DC motors by the Raspberry Pi, the H-Bridge module will be used, and the most popular solution is L298N chip, which has been shown below in Figure2.5. The L298N motor driver module has two channel H-Bridge circuits, so during its implementation, four DC motors can be divided into two group,and be inserted into both sides of the chip, like the wheels on each sides of the embedded car, on both sides of the "output A" and "output B" . L289N provides two kinds of voltage(+5V,

+12V) and a “GND”. what should be laid emphasized on is that this “GND” not only needs to be linked with the negative pole of the power supply, but also the the “GND” pin of the Raspberry Pi. Otherwise, the Raspberry Pi IO cannot to identify whether the current L298N outout is “high” or “low”, and is unable to form a loop.

The main point of using the L298N is managing the Enable A and Enable B pins, and the 4 four logical current input pins. The Enable A and Enable B pins are used to control the speed of the motors, which are usually placed on jumpers. If removing the jumpers and link to Raspberry Pi, people could adjust the speed of motor by GPIO PWM[32]. In addition, four logic input pinS(IN1, IN2, IN3, IN4) control the rotation direction of the four motors. The Raspberry Pi controls these variables by change the level output.

An interesting thing in regards to Raspberry Pi is that the default power-on voltage of each pin is different. In the same channel, if the received input value of every motor is different, the motor will rotate without any control. As a result, during the period of circuit design, people should guarantee that the used Raspberry Pi GPIO have same default level. Generally speaking, GPIO 0-8 default to high, while GPIO 9-31 default to low[33].

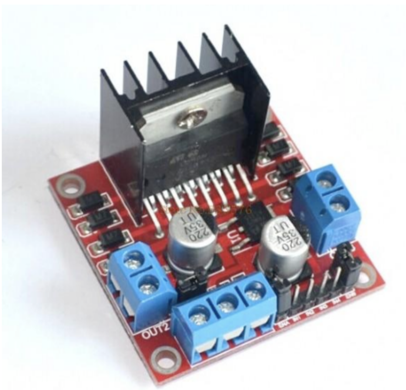


Figure 2.5: The Motor Driver Module L298N [46]

Servo Motor

In addition to the DC motor, another motor that is often used in embedded development is the servo motor, in this project, the servo motor will be use to improve the ability of the ultrasonic sensor obstacle avoidance as well as to expand the surveillance range of the USB camera. In general, if users want to change the angle of the sensor or external circuitry, the servo motor can be applied. The servo motor itself is based on a DC motor, a potentiometer and a control circuit[34]. The existence of these three components indicated the servo motor is able to accept external level feedback.

Compared with a DC motor, it is easier to control the servo motor by micro -controller. Under the premise that the control accuracy and the stability are not required specifically, people could control the servo motor by Pulse Width Modulation (PWM) smoothly, which is supported by RPi.GPIO packet defaultly. In this mode, the value of pulse is directly related to the angle of servo motor, which has been shown reflacted in Figure2.6. In normal circumstances the servo motor can rotate 180 degrees only, thus, before using this motor, we need to pre-adjust the angle of the servo motor gears. Otherwise, effect of using the servo motor will be reduced.

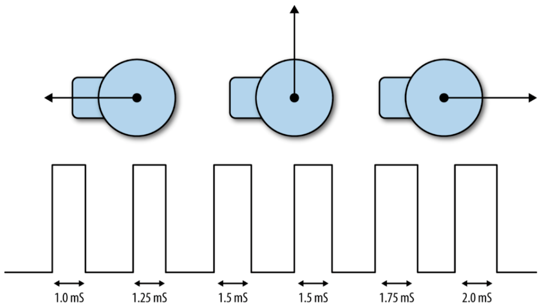


Figure 2.6: Control Rotation Degree of Servo Motor by PWM [47]

2.5 TCP/IP Socket

What is know to all is that the various network protocols has been split into six layers by the OSI model. The function of the transmission layer is as a provider of end-to-end logic communication for the application layer, and this layer is between the network layer and the top layer(the application layer). In this layer, there are two famous communication protocols, which are called TCP and UDP. the UDP does not establish any connection before data transmission, and as a result, the UDP interactive data transmission becomes unreliable(server does not know whether the message has been received by the client). in comparsion, the TCP protocol is more complex. Here the main technology of TCP network programming, which is named socket, will be introduced in detail.

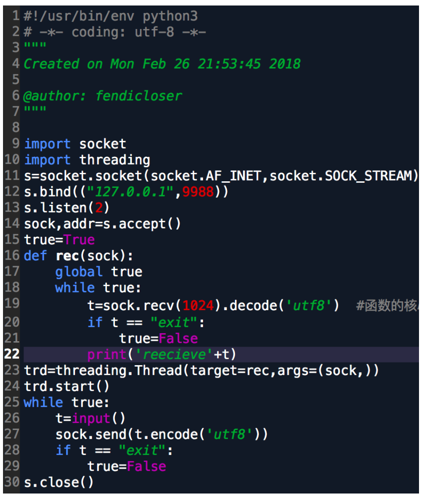
To complete a TCP socket programming, it requires at least two data ends. One end is the server, another is the client. The essence of this system is two programming that can conduct two-way data transmission. The main steps to make it are as follows: Firstly, due to the TCP protocol, two programs must make a connection before data communication happens, which is famous three way handshake, Second, a serverSocket need to be created, which need to knows the IP address of the server machine in advanced(the localhost IP address is 127.0.0.1, the IP address of android studio simulator should be done in the format of 10.0.X.X). Developer also needs to assign a free port for the serverSocket, and the serverSocket should be able to bind and listen to the assigned IP address and port. Thirdly, for the client end, after knowing the server's IP address and port, it can request access to serverSocket, ServerSocket will receives the request through "accept()". At that time, the connection will be established, the remaining problems are related to design information transmission operation, such as the encoding and decoding of the data, the InputStream, the OutputStream as well as multithreading. Finally, if the client end have not had any requirement of data transmission. Releasing related resource by close().

In accordance with what was mentioned above is the method of making a TCP socket programming. Compared with UDP, the process is more complicated, the flow control and the blocking control should be laid emphasis, but the obvious advantage is that the information transmission at ends is reliable. Socket programming is belonged to model of TCP/IP protocol, so these is not any requirement of what kind of programming language is selected. therefore, at two ends of this connection, we can use different programming language, which is also the principle of this project can be implemented. Example, the Figure2.7 below is implemented Python socketServer and socketClient. We can rewrite the socketClient by Android Java and remain the Python socketServer.

For this project, there are two schemes for the connections between the Raspberry Pi and the Android phone. The first one is that the Raspberry Pi is regarded as a "Socket Server" and the Android phone will be designed as a "Socket Client". The premise of this setting is that Android phone and Raspberry Pi could be regarded as a complete IoT system without requirement of adding other electronic components.

However, if users want to control more kinds of devices, such as the switch of household appliances, and the IoT system will be expected to add more functions and as such the second method will be a better choice: Using a third-type device as the "Socket Server" allow more port using for listening additional hardware devices. In this case, both the Raspberry pi and the the Android phone serve as the "Socket Client". For example, a personal computer will serve as the "server", and the android phone will send a "Socket message" to the computer. Then, the computer make a process of the "Socket message" by decoding, compiling and encoding it again. Finally, the Raspberry Pi and IoT devices will be triggered by received "Socket message". Because of one more Socket interaction, the network latency will be more serious, but more interactive and extensible.

Due to the fact that this project is just implementation of a model of a smart home monitor, it is not necessary to consider the implementation of other functions. To reduce network delay, the first connection mode will be choosen.

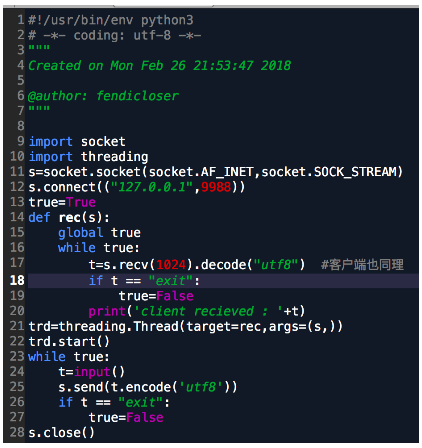


```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 26 21:53:45 2018

@author: fendicloser
"""

import socket
import threading
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind(("127.0.0.1",9988))
s.listen(2)
sock,addr=s.accept()
true=True
def rec(sock):
    global true
    while true:
        t=sock.recv(1024).decode('utf8')  #函数的核
        if t == "exit":
            true=False
        print('reecieve'+t)
trd=threading.Thread(target=rec,args=(sock,))
trd.start()
while true:
    t=input()
    sock.send(t.encode('utf8'))
    if t == "exit":
        true=False
s.close()
```

(a) ServerSocket



```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 26 21:53:47 2018

@author: fendicloser
"""

import socket
import threading
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("127.0.0.1",9988))
true=True
def rec(s):
    global true
    while true:
        t=s.recv(1024).decode("utf8")  #客户端也同理
        if t == "exit":
            true=False
        print('client recieved : '+t)
trd=threading.Thread(target=rec,args=(s,))
trd.start()
while true:
    t=input()
    s.send(t.encode('utf8'))
    if t == "exit":
        true=False
s.close()
```

(b) ClientSocket

Figure 2.7: A ServerSocket and A ClientSocket

Chapter 3

Analysis

3.1 Objectives

The main objective of this project is to provide a kind of smart home monitoring solution, this solution must hold the feature that low-cost and has practical use value. This solution must be easy to be accepted by ordinary users. For software, the implemented system must has good robustness, readability and portability. In terms of hardware, the hardware devices of the system should have a stable supply of power, never power outages. The connection and assembly of the hardware must be firm and reliable. The data interaction between hardware kernel and software, whether wired or wireless, must be reliable. No serious network delay and loss of data.

According to the function, as a smart home monitor, the device should have the same camera or microphone as other monitors. The real-time image captured by the camera should be transmitted to the users handheld device. What is more, this monitoring device should have a wider monitoring area. Therefore , this device should not be limited to a certain fixed position. But should have the ability to move. Further, the device should have a variety of different ways of moving. Such as moving itself, or parsing the users instructions and be remotely controlled. For self-moving, there are many options, such as automatic obstacle avoidance moving, introducing the concept of AI. moving among a preset path or Phototaxis moving. In addition to the above features, the solution should be able to acquire other types of data by sensors. Such as temperature, humidity, air pressure, or concentration of some gases.

Through the above description of requirements, we have made some analysis and assumptions: this smart home monitoring device may be a model of a smart embedded 4WD car that can control the motor to achieve the purpose of moving. This model may be equipped with various sensors, cameras and microphones for receiving external data. What can be affirmed now is that this device must equipped Wi-Fi module, Bluetooth module or network cable that will be used for the network data interaction with other devices. Correspondingly, the users handheld devices that may be a smart phone equipped with an android system or an IOS operating system. On the mobile phone, there is an app installed to access the internal data of the model and send requests to achieve the purpose of remote control. We can follow these assumptions and inferences. A hypothetical listing of the areas and technologies involved in the entire project:

- The Linux system operation;
- Embedded system GPIO port control and circuit design;
- Network programming technique, socket.io, SMTP;
- Multithreading;
- Event listening and processing;
- Exception handling;
- Processing the transmissions of different types of data (image, command, String, etc.);
- Android front-end design and function implementation;
- Using Python RPi.GPIO, C++ or other mainstream hardware programming languages and their main library files to control sensors and motors.

The key of this project is to work out how to link the model of the 4WD monitor car and the Android phone by socket or other network transfer protocol and technology. including multithreading, exception handling and so on. Another difficult problem is related to the various types of data transmission, especially the graphic data, how to transmit the video stream data through the socket to the phone and then playing it through the special model phone application interface is also a serious challenge. Furthermore, network delay caused by video stream data is another hurdle. For wireless network transmission, the network delay is unavoidable[39]. But, under the premise of ensuring the quality of video, how to reducing network delay is an urgent problem to be solved.

The main requirements have been listed in the following tables, including a series of functional requirement and non-functional requirement. In the section of project implementation, all requirements will be implemented strictly following what is necessity.

3.2 Functional Requirement

Table 3.1: Functional Requirement

Requirement ID	Necessity	Requirement Definition
FR1.0	Mandatory	The model must be able to move smoothly, go forward, back, left and right.
FR2.0	Mandatory	The model must be able to move on preset paths and back to the origin
FR3.0	Mandatory	The model must be able to do a obstacle avoidance movement
FR4.0	Mandatory	The model must be able to move along the presetting track
FR5.0	Mandatory	The model must be able to detects toxic and harmful gases: Methane, LPG, smoke. Selecting suitable model of gas senor
FR6.0	Mandatory	The model must be able to reading outside temperature-humidity information
FR7.0	Mandatory	The model must be able to control camera capture video stream data
FR7.1	Mandatory	The model must be able to controls servo motors, enabling the camera module to rotate 180 degrees horizontally and 180 degrees vertically
FR8.0	Mandatory	Implementation of networking programming connection between the model and mobile phone by Python and Java or other two different programming language
FR8.1	Mandatory	Different types data are effectively transmitted through the TCP/IP Socket or other Web Technology
FR9.0	Optional	Integrating some part of the system by the development board
FR10.0	Mandatory	System provide smart phone application interface, which is used to operate the 4WD car model
FR10.1	Mandatory	The smooth switch between each app Activity.
FR10.2	Optional	The back-end programs and front-end layouts, widgets, containers work on all major mobile devices
FR10.3	Mandatory	The video display on the smart phone correctly, which is obtained by camera
FR10.4	Mandatory	App Activities optimization, No blocking, flashback and other exceptions occur during program operation.
FR11.0	Mandatory	The System is able to store important environmental information by Database
FR11.1	Mandatory	User is able to check database data from the Android client app
FR12.0	Mandatory	Have the function of self-protection, prevent to be destroyed arbitrarilyguard against theft
FR12.1	Mandatory	Anti-theft, alert users by text message or email

3.3 Non-functional Requirement

Table 3.2: Non-functional Requirement

Requirement ID	Necessity	Requirement Definition
NFR1.0	Optional	The system will be user-friendly for all levels of technical abilities
NFR2.0	Mandatory	The functional testing of every modulars the 4WD car model, repairing and recording of occupied problem
NFR3.0	Mandatory	The functional testing of the smart phone client, repairing and recording of occupied problem
NFR4.0	Mandatory	The functional testing of the model and the smart phone client network data interaction, repairing and recording of occupied problem.
NFR5.0	Mandatory	The entire system should do its best to comply with the low-cost principle
NFR6.0	Mandatory	The system has reliable power supply to ensure the effect of the system
NFR7.0	Mandatory	When the mobile terminal interacts with the model, the model can respond quickly to the request of the mobile terminal to ensure the efficiency of the system.

Chapter 4

Design

The system consists of an Android phone and a model car controlled by a Raspberry Pi. The Raspberry Pi acts as the server side of Socket.IO, and the Android phone acts as the client side, which is connected by a router. The Raspberry Pi is loaded with sensors, cameras, databases and motors. The Android side has a dedicated app for control. The structure of the system is as following Figure 4.1:

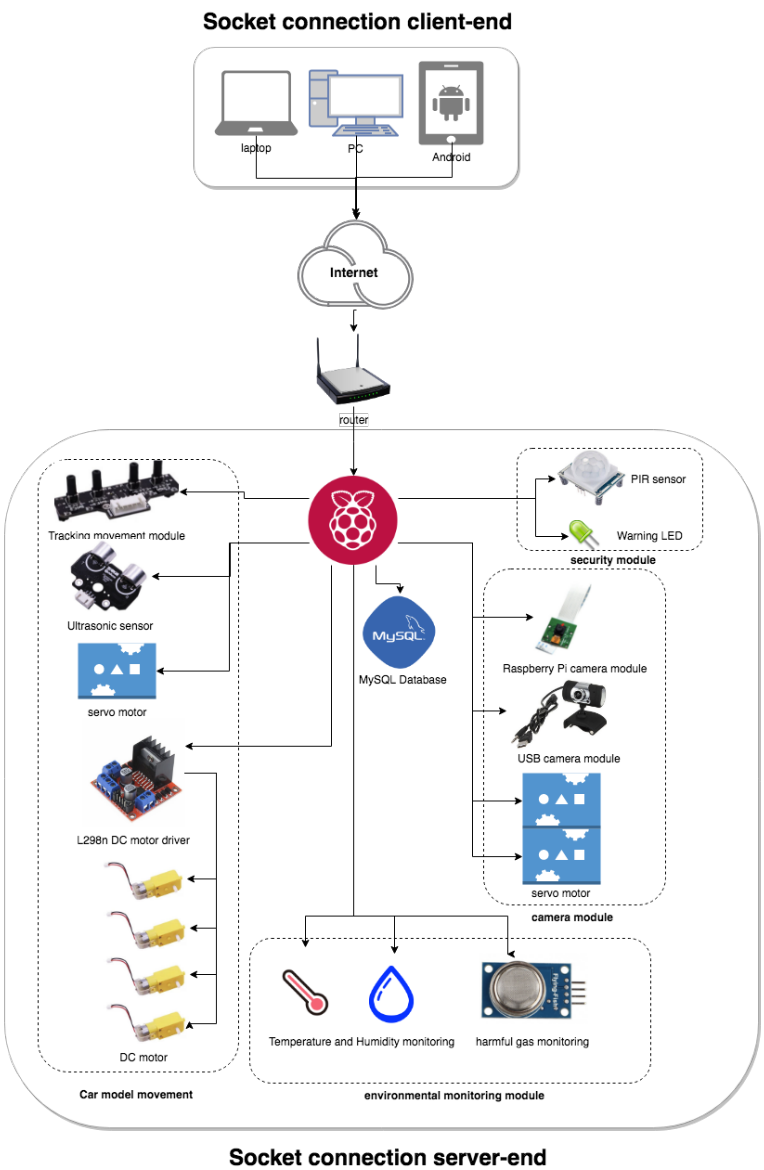


Figure 4.1: System Infrastructure Diagram

4.1 Raspberry Pi Server-end Design

The program development of the Raspberry Pi-end is different from ordinary PC's. Compared with ordinary PCs, the Raspberry Pi is just a piece of the most basic development board. No mouse, keyboard or display screens. It is important to choose the most appropriate hardware device and download the suitable IDE to ensure a comfortable development process. The development period of this project will long, so the safety of the hardware had to be ensured during the daily handling and the processing of assembling of the car model. During the development, the completed content is required to be backed up to a cloud server frequently, because any damage to the hardware device may have caused serious and irreversible consequences. In order to improve the safety of the system, the hardware system will introduce a PIR(passive infrared sensor) sensor and a LED indicator light. The expected effect is that, when living creature is closing to the car model, the living creature can be detected. Meanwhile the indicator light is on, and a warning message will be sent from specified email address to the destination address, this project plans to send warning email to student's QQ email by the Google Gmail's SMTP service.

For material selection, the principle of Low-cost principle should be followed, combling cost and efficiency to find the most optimal material. The weight of the car model is also important to consider, as every component of the car model is required to be as light as possible. Otherwise, it will put a strain on the motor and batteries to move it.

In terms of functions, the car model is required to be capable of four types of movement. The first movement should be done is the human control movement. Because, in the model, the Socket-client end will control the direction of the car's movement which will be regarded as the basis of other movement types. After that, the performance and speed of the car model is required to be tested and recorded immediately.

The problem that need to be considered involves the performance of the tracking sensor, and it includes the sensitivity of the photosensitive resistance (LDR), and the width of the black insulating tape. The distance between sensor probes and the ground, the car model speed as well as the indoor lighting intensity can all affect this problem.

For ultrasonic obstacle avoidance, apart from ultrasonic sensor, in order to widen the ultrasonic wave range and improve the flexibility of the car model, we should introduce a servo motor alongside the already existing ultrasonic sensor.

For human control movement, the user can not only control the rotating direction of wheels, but also control the camera's platform to encourage appropriate steering according to the user's commands. The platform consists of two servo motors, one turns the camera to right and left, another turn the camera to up and down. For the servo that control the rotation up and down, a 180°rotation angle is clearly meaningless(cameras facing the ground or behind the car are meaningless). For the servo that controls the rotation left and right, there are three direction should be considered: turn left(0°), turn right(180°)

and go forward(90°)

Presetting path movement is the easiest movement to be implemented. It should be planned ahead that the car model should be able to return to the point of origin, which is means that the relationship between the speed of the car model and actual moving distance should be emphasized. After entering this movement model, the car model is able to move following the preset path. The sketch diagram has been shown in Figure 4.2 below:

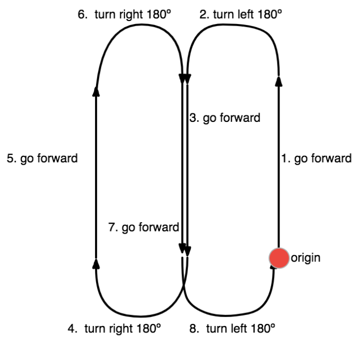


Figure 4.2: Preset path of car model in this project

4.2 Android app Activity Design

For the user interface, in order to follow the principles of user-friendly design, the Android-end of this project requires at least six or seven activities.

The first activity is used for the Socket connection. This activity has three textareas, which are used to input the IP address, the port address of the server as well as the URL of the video. In addition, there is a "connect" button, which is used for build a TCP/IP Socket connection, and turn to the second activity.

The second activity is "Function Selection Activity", due to the fact that the car model is planned to have the ability to run for a long time, the Android application is also required to have the ability to run and receive data over a long time in the back end. This feature means that users do not need to enter or return to the first activity frequently for building the Socket connection. So, there is not "back" button to the first activity in this activity. This activity has five buttons for entering four "movement applications activities" or entering the "database query activity". These five buttons are established vertically, and the button for "database query activity" should be set at the bottom, which is in line with the usage habits of users.

The third activity is the activity of basic movement. There are three similar activity, which display the same behaviours and views for different movements: the tracking movement, the obstacle avoidance movement and the presetting path movement. When the user selects the related movement in "Function Selection Activity" and sends the request, the Raspberry Pi server-end will run the related function, and the car model will start to move. The Android application turn to corresponding activities. At the top of these activities, the title section will display which kinds of movement being done, and the "back" button at the top left corner is used to end the thread. Two textareas below the title section are used for show the current temperature and humidity respectively. There is a videoview or webview at the bottom, which is used for interpreting and playing the URL video stream. The three interfaces can be inherited by this basic activity, but the Socket requests sent are different.

The fourth activity is responsible for the human control movement, which is more complex. Apart from the two textareas, the "back" button and the videoview, the view specific to this page is eight direction buttons. Eight buttons are above the videoview, and will be divided into two groups. The four buttons on the left are used to control the movement direction of the wheel. The four buttons on the right are used to control the rotation direction of the camera holder.

The last one is related to database data query. This activity required 24 textareas, which are combined into a text table. The top three textarea are similar to the headers of columns, setting the text to "date", "temperature" and "humidity".

In general, the "Socket Connection Activity" does not have any restrictions to its horizontal or vertical display. However, the "Function Selection Activity" and every "Movement Activities" can only be displayed on the horizontal screen, and the "Database Query Activity" can only on a vertical screen. In this way, the design can be more strict, which simplifies the user's operation and conform to the principles of user-friendly design.

4.3 Database Design

The system should have the ability to collect data from related environmental monitoring sensors, and save that data in the database. In addition, the android-end app should provide corresponding interface, so that users are able to view the data from the app.

Actual stored data: every 24 hours, the car module saves the current date and the AM2302 sensor data in the database. The type of data should be Int or Varchar. The stored data must be reasonable and consistent with people's understanding of environmental data in the general sense. This is mean that the Python program is expected to has the ability to filter out incorrect data. The basic principles of sensors and the database commands should be mastered.

Actual viewed data: the user sends the request, the database is able to retrieve the latest seven values of temperature and humidity, that is the recorded date, temperature and humidity of the last seven days.

According to the concept in the previous design, we need a relatively popular database that can be controlled by the Raspberry Pi and Python3. A MySQL database can be easily controlled by Python using some special external libraries. As well, the basic operations of MySQL(insert, select, update, delete) are simple to be implemented.

The function of data storage should be a fixed in the specific threading, which is mean that no matter how the car module running, the external data collected by the sensor should be stored in a fixed information format in the database regularly. For the android-app, if the user requests to query the database data, no matter which movement the car module is currently in, the process should be stopped immediately and the device should stay in place. When the user leaves the data query activity, the movement of the car can be re-selected.

Chapter 5

Implementation

5.1 Launching the script on boot automatically

If the Raspberry Pi that acts as the server-end of the socket communication in this system is expected to be started automatically, there are three conditions that must be met. First, the Raspberry Pi and other related components must be powered normally. Second, the Python program in the Raspberry Pi can be started normally at a boot time. Third, the android client is expected to be able to send request normally, making sure that the IP address of the Raspberry Pi has not been modified. The port to be allocated is idle.

The system requires two independent power supplies. The first one is the power supply for the L298n DC motor driver module. The selected power is four AA rechargeable batteries(circuit of the battery and the L298n will be described in detail in the following section). After installing the four-cell dry battery into the battery box, the power-on indicator light on the L298n will be on. This indicates that the power supply is normal. There is also the power bank for the Raspberry Pi and other hardware. In fact, the electricity demand of the whole system is borne by this power bank, as ordinary dry batteries cannot meet the system's demands for voltage. In this case, I choose a the 24000mA charger. When the charger is plugged into the supply port on the Raspberry Pi, the indicator light on the board will be on. By rough calculations, such a 24000mA power bank and four AA rechargeable batteries has ability to keep the Raspberry Pi and other hardware running steadily for about a week.

Another important thing is to set the Raspberry Pi's boot self-starting service. There are three services that the Raspberry Pi needs to boot. The first is "pigpiod", which will be used to monitor the value of the temperature and the humidity. The second is the main python program on the server side. The third has two options, two kinds of camera, the Raspberry Pi camera module and the USB camera. The method to start these three services is: modify the /etc/re.local file. And use the following command on the terminal:

$$Vim \quad /etc/relocal \quad \#opening\, the\, configuration\, file\, with\, compilation\, mode \tag{5.1}$$

After entering the file, add following command at the bottom of the file:

$$sudo \quad pigpiod \quad \#launches\ the\ pigpio\ library\ as\ a\ daemon \tag{5.2}$$

$$sudo \quad /Desktop/new/Piserver.py \quad \#run\ the\ main\ python\ server\ program \tag{5.3}$$

$$sudo \quad motion \tag{5.4}$$

or

$$raspivid\ \text{-}\ o\ \text{--}\ t\ 0\text{-}w\ 640\ \text{-}h\ 360\ \text{-}fps\ 25 \tag{5.5}$$

$$cvlc\ stream:///dev/stdin\ \text{--}sout\ `\#standard\{access = http, mux = ts, dst =: 8090\}'\ : demux = h264 \tag{5.6}$$

Here, command 5.5 and 5.6 should be executed simultaneously, the actual meaning of the command will be explained in detail in the following section.Finally save the modification and exit.

It is worth to noting that the IP address of the Raspberry Pi in the project is not fixed but that is dynamically allocated. Therefore, when the Raspberry Pi is started the first time. The IP address of this microcomputer must be confirmed by input “ifconfig” in terminal. Then, modify the IP address in the code segment of initialization funciton.

5.2 The Raspberry Pi Physical Layer Implementation

The most important hardware material used in this project is the microcomputer. This project selected the Raspberry Pi 3 model B as the mainframe, which is the latest version of the Raspberry Pi series. In this project, the important factors in this decision are as follows:

- Enough GPIO pins, capable of carrying more pairs of external hardware devices;
- The Raspbian system, which carried into the Raspberry Pi, regards Python as the main development language;
- Compared to earlier versions, the Pi3 model B has a built-in Wi-Fi module, so it does not require an external Wi-Fi module, which simplifies development.

For the implementation of the physics-layer, a development board is planned to be used. However, in the beginning of development, it was unexpectedly discovered that connecting the Raspberry Pi with the development board leads to the Raspberry Pi can not be started normally. This problem could not be solved in a short time, therefore, the use of the integrated circuit development board mentioned above was abandoned. Meanwhile, the circuit of system was redesigned independently by jumpers and breadboards. The positive side of this problem is that the wiring will not limited by the fixed development board port any longer, which may have been defined already. And the models and types of sensors are not be limited any longer. The negative influence is that a large number of jumper wires are needed. The cables are bound to be very messy and increase the risk of hardware damage, such as short circuits. The consequences of such risks are often irreversible.

This project uses a large number of hardware sensors, DC motors, GPIO port, camera, jump wires, power and related development equipment, such as the display, keyboard and so on. What kind of equipment to purchase and how to connect the equipment to the Raspberry Pin in a reasonable way is a significant issue. Based on the principle of low-cost, a variety of sensors and development boards have been repeatedly attempted. There are 19 GPIO used in the whole project, which has been listed in Table 5.1. Apart from pins that meitoned in Table 5.1, the camera module port and a USB port of Raspberry Pi is required to be used. All of the pins of the Raspberry Pi are controlled by “BCM” work mode, which is the serial number of the GPIO, and not the number of the pin on the Raspberry Pi. Note: This table does not list the VCC pin and the GND pin of some components, because these omitted pins must be connected with the VCC and the GND of the Raspberry Pi. Sensor modules of this project do not have serious demand for current, so ther is no need to worry about the problem that the shared current is lower than minimum current.

Table 5.1: Hardware usage of the project

Hardware	Interface	TO
L298n DC motor driver module	IN1	Raspberry Pi 3 GPIO 20
	IN2	Raspberry Pi 3 GPIO 21
	IN3	Raspberry Pi 3 GPIO 19
	IN4	Raspberry Pi 3 GPIO 26
	ENA	Null
	ENB	Null
	+12V	4 batteries Vcc
	GND	4 batteries GND & Raspberry Pi 3 GND
Front-left DC motor	Vcc	L298n OUT2
	GND	L298n OUT1
Back-left Dc motor	Vcc	L298n OUT2
	GND	L298n OUT1
Front-right DC motor	Vcc	L298n OUT3
	GND	L298n OUT4
Back-right DC motor	Vcc	L298n OUT3
	GND	L298n OUT4
Ultrasonic-control servo motor	OUT	Raspberry Pi 3 GPIO 23
Camera up down servo motor	OUT	Raspberry Pi 3 GPIO 9
Camera left right servo motor	OUT	Raspberry Pi 3 GPIO 11
AM2302	IN	Raspberry Pi 3 GPIO 2
MQ-2	DO	Raspberry Pi 3 GPIO 8
	IO	Null
HC-SR04 Ultrasonic sensor	ECHO	ID_SD
	TRIG	ID_SC
Track Sensor	X4	Raspberry Pi 3 GPIO 18
	X3	Raspberry Pi 3 GPIO 4
	X2	Raspberry Pi 3 GPIO 5
	X1	Raspberry Pi 3 GPIO 3
HC-SR501 PIR	SIG	Raspberry Pi 3 GPIO 14
LED indicator	IN	Raspberry Pi 3 GPIO 15

5.3 Implementation of L298n drive four DC motor

The most obvious difference between the car model developed in this project and other models is that there are many movement possibilities. If DC motors are planned be to controlled by the Raspberry Pi GPIO pin, the system must be equipped with relevant motor drivers, because the ordinary DC motor only has two pin for Vcc and GND, and even no level I/O port.

L298n has four input pin(IN1 IN4), which are connected to four Raspberry Pi GPIO pins. Meanwhile, four output pins are connected to four DC motors respectively. The wiring diagram has been shown in Figure 5.1. For wiring, there are still two issue that should be emphasized:

- L298n use +12V voltage, so it need external power supply. At the same time, L298n GND do not only need to connect with the external power GND, in order to form a complete circuit, but also needs to be connected with the GND of the Raspberry Pi.
- Apart from the four input pins mentioned above, L298n also has two enabled ports, ENA and ENB. If there is special requirement of the speed of wheel(PWM motor frequency, the moving speed of car model), then the ENA and the ENB can be applied.

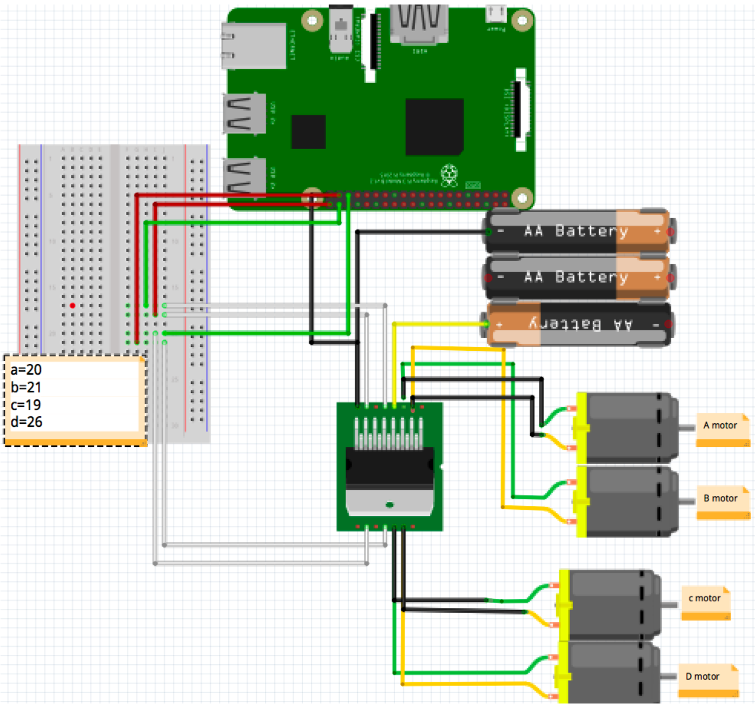


Figure 5.1: Circuit Diagram of L298n Motor Driver

For the Raspberry Pi end, setting the work mode of four GPIO pin by "GPIO.setup(pin,GPIO.OUT)". When the level of pin is HIGH, the related wheel starts turning. In this project, the car model has seven kinds of movement methods, which has been show in Table 5.2 below.

Table 5.2: Movement Direction of L298n

Level of left-front wheel	Level of right-front wheel	Level of left-rear wheel	Level of right-rear wheel	movement
HIGH	HIGH	LOW	LOW	Go forward
LOW	LOW	HIGH	HIGH	Back
LOW	LOW	LOW	LOW	Stop
LOW	HIGH	LOW	LOW	Left
HIGH	LOW	LOW	LOW	Right
LOW	HIGH	HIGH	LOW	Turn left in place
HIGH	LOW	LOW	HIGH	Turn right in plac

As mentioned above, the Enable pins of the car are limited, so the speed is fixed. If the car tries to turn in a certain direction, the python program just need to call the function and use a delay function to keep it moving. For example: the function to make the car model go forward for 5 second has been shown in code segment blow(Figure 5.2).

```
def forward():
    GPIO.output(a,GPIO.HIGH)
    GPIO.output(b,GPIO.LOW)
    GPIO.output(c,GPIO.HIGH)
    GPIO.output(d,GPIO.LOW)
    time.sleep(5)
```

Figure 5.2: The instance of L298n drive the DC motor simple go forward

5.4 Four kinds of movement

5.4.1 Tracking movement

Tracking movement is done applying an integrated infrared sensor. The sensor of the four infrared sensing heads (LDR) are position vertical towards the ground, and make sure the four sensing head are perpendicular to the car module. This will allow them make to sense whether there is a change in color on the ground. According to the distance of each LDR, putting black insulating tape on the floor(the width of tape should in the range of 16mm to19mm), so that the car module can move forward along the black track after power supply. The next step is to number the four LDRs and define a bool type value for each pin(Figure 5.3).

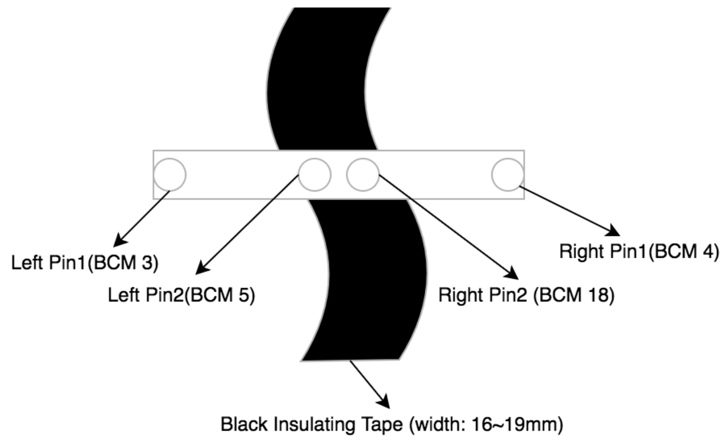


Figure 5.3: Schematic diagram of four pins of Tracking movement

The main function of the tracking movement is implemented by estimating “whether the four sensor heads sense the black tape”. Before developing this process, some rules must be remembered:

- When the voltage of pin is high, the boolean value is “False”, which prove that the LDR sensed the tape, or that the car has left the ground, and the distance between the obstacle and the LDR exceeded the induction range. At that movement, the blue indicator light of the LDR will be on.

- If the black tap is not detected, the boolean value will be "True" and the light will not be on, when that happens, the voltage of pin is low. The user can turn the knob on the sensor to adjust the sensitivity of the four sensors.

For developing: the occurrence of acute angles in the pathway is not considered, which is rare and not usable. A judgement needs to be defined in the Python function:

Table 5.3: Judgment need to be detected by Tracking movement

The stutiation pathway and car module	Left-Pin1	Left-Pin2	Right-Pin2	Right-pin1	The movement of car module
Straight line	True	False(light)	False	True	Go forward
The car faces an obtuse Angle or right Angle to the right	False	False	False	True	Turn right
The car faces an obtuse Angle or right Angle to the left	True	False	False	False	Turn left
The car deviated to the right of the pathway	True	True	False	True	Turn left in place
The car deviated to the left of the pathway	True	False	True	True	Turn right in place
The car has veered seriously to the left	False	True	True	True	Turn Right
The car has veered seriously to the right	True	True	True	False	Turn Left
The car was completely off pathway	True	True	True	True	Exception, stop
The car is completely perpendicular to the pathway or off the ground	False	False	False	False	Exception, stop

5.4.2 Obstacle avoidance movement

Obstacle avoidance movement uses a servo motor and a HC-SR04 Ultrasonic sensor. The working modes of the Ultrasonic trigger port and receiving port of the HC-SR04 sensor need to be set respectively, which means that the "TRIG" is the output and the "ECHO" is the input. Defining a distance-testing function in program, the Raspberry Pi control the servo motor by PWM model for changing the direction of the motor, The relationship between the working voltage and pulse width(and the relationship between pulse width and rotation degree) need to be identified through practical tests. The conclusion is:

- ChangeDutyCycle(7): turns to 90°.
- ChangeDutyCycle(4.5) turn to roughly 45°on the right.
- ChangeDutyCycle(13): turn to roughly 180°on the left.

The reason why the right turn reaches 45°rather than 0°is that the pillar of the camera haeundae board is located in the right side of the HC-SR04 Ultrasonic sensor. If using ChangeDutyCycle(1) turn the Ultrasonic sensor to 0°, the pillar will make negative influence on the accuracy of the distance monitoring result. This is regarded as a minor problem.

There is another thing also should be noted: in this project, both the servo and the Ultrasonic sensor are controlled by the GPIO of the Raspberry Pi. The accuracy of the distance testing result can be interfered with by the servo motor if the voltage of the servo motor is not released by "ChangeDutyCycle(0)". The actual wiring diagram of obstacle avoidance movement is shown in Figure 5.4 belowThe problem to be noted is that according to the measurement principle of the Ultrasonic sensor, the actual Range is (5cm, 600cm) [50]. Therefore, if the distance between the obstacle and the ultrasonic probe of the sensor is less than 5cm, the sensor is likely to fail.

The detailed action of obstacle avoidance movement has multiple step. First, the servo motor turns forward, using ChangeDutyCycle(7) to turn the servo motor to 90°. A 0.5 second delay is used to ensure that the servo motor has enough time to turn to the specified position and release the voltage. Calling the distance testing function then completes the first distance measurement. If the distance value obtained is greater than 100cm, the car model goes forward for 3 seconds. If the distance value is between 40cm to 100cm, the car model moves forward for 2 second. If the distance value is less than 40cm, the obstacle in front is very close to the car model, so the servo motor turns to left and right and completes the distance measurement for each side respectively. If the distance value obtained on both sides is less than 40cm, is means that there is no suitable route, so the car model should turn left in place for 6 seconds to complete a U-turn. Else, comparing the results of the last two distance measurements should lead to the car turning to the side with a largest distance value.

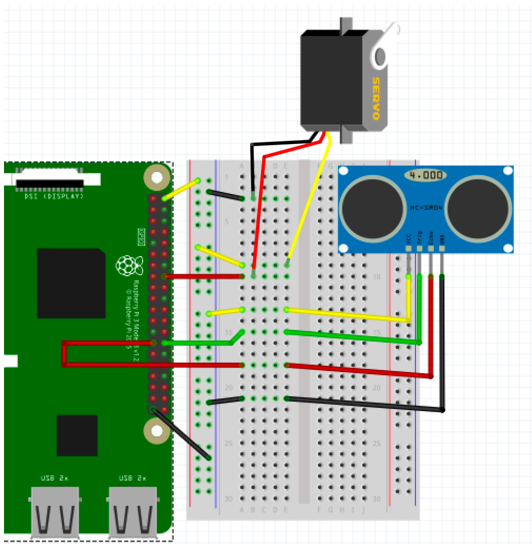


Figure 5.4: Circuit Diagram of HC-SR04 Ultrasonic sensor

5.4.3 Human control movement

For this mode, the implementation of the car model's movement is relatively simple. The DC motor makes a corresponding turn according to the command obtained by the Socket decoding. However, in this mode, the camera platform also need to be control by users. In order to make the car turn left and right horizontally more smoothly, the simple ChangeDutyCycle(frequency) should be placed by setting the frequency dynamically with a "for" loop. We already know that when the servo motor turn to 0°it means that the servo motor turn to left. So:

```
def servo_left():
    for i in range(180,0,-10):
        pwm_LeftRightServo.ChangeDutyCycle(2.5+10*i/180)
        time.sleep(0.04)
```

Figure 5.5: Code segment of smoothly steering servo motor

The reason why the time interval is 0.04 is that, generally speaking, the required for the servo motor to rotate 60°is 0.04 seconds, and this time interval is alway regarded as the unit speed of the servo motor. If you want the servo motor to rotate in a more gentle way, this time interval can be increased. Another thing worth mentioning is that, after above compilation is finished, despite of the fact that the servo motor can complete the rotation request, there will be seriou jitters. Because, although the servo motor has rotate to the specified degrees, the servo motor is still in the state of being powered, so the rotation of the gear and the frequency of pulse is inconsistent. The internal gear is easily damaged. So, after complete rotation, the pulse needs to be released using ChangeDutyCycle(0) every time.

5.4.4 Presetting path movement

Implementing other kinds of movement, and obtaining the moving speed of the DC motors is the fundamental step of the module. After the actual test, the obtained speed value are: the car model takes roughly 4.1 seconds to move 100cm forward, and roughly 7 seconds for rotate 180 to right or left. After obtaining the above values, the moving plan for the car model in the design chapter can be implemented.

In summary of the four movement implementations, the most difficult movement to be implemented is the tracking movement. Hypothesis and verification of the various monitored results is required; adjusting the sensitivity of the LDR according to the environment and the the relationship among the Voltage of input pin, boolean value and indicator light status also should be considered frequently. The most cumbersome of these is the humidity movement, not because of the difficulty but because it involves more Socket instructions. Among the four movements, the tracking movement has strict requirement on external environment conditions(cannot be used under the stutiation of poor visibility or poor lighting). If necessary, the developer can turn on the two Enable Pins on the L298n, after defining the pin and setting their work mode(PWM), the speed of car model can be controlled.

5.5 Anti-theft Implementation

In order to achieve the anti-theft function, the first thing is determine the code segment of this function should be placed in which thread. Actually, the car model should always be in the state of running this function, which is means that this function has same priority as the harmful gas monitoring. In order to achieve the anti-theft requirements, there are two parts must be implemented. The first is the sense of living creature, in detailed, implement the function of PIR and LED indicator, meanwhile, send simple plain text email by Python programming language.

This first part is relatively simple, after connect the GPIO of the Raspberry Pi to the PIR and LED correctly, setting the work mode of PIR to input. In the fact, this sensor can be regarded as a trigger or switch. The second part is to send email, despite of the fact that it is quite simple to control SMTP by Python, which just need to following fixed rules and to operate the smtplib library. However, there is a potential problem, Google Gmail has strict requirement on the security of the application, Applications of the Raspberry Pi just has a low security credit level, so the System server side is not be granted to send emails. To solve this problem, login to Google account on Raspberry Pi's browser to complete the security registration, Google will treat this car model as security devices. Then, the most important point is that student is required to complete Google's 2 step verification. Because, only after completing its, student is able to apply for an application specific password for the Python Socket server program. The filename of this project's main program is "testmysqldb.py", apply a specific password for its. This password is used for login Google's SMTP server, which has been shown in Figure 5.7 below. In this figure, the variable toAdd is the destination email, this variable can be set to a String array, which is means that an email address can broadcast mail to multiple destination email address once.

The code segment in Figure 5.6 should be written in the initialization section, the code segment in Figure 5.7 should be written in warningfucntion(). The most important function is .sendmail(), the first parameter is email sender, second one is receiver, and the third one need to be need that this parameter sets both the subject and body of the mail and splits them by '\n\n'. If there is only '\n', the mail can still be sent normally, but the body part is empty. Since the two camera in this system will be used for other functions, the real-time photo shooting or image email tranmission will not be implemented for the anti-theft fucntion.

```
if GPIO.input(tpir):
    print("pir detected")
    GPIO.output(led,GPIO.HIGH)

    s=smtplib.SMTP('smtp.gmail.com',587)
    s.ehlo()
    s.starttls()
    s.ehlo()
    s.login(user,password)
    s.sendmail(fromAdd,toAdd,header+'\n\n'+body)
    s.quit()
```

Figure 5.6: code segment of send mail after the PIR is be triggered

```
user='fendicloser@gmail.com'
password='vjtzisqrywdovqlh'
subject='There is a living creature approaching your device!!!'
toAdd='1033285894@qq.com'
fromAdd=user
header='To: '+toAdd+'\n'+'From: '+ fromAdd+'\n'+'subject: '+subject
body='if it is not your families or your pet, please pay attention to the security of your monitoring device, someone may be vandalizing, moving, or stealing its.'
```

Figure 5.7: code segment of initializes the relevant parameters

5.6 The Environmental Monitoring Implementation

The implementation of various movements expands the monitoring range of the car model, which also enables the car to be used normally in different external environments. As a smart home monitoring device, the most important function is external environment monitoring.

5.6.1 Temperature and humidity monitoring

Actually, The capture of the value of temperature and humidity is implementation the control of AM2302 sensor. After complete physical hardware connection between the AM2302 and the Raspberry Pi, we have two kind of methods to control the AM2302 temperature and humidity monitoring sensor. The first method is to group the temperature value, the humidity value and the check bit separately according to the returned 40-bit electrical signal(40-bit binary data). The temperature value and the humidity value are extracted and grouped respectively, and the Raspberry Pi is able to print the readable data in the console. This method involves converting between binary and decimal numbers, as well as simple array operations, which are relatively complex. The second method is to invoke an external python module. DHT22(AM2302) is a must-try sensor for beginners to simplify operations. Many sensor companies have launched their own AM2302 module driving library.

Student chose the second option. The applied Python library is named “DHT22”, which was published by Adafruit. In addition, student introduce a more advanced way of controlling the GPIO of the Raspberry Pi, called “pigpiod”. This control method can be interpreted as a daemon that need sudo permission to boot, which is also the reason why the startup program for the Raspberry Pi needs to run “sudo pigiod”. The main code segment for implementing this function is:

```
pi=pigpio.pi()
dht22=DHT22.sensor(pi,THPin)
def getTemperature(sock):
    dht22.trigger()
    time.sleep(0.5)
    y1=dht22.humidity()
    x1=dht22.temperature()
    return (x1,y1)
```

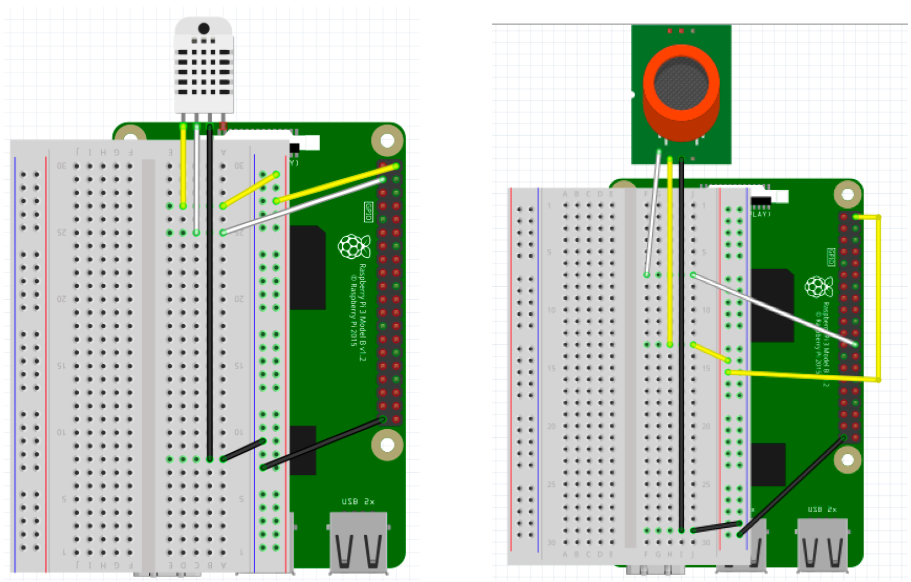
Figure 5.8: Code segment of getting temperature and humidity

There is a problem with this set-up, however The newly launched AM2302 sensor is likely to get the wrong value of temperature and humidity and report back saying "-999". The main reason is "DHT11 tries to pull the data line low as soon as it went up after the 18msec LOW phase. And this issue likely came from the unclear wording in the DHT11 datasheeT"[51]. But acquiring normal value can be detected after the second trigger.

5.6.2 Harmful Gas Monitoring

Both analogue level and digital level can be outputted by the MQ-2. If this project regarded the MQ-2 as a harmful gas concentration measurement instrument, and if the concentration of harmful gases is obtained accurately according to the output level, the analogue pin should be connected with the GPIO of the Raspberry Pi. However, if the project simply treats it as an alarm device, in other words, if any of the four gases are detected (Methane, Butane, LPG, smoke), the internal program can be triggered. In that case, we should connect the digital pin of MQ-2 with the Raspberry Pi. The main implementation principle is that the Python program monitors the level of the MQ-2 output. If the output level is high, it indicates that no harmful gas has been detected, setting a signal. If the output level is low, it indicates that the harmful gas has been detected and thus setting a detected signal. Finally sending this signal bit to the Android-end, it can be recognized and processed accordingly.

The wiring diagram of the AM2302 and the MQ-2 have been shown as following Figure 5.9:



(a) AM2302 Sensor (b) MQ-2 Sensor

Figure 5.9: Wiring diagram environmental monitoring sensors

5.7 Camera Implementation

To use the camera, there are two different implementations. What the two methods have in common is that when trying to convert the video captured by the cameras to video streaming, the Android-end is able to access the URL of the video stream, and display the video in some kind of activity view.

The first one uses the USB camera and the Motion software. Connecting the USB camera with the Raspberry Pi, check whether the camera has been detected in Linux terminal. After downloading the Motion software in terminal, some parameters in the configuration file of the Motion need to be modified:

- In /etc/default/motion, make sure the value of "start_motion_daemon" is "yes". To guarantee that the Motion can run in back-end of the Raspberry Pi as a daemon.
- The default port of the Motion is 8081, which does not need to be modified if there are no other special requirements. However, you must ensure that this port is not occupied
- To enable the video to be displayed on another computer, change the value of "'stream_localhost' from "on" to "off".
- The default value of "stream_maxrate" should be "1", increase its to "50" or "100" for easing traffic restrictions and network latency.

If this method is applied, try to interpret and display the video stream by a webview on the Android-end app.

The second method is the Raspberry Pi official camera module. The Pi camera is inserted into special port of the Raspberry Pi. in terminal, input "sudo raspi-config" to enter the configuration file of the Raspberry Pi. Enable the microcomputer to have permission to operating the camera, save the change and reboot. It is very important that the hardware of Pi camera is very fragile. When making a physical connection, make sure to avoid any damage to this hardware, such as static electricity and wet hand. It is necessary to equip this expensive element with a protective case.

This method use "raspivid" and "cvlc" two Linux command, "raspivid" let camera shoot video, and "cvlc" is used to send the video to VLC player, and transcoding the video stream into h264 network video stream. The actual terminal command is command 5.3.

For "raspivid", "-t 0" means to set the shooting time to infinity, and "-w" and "-h" sets the width and height of the video. "-fps 25" means 25 frames per second. For another, for "cvlc", The network protocols that the client can choose include: HTTP, RTP, RTSP, and in this project, the selected protocol is HTTP. Video captured by the raspberry PI camera module is defined as the standard input stream (stream:// dev/stdin). "access" defines the output protocol as HTTP, and "mux" defines the output multiplexer as ts multiplexer. The output port is 8090. For this method, the android client app can play the VLC video with videoview by interpreting the video URL.

5.8 Database Implementation

Due to the fact that the developing language of the Raspberry Pi server-end is Python3, the "pymysql" library is applied for the operation of MySQL. This external library is specially provide for Python3. If the developing language is Python2, this library can be taken place by "MySQLdb" library. The first thing that needs to be done is to make sure that there is a working MySQL database in the Raspberry Pi, and if not, the command below can be used to install the database:

$$Sudo\ apt - get\ install\ mysql - client \tag{5.7}$$

Meanwhile, the username and a password(username: root, password: aptx4869) needs to be set, in this project, the name of database is "sensordb" and the name of datatable is "TemAndHumTest", which need to be created in MySQL command line. No matter what kinds of operation will be in MySQL, The following code segment needs to be executed to establish the connection and set up the cursor.

```
db=pymysql.connect("localhost","root","aptx4869","sensordb")
cursor=db.cursor()
```

Figure 5.10: MySQL establish the connection and set up the cursor

First of all, the "insertion" operation, before the "insert", making a judgment. If the data table is empty table. it proves that this is the first time the car model is used, and inserting data immediately. There are three data set in the data table: current time, current temperature and humidity. To facilitate subsequent operations, all data types sets in Varchar. If the judgment result is "the data table is not empty", insert data at a specified time interval normally. So how to implement "inserting data at a specified time interval"? The normal delay function should be avoided, because this part of the database operation is written in the main thread. If using the simple "time.sleep()" function. It will effect on other process, and the efficiency of this method is extremely poor.

The method used is to execute the command line "select * from TemAndHumTest limit1 order by date desc;" at a intervals of one second, and find the existing data in the table according to "reverse ordering of date", and extract the date of the latest row. Actually, the data in this MySQL data table can be regarded as a two-dimensional array, and the data of the latest row is result[0][0], regarding its as "oldDate". Meanwhile, the program calls "time.strftime('%Y% M%D%H%M%S', time.localtime(time.time()))" to acquire the current time "newDate". Convert there two data into Int(). If the result of newDateInt-oldD ateInt is greater than 1000000, it is means that no new data has been stored in the database for more than 24 hours. Call dht22() to get the current temperature and humidity immediately and execute an "insert" operation. Overall, this method may seem inefficient, but security and does not have any negative influence on other functions of the program.

Another operation is "select", which means how to get temperature and humidity of the last seven days, as well as send selected data to the Android client-end. The important thing to figure out how to encapsulate the "select" data. The query results obtained by python can be treated as a two-dimensional array, and we need to convert it to a string type. Adding a separator between each two elements is important to avoid getting the wrong temperature and humidity, and to remove the possible wrong values.

5.9 Threads and protocols

The biggest difficulty in this project is the design and implementation of the thread, which is how to implement each task, and how to define the timing and sequence of the use and release of resources through multithreading. Independent processes cannot be influence by each other. Further, each function and hardware module is able to be reused.

First, obtain current IP address and distribute the Socket port. Second thing is the implementation of socket communication, including setting TCP/IP protocol, binding IP address and port address and listening to the connection request of the client-end(The maximum number of connection is five, and this number can be reduced to 1 or 2 for security reasons).

Then, create a thread, to use the project implement multithreaded application via the "threading" library. The running model is setDaemon(). In this way, if the parent thread in the process is regarded as a daemon thread, and this thread ends, its child thread will be forced to end. The advantage of this design is easier to control. Note that an error that the program has been logged out of a working mode, but a hardware or function is still running inside of the child thread. In addition, two other issues should be addressed:

- The "arg" parameter of a thread, the function that the thread executes must contain a Socket command.
- Both the hardware and threads of the car model have a delay, so, In order to avoid the error where some function still send command after the Socket connection is closed directly, closing Socket connection is delayed.

If the initialization of the Socket and the thread is completed, and the thread lock is set, the program will enter a infinite loop. In this loop, the car model connects with the Socket client and starts to receive and process requests. The main description of the project's thread is: The main thread is used to control MQ-2 and store the temperature and humidity in the database on time. This thread has the highest priority, and any functionality related to the whole program can be written in this thread respectively. Further, four movement methods and the "select" operation of MySQL database have their own threads. When the client-end sends a request (actually a string object ending by '#‘"), the program parses the request under a special protocol. Furthermore, the corresponding function will be called.

The detail of this Socket communication protocol is shown in table 5.4 as follows:

Table 5.4: Socket communication protocol

Command (without "#")	function	Command (without "#")	function
1	Tracking movementGet temperature & humidity	end1	Stop tracking movementStop to get temperature & humidity
2	Obstacle avoidance movement Get temperature & humidity	end2	Stop Obstacle avoidance movement Stop to temperature & humidity
3	Get temperature & humidity	end3	Stop to get temperature & humidity
4	Presetting movementGet temperature & humidity	end4	Stop presetting movement, Stop to temperature & humidity
5	MySQL "select" operation	end5	Stop MySQL "select" operation
movingForward	Car model go forward	movingBack	Car model go back
movingLeft	Car model turn left	movingRight	Car model turn right
camUp	Camera go forward	camDown	Camera up
camLeft	Camera turn left	camRight	Camera turn right
Null	Car model stop		

5.10 Android client-end Implementation

The development of the Android-end user interface follows the idea that has been mentioned in the "design" chapter. There are not too many serious exceptions in the development process. Therefore, a detailed description of the problems encountered in the implementation of the back-end functionality is as follows.

Due to the fact that the android application acts as the client-end of Socket connection, the most important feature is Socket connection, which is the first thing that should be implemented. The operation of the Socket connection includes request, binding, receiving data, send data as well as interpreting data. The next step is the implementation of each function. During the implementation, the program should be able to print the received and sent data in the console so that developers can debug or check if the data is correct. Functionally, after establishing the Socket connection and turn to "Movement Selection Activity", the Raspberry Pi Python program should have the ability to monitor harmful gases and to store the data immediately. However, the execution of query for the current temperature and humidity should wait until the "changing movement" request is sent and the related activities are entered. The Socket connections can use observer mode, so that app can keeps a steady connection with the Raspberry Pi server-end. It is not affected by the activity transition. The received data will be decoded by internal function and published to correct position. Considering the properties of MySQL and the principle of Socket connection, it is necessary to process the data transmission and encapsulation in the "Data Query Activity". The type of data should be String in this project's Socket connection. Due to the fact that program of the server-end has inserted serval special element(“#”) in this string. The Android-end program should extract each element with delimiter “#”. That converts it to an array of string types. Finally, each elements of the array is posted to correct positions(textarea).

This android program needs to activate following permissions in AndroidManifest: android calls the network connection(android.permission.INTERNET) and vibrates (android.permission.VIBRATE). The implement process of the whole app started from drawing individual .xml files, and there are 6 .xml file in the project. The actual rendering of the two special .xml files has been shown in Figure 5.13.

Table 5.5: Drawing .xml files

File name of .xml	Description
activity_main.xml	Socket connection, main activity of app
activity_msql_data.xml	display MySQL data, consist of 24 textarea, note: the height of textarea used for show the record date is not be fixed
home_activity.xml	function selection activityincludes 5 buttons, the five buttons are of the same size and fill the whole Home_activity view
humanity _move_activity.xml	single activity made for human control movement, compared with activity_track_movement, there are 8 special direction button.
warning _view_activity.xml	activate when directed the MQ-2. due to the fact that the Raspberry Pi is not able to identify which kind of harmful gas is directed, just fix the text to detected temporarily
activity _track_movement.xml	The following activity will change the title of various movement and the sent Socket command. The main style is that, top content is title, display the current mobile mode of Raspberry Pi, the left side of the title is a back button, two textareas below show the value of the temperature and humidity. The rest of the screen is covered by webview

After drawing the XML file, In the mainActivity the “connect” button is used for capture the String type IP address and the port of the Raspberry Pi server-end by onClick() function. /service/ClientService encapsulates both values and passes parameters. The next step is to build the Socket connection by adjust the Socket library of Java. Actually, it is similar that building Socket connection in different programming language, which has been shown in Figure 5.11:

```java
@Override
public void run() {
    try {
        System.out.println("start connect the server " + host + ":" + port);
        socket = new Socket();
        socket.connect(new InetSocketAddress(host, port), timeout: 3000);
        inputStream = socket.getInputStream();//get socket input
        outputStream = new PrintWriter(socket.getOutputStream());//generate socket output
        SocketEventObserver.getInstance().notifyConnectObserverSuccess();
        while (isRunning && socket.isConnected()) {
            Thread.sleep( millis: 100);
            byte[] data = new byte[1024]; //set the max length of Socket information
            int length = inputStream.read(data, off: 0, data.length);
            String s = new String(data, offset: 0, length, charsetName: "utf-8");
            System.out.println("the receive is " + s + " and length is " + length);
            SocketEventObserver.getInstance().notifyReceviedData(s);
        }
    } catch (IOException e) {
        SocketEventObserver.getInstance().notifyConnectObserverFailed();
        e.printStackTrace();
    } catch (InterruptedException e) {
        e.printStackTrace();
        SocketEventObserver.getInstance().notifyConnectObserverFailed();
    }
}
```

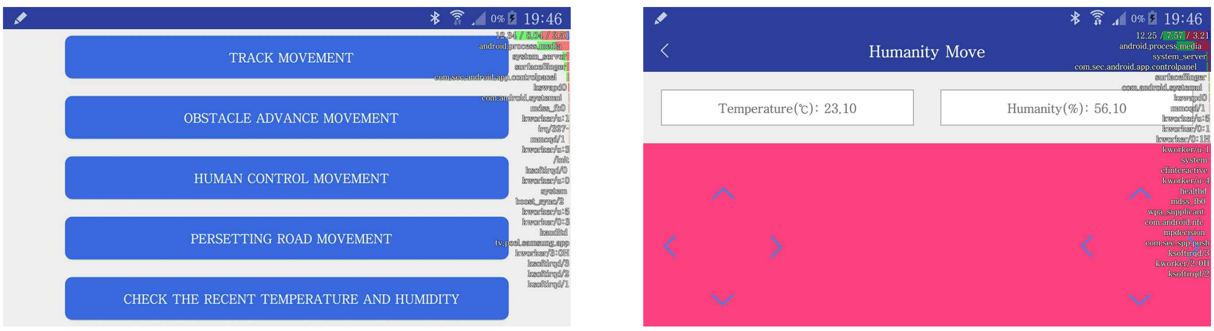
Figure 5.11: Code segment of Client-end Socket connection

```
public void notifyRecevidedData(final String data) {
    synchronized (socketEventListeners) {
        handler.post(() → {
                if (data.equals("z#")) {//if recieved data is "z#", process the reaction of MQ-2 directly.
                    if (context == null) {
                        throw new IllegalArgumentException("please init context before this method");
                    }
                    Intent intent = new Intent(context, WarnViewActivity.class);//here,enter the warnViewActivity
                    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                    context.startActivity(intent);
                    return;
                }
                for (int index = 0; index < socketEventListeners.size(); index++) {
                    socketEventListeners.get(index).socketRecevidedData(data);
                }
        });
    }
}
```

Figure 5.12: Code segment of Client-end process received Socket data

The Socket connection is two-way, Separating the input and output into different thread for decoding and encoding. In order to avoiding the blocking, preparing a string type array for user requests, and encoded requests are pushed to the end of the queue, the function of the notifyRecevidedData in the figure is that: if the received data is “z#”, enter the warnActivity, otherwise, encapsulate Socket data.

After completing the Socket connection, enter the homActivity, there is not any Socket data transformation, but the response of five buttons through five intent view. The baseMovementActivty is the parent class of three movement Activity. In this activity, humidity and temperature received are recorded, and three subclasses just override relate function to modify their private title and Socket command respectively. The humanityMoveActivity has the most complex instr-uction set, each rotation direction of motors and the camera holder has its own command. Note: in order to fix into users’ usage habits, the responding function of the eight buttons has been changed from onClick() to onTouch(). For instance, if users want to control the car to go forward, touching the “go forward” button, when the finger leaves the screen, the car model stops. The last function is to check the MySQL data, similar with the movement mode, the data processing and Android operation are placed in an activity, but sending Socket request, ending Socket request and modifying the prevate title are in the same format as above movement modes. In fact, the assignment of 21 textareas is not as complicated as it is supposed to be, because the difficulty of this aspect has been considered when implementing the Raspberry Pi-end database operation. The data has been encoded in the format that is most convenient for Android decoding.



(a) Feature Selection View (b) Human Control View

Figure 5.13: Test the function of PIR

Chapter 6

Evaluation

The development process and testing process are carried out simultaneously, which means that each time a module of the project is finished, this module is tested immediately. The benefit is to improve the efficiency of development. For example, the physical is texted at the beginning, and it was found that the 4WD development board might cause the Raspberry Pi can not be boot properly. The whole project involves a great number of knowledge points. Every link is likely to be abnormal, which is mean that the workload of the debug will be increased. For example, when found the left-front wheel to be uncontrolled, student had to start debugging from the Android app, then the Python program in the Raspberry Pi , the left front wheel wiring, after repeated debugging, the cause of the problem was found.

6.1 TCP/IP Socket Connection

To test the whole IoT system, the first thing to test is the network connection between the Raspberry Pi Socket-server end and the Android Socket-client end app. Connect the the Raspberry Pi to the monitor and use “ifconfig” to query the IP address of the server-end, boot the related service software at the terminal. Modified the main Python program by new IP address and run its.

Launch the Android app and input the IP address and port address on the server side in the mainActivity. If user click on the “connect” button, normally a “connect successful” pop-up appear on screen and enter the second activity.

The related error code and their solution:

- Error code: “AttributeError: “NoneType” onbject has no attribute ‘send’ ”. The reason for this error is the “pigpio” GPIO driver is not launched correctly. The AM2302 sensor can not obtain meaningful temperature value and humidity value. The solution is that try to restart this service in terminal by “sudo pigpio”.
- Error code: “OSError: [Errno 99] Cannot assign requested address”. The reason for this error is the current program’s IP address is not allowed to become TCP/IP socket server-end address. It is likely that the IP address in the program is not the address of the localhost under this gateway. The solution is that make sure the Raspberry Pi is connected to the network and query the IP address. Meanwhile, the user-defined socket connection in this project is likely to be regarded as an insecure

network connection by the system, so it is necessary to ensure that the firewall on the Socket end is closed. Users can use UFW to manipulate the firewall, which is a host side iptables firewall configuration tool.

- Error code: "OSError: [Errno 98] Address already in use". The reason is the currently assigned port address is already occupied.

Evaluation: the Socket connection between the Raspberry Pi and the android app is stable. Under normal circumstances, network delay caused by the connection can be ignored, it means that once the user send any request(click a button) on Android app, the Raspberry Pi Python program can adapted immediately. However, incorrect operation by the user also has the potential to cause communication jams. Thus, As a means of communication. The socket connection of this project can work smoothly as a basis for other functions.

6.2 Movement and related sensors testing

In the case of that the battery of the L298N is abundant, the car model is able to move on the smooth ground. But due to the car model is too heavy, the steering in place is not very smooth, which is not considered in the selection of materials, and it is impossible to find a good solution by software development. therefore, try to avoid using spin_right() and spin_left() function.

After a period of running this car model, the power of L298n is consumed. When the battery is low, the speed of the car will slow down, after testing, the first wheel affected was the left-front one, then the right-front one. In other words, when the power is not enough, the motor that control this two wheels will lose control by the L298n earlier. Once this problem is found, replace the battery of L298n immediately.

There are two sensor that related to movement, tracking sensor and the HC-SR04 ultrasonic sensor. The function test is not complex, simply placing the car model on the ground, and send the request by clicking on "Obstacles avoidance movement" button on the Android end. Then, then car model could move according to the behavior prescribed by the Python program and avoid obstacles.

For tracking movement, after launch the the Raspberry Pi car model, the indicator lights of tracking sensor will light up, adjust the height of the car model, until all the four lights are out, put the black insulating tape directly under probes, and corresponding indicator light will be on. If the car model is in the working mode of Tracking Movement, the wheel start spinning, which has been shown in Figure 6.1. For the example of Figure 6.1, the three probes on the left sense the change of the ground's color, and the their indicator lights are lit. this situation simulates a right-angled bend, so the wheel on the left drives the car to right, while the wheel on the right does not provide voltage.

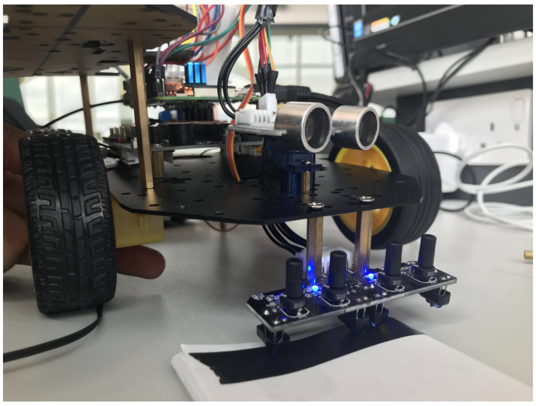
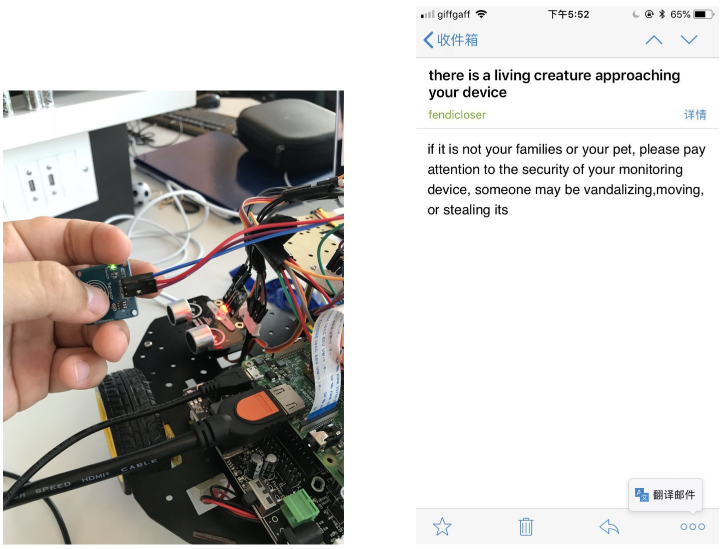


Figure 6.1: testing the function of tracking movement

6.3 Anti-theft Function Testing

The model of PIR introduced in this project is HC-SR501, whose monitoring range is about 3 to 7 metre. Due to its special function, and in order to simplify the test difficulty, student introduce a touch sensor to replace this PIR module. The reason for this is that the touch sensor and PIR work work in a similar way, both of them can be regarded as a trigger or a switch in this project; meanwhile, they has same number of pins, and only a input signal pin need to be connected with the Raspberry Pi GPIO.

The actual testing method is that, after the replacement is completed, touching the sensing area of the touch sensor by finger briefly. The LED indicator will be light on after a short delay, at the same time, receive a reminder QQ email on PC or mobile phone. The reason why it can only be touched for a short time is the program does not limit the number or time interval for sent emails. If the finger is placed on the sensing area for a long time, repeated mails are sent in a short time. If the humanization of this project need to be improved,control the time interval between each two mail by introducing database or timestamp. The test results has been shown in Figure 6.2.



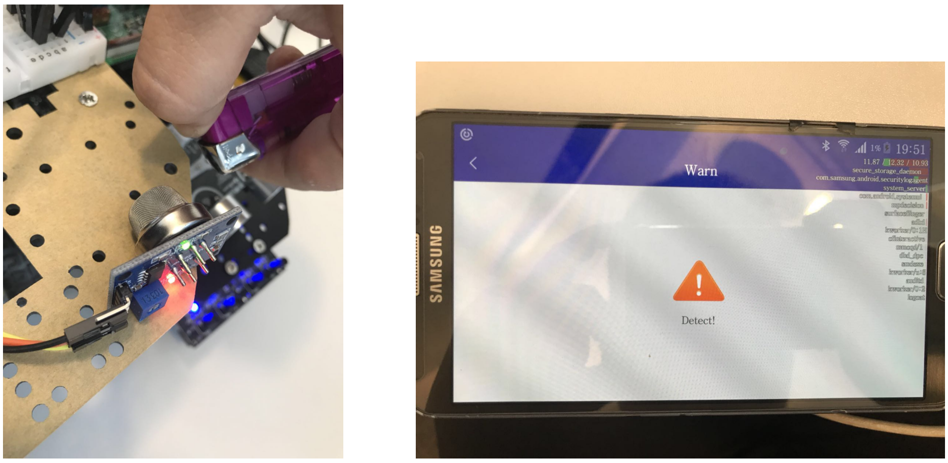
(a) Touching the sensing area (b) QQ email content

Figure 6.2: Test the function of PIR

6.4 The harmful gas and warning Testing

After the Raspberry Pi is powered, the power indicator of the MQ-2 gas sensor will light up and the the resistance inside of the MQ-2 will take about 20 seconds to warm up.

The test used for testing the function of MQ-2 is to place the lighter that is releasing the gas close to the wire mesh of the MQ-2 sensor. When the concentration of harmful gases around the sensor exceeds the critical value, the indicator light on the MQ-2 will illuminate, which has been shown in Figure 6.3 (a). Meanwhile, the Android client app immediately responds, enter the warningActivity and the phone starts to vibrate. The warningActivity of client has been shown in Figure 6.3 (b):



(a) Testing MQ-2 (b) Warning Activity of Android

Figure 6.3: Harmful gas warning

6.5 Camera Testing

There are two kinds of camera in this project, and due to their different connection form to the Raspberry Pi, both cameras can be used simultaneously.

Firstly, test whether the two cameras can shoot video normally, the first one is the USB camera. Open the browser on the PC, enter the IP address, the correctly captured video can be saw. Since the Motion software has been set to be a daemon, if this system thread is required to be shutted down, input the command line in the terminal: "sudo killall -TERM motion". Another is the camera module of the Raspberry Pi, similarly, after entering the relevant command in terminal, the captured video is displayed immediately on the Raspberry Pi. At the same time, open the VLC player on PC, and input the URL of the video stream, the video can be displayed normally. The actual captured video is shown in Figure 6.4 below. As the figure indicates, in contrast, the image captured by the Raspberry Pi camera is very smooth, and there is never a snowflakes screen or a stagnation, however, this camera has a fixed delay of roughly three seconds. For the USB camera, the network delay is not a serious problem, the video appears in Google Chrome immediately, in addition, the video captured has a higher resolution. However, even if increase the value of the "stream_maxrate" parameter , which has been mentioned in

Chapter 5, to 100, the video also has obvious stagnation, which is mean that the noise immunity of this method is weak.

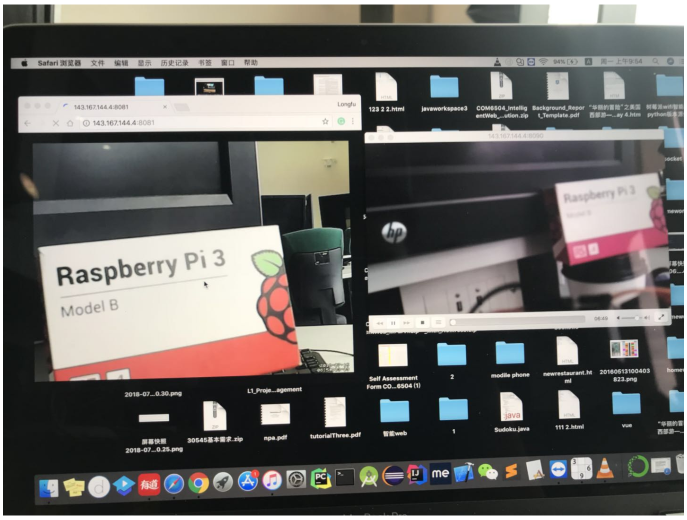


Figure 6.4: Left-side is taken by the USB camera, Right-side is taken by Pi camera module

Further, test whether the video stream of two camera can be displayed normally on the Android app. First of all, the USB camera, in previous implementation, plan to play the video stream by the Android webview. The test result is that the Android webview cannot interpret the URL of the video stream, so this method has failed. Student test the second method, which is combined with the Raspberry Pi camera, the VLC technology and the Android videoview. The Android phone used for testing in this project is Samsung Note 3, and the version of Android is Lollipop. The result of the test is disappointing, and the URL of the video also cannot be played on the videoview of app.

If the video stream must be displayed by the Google chrome or VLC player of PC of mobile phone, rather than in the developed app, the actual performance of whole system will stuff a obviously negative influence. The current situation indicates that the two kind of the camera can shoot video normally and generate the video stream, so the problem is the Android app. For USB camera, student used webview, which is able to show simple and standard online content in this view. For example, input the URL of google search engine (http://www.google.co.uk) in the textarea of camera URL, further, the related website content can shown in webviews. However, it is possible that the video stream data is not the simple online content, Google Android may regards the "interpret Camera URL' as a request. A plain webview cannot respond. The likely way to be accepted is take place the webview() to setWebViewClient(), which works for deal with various received requests and notification. Then, for PiCam, the implementation process of this method is much simpler and more standard, videoview is not a complex screen view, so student think that the problem should be inside of the Android operating system. For example, the version of Android, the version of tested phone is android 5.0.This version and later versions may have modified the RTSP Structure. But, student do not have available past Android version, even if the test is successfully completed, the backward android version of the phone has been eliminated.

6.6 Database testing

In movementSelectionActivity, click on bottom button “Check the recent temperature and Humidity”, and enter the database data query activity, there is indeed reasonable data to be displayed on the screen, which has been shown in figure 6.5 below. similarly , student can also access the MySQL database client-end to query same information through the terminal of the Raspberry Pi.

Analyzing the data in figure 6.5, despite of the fact that there is indeed some value of temperature and humidity is “-999” in MySQL, but there is no such data in the Android app side. The conclusion is that incorrect data has been filtered out by server-end Python program. To facilitate testing, the time interval has been reduced from 24 hours (100000) to one minute (100).

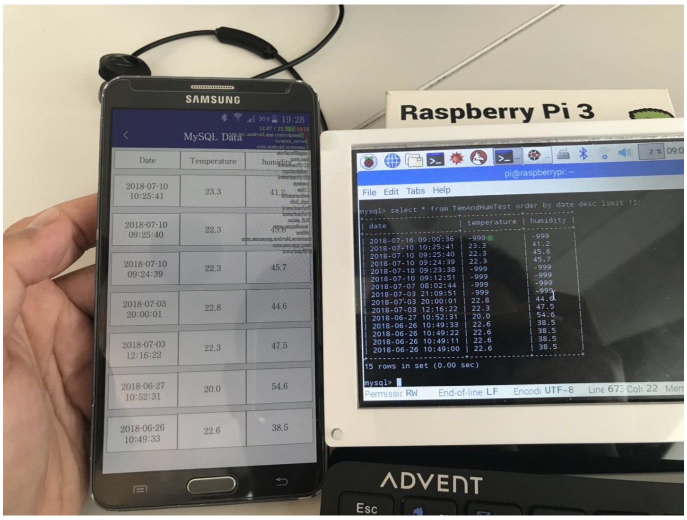


Figure 6.5: comparation of the Client-side data and Database data

6.7 Functional System Testing

Despite of the fact that each function of the whole system has been tested respectively, there is no guarantee that the program will always be run properly. It is necessary to combine functional module together and do a comprehensive system test. Testing the fully integrated applications, including external periph
-erals in order to check how components interact with one another and with the system as a whole, this is also called end to end scenario testing[56].

First of all, student need to extract the tastable features, all user-feasible option are represent possible testable features, in this system, the feature includes the Socket connection, the Main interface, display the video stream data, tracking movement, Obstacle avoidance movement, human control move -ment, presetting path movement, camera holder rotation, the handling mecha -nism of emergencies, capture the value of temperature and humidity as well as the operation of MySQL database.

After obtaining above testable feature, define the category and partition of each feature. For example, the Socket connection, the textarea that used for input the IP address and the port address is required to be tested. In detail: if the IP address and port number of the server side are entered correctly, how will the system respond? If the illegal address is entered, will the system prompt help?

After defining the categories and partitions, test cases are generated. Each partition is one possibility for test cases. However it is not possible to test each test cases. This is not a realistic approach. To reduce the test cases, they are eliminated meaningless test cases. For example length of string can not be less than zero. There are three type of property that are used to reduce combinations. These are property of if, property of error and property of single. Also in here it can be defined a property for property of if. For another, as what has been mentioned above, a certain number of buttons or textarea of four kinds of movement are derived from a basic activity. Therefore, in order to reduce the number of test case, student just consider the back button and two environmental value of the tracking movement activity. In fact, all three other activities have these categories. In this section first of all it is calculated the all possible test cases and then they are eliminated and reduce these test cases using these properties. In order to simply the testing process, student change the PIR to the touch sensor again.

Table 6.1: testable features

Component	Parameters	Categories	Partitions	Constraints	ID
Socket Connection	IP address text area	Input String IP	length <= 0	[Error]	
			Correct IP	[Single]	AA1
			Incorrect IP	[Error]	
	Port address text area	Input String Port	length <= 0	[Error]	
			Correct Port	[Single]	AB1
			Incorrect Port	[Error]	
	Camera URL	Input string URL	length <= 0		AC1
			Correct URL		AC2
			Incorrect URL	[Error]	
	Button	Area	Connect	[Single]	AD1
Android Phone	Home button	Click time	time=1		HA1
			time>=2		HA2
	Android screen	App icon	clicked		HB1
	Socket Client	Connected number	Number=1		HC1
			Number>=2,Number<5		HC2
			Number>5	[Error]	

From the perspective of the Android end, the overwhelming majority function of the system need to complete Socket connection firstly. In addition, we also need to monitor the adaptability of the app in the Android system, in other words,, some Android features also need to be tested. The third point that need to be considered before generate test features is that the activity of the three movements is inherited by a basic movement activity, so in order to optimize the testing process. Only the relevant components of tracking movement will be set as test features. Presetting path movement and obstacle avoidance movement have similar implementation effect, so they are omitted.

Table 6.2: testable features

Component	Parameters	Categories	Partitions	Constraints	ID
Main interface	Button	area	Tracking movement	[Single]	BA1
			Obstacle avoidance movement	[Single]	BA2
			Human control movement	[Single]	BA3
			Presetting Path movement	[Single]	BA4
			Database	[Single]	BA5
Tracking movement	Textarea	temperature	1111	[error]	
			Temperature value		CA1
		humidity	1111	[error]	
			Humidity value		CB1
	Button	image	back	[single]	CC1
Human control movement	Button	Moving direction	Go forward		DA1
			Go back		DA2
			Turn right		DA3
			Turn left		DA4
		Camera holder direction	Turn right		DB1
			Turn left		DB2
			Turn front		DB3
			Turn up		DB4
Event	External environment	Sensor operation	MQ-2 to Android warning activity		EA1
			PIR to send email		EA2
			AM2302 store data to MySQL		EA3
	Camera	Video URL	PiCam video in VLC player		EB1
			USB camera video in browser		EB2
Warning activity	Button	area	back	[Singal]	FA1
Database Activity	Textarea	date			GA1
		humidity	-999	[Error]	
			1111	[Error]	
			Humidity value		GB1
		temperature	-999	[Error]	
			1111	[Error]	
			Temperature value		GC1

Table 6.3: Test Case Flow

Test	Descriptions
HB1-HA1	Open the Android app, return to the Android main interface without do any operation
HB1-HA2	Open the Android app, open the Android task manager, forcibly terminates the app
HB1-AA1-AB1-AC1-AD1-HA1	Open the Android app, complete the Socket connection once, return to the Android main interface
HB1-AA1-AB1-AC1-AD1 -BA1-HA1-HB1-BA1	Run app, complete the socket connection , return to the Android main interface, run the app, request a Socket service
HB1-AA1-AB1-AC1-AD1-HA2-HB1-AA1-AB1-AC1-AD1	Run the app, complete a Socket connection, forcibly terminates the app, restart the app, complete the Socket connection
HC1-HB1-AA1-AB1-AC1-AD1-HC2-HB1-AA1-AB1-AC1-AD1	Run the app, complete the Socket connection, meanwhile, run the app in another phone, complete the Socket connection
HC1-HB1-AA1-AB1-AC1-AD1-BA1-HC2-HB1-AA1-AB1-AC1-AD1-BA1	Request a Socket service in a phone, complete another Socket on another phone
AA1-AB1-AC1-AD1	Complete a Socket connection. Note: we know the app cannot display the video of cameras, but did not delete the textarea of cam URL and the videoview, because relevant fucntion are still possible to be implemented
AA1-AB1-AC1-AD1-BA1-CC1	Complete a Socket connection, complete a tracking movement
AA1-AB1-AC1-AD1-BA1-CA1-CB1-CC1	Complete a Socket connection, Complete the tracking movement, check the value of temperature and humidity, end the service
AA1-AB1-AC1-AD1-BA1-CC1-BA2-CC1-BA3-CC1-BA4-CC1	Complete a Socket connection, transiting the work mode among four kinds of movement
AA1-AB1-AC1-AD1-BA1-CA1-CB1-CC1-BA2-CA1-CB1-CC1-BA3-CA1-CB1-CC1-BA4-CA1-CB1-CC1	Complete a Socket connection, transiting the work mode among four kinds of movement, meanwhile, check the value of temperature and humidity in each activity
AA1-AB1-AC1-AD1-BA3-DA1-DA2-DA3-DA4-DA2-DA2-DA3-DA1-CC1	Complete a Socket connection, enter the human control activity, testing the rotation direction of the Car model repeatedly
AA1-AB1-AC1-AD1-BA3-DB1-DB2-DB3-DB4-DB2-DB2-DB3-DB2-DB3-CC1	Complete a Socket connection enter the human control activity , testing the rotation angle of the camera holder repeatedly
AA1-AB1-AC2-AD1-BA1	”Video cannot play” pop-ups appear
EB1-EB2	As long as the car model is started, the data of the two cameras can be displayed on the PC normally without socket connection.
AA1-AB1-AC2-AD1-BA1-EB1-EB2	After the connection is completed, the two ends of the socket are already transmitting information. At this time, video can still be displayed normally, proving that system can carry out simultaneous transmission of multiple video streams and the Socket connections
AA1-AB1-AC2-AD1-EA1-FA1-BA1-EA1-FA1-BA1	Activate MQ-2 sensor after the Socket connection, enter the warning activity, back to the main interface, enter a movement activity , Activate MQ-2, back to main interface, enter a movement activity again.
AA1-AB1-AC2-AD1-EA2 -BA2-EA2	Complete the Socket connection, activate the PIR(touch sensor) in main activity, receive the email, enter a movement activity, activate the PIR, receive the email.
AA1-AB1-AC2-AD1-EA1-EA2	Complete the Socket connection and activate both MQ-2 and the PIR simultaneously in main activity.

In conclusion, above content is the record of part of main function, due to the limiation of text and images, the evaluation of some funciton is not be described in this chapter, such as the display of termperature and humidity as well as the simple rotation of DC motors. However these features have been applied in other more complex function. For example, MySQL necessarily requires AM2302 normally running for retrieving the temperature value and humidity value, besides, the functions of two sensors following moving modes must be implemented on the basis of L298n and simple movement.

Overall, the essential objective of this project is to find a solution to control the Raspberry Pi hardware with mobile deivce, that is, the Android operating system controls I/O ports and software service of a embedded microprocess. Rather than how good the function of the Raspberry Pi sensors are implemented. For example, tracking sensor can only detect simple tracks, If the movement path of tracking movement is too complex, such as a crossed black insulated tape track, the car model is not able to detected normally.

Chapter 7

Improvement and Conclusion

At this point, students have basically completed the analysis, design, implementation and testing of this project. Compare to the functional requirements listed in chapter 3. What is clear is that one problem will continue to be fixed, which is find the solution of displaying camera video on Android phones. Besides, function of presetting path movement has the potential for further development, if uses are allowed to set the presetting path by themselves, although this function seems very complex, the essence is still TCP/IP Socket data communication. his project still have very strong expansibility, so some improvement can be made in the future:

- This project just provide a solution for smart home monitor device, many hardware definitely have better alternatives, such as the batteries, the DC motor, etc.
- Adding artificial intelligence algorithms to improve accuracy and flexibility of movements. Or introducing other kinds of sensors, such as infrared obstacle avoidance sensor.
- Development the IOS version of Socket client to improve the universality of the system.
- Network delay is a problem that has always plagued me in this project, the reason for this problem is the limitation of the technologies. For example: serious network delay of the Motion software as well as the backward of plain Socket network development.

From aspects listed above, the Raspberry Pi is very powerful and is the right way to select it as the project's hardware microprocessor. During the implementation and testing, student recorded many additional details, such as exception, the price and various parameters of purchased materials as well as unused technologies and parts. These issued may not be related to the final projects, but reduce barriers encountered when developing or mass production. Understanding the requirements constantly and continuously expand the requirements. At the end of the development, the Raspberry Pi still has idle gpio ports and USB interfaces, so the Raspberry Pi still has the ability to expand its functions and to calculate or store big data sets.

This project provide a kind of smart home monitoring solution. As a model of IoT, it involves circuit design, hardware programming, Socket network programming, mobile device development, multimedia device control, etc. this device uses the Raspberry Pi as the main implementation microprocessor, the

TCP/IP and HTTP as main network connection protocol, the Android app as remote controller. In the chapter of implement, applying large number of tables and simulation diagrams to record the whole process of developing, including wiring and hardware assembly, main program development, solution of some serious bugs as well as some protential problem that need to be noticed, such as the how to lauch the script on boot automatically.

In the section of system testing, all testable feature arelisted and divided by functional modules, no other problem was found except the known camera video problem, therefore, the detailed of various functions of the system was carried out, the reason for the problem were analyzed, and then the system testing was carried out following the using process of users.

Though nearly two months of development, debugging and testing, the initial requirement requirements were basically completed. In general, the implementation of each aspect is not difficult, however, compared to other project, this project requires students to have certain knowledge of logic circuit and hardware equipment. Although these also belong to the field of computer science, but relatively little attention was paid. During the implementation, the difficulty of this project mainly involves four aspects, including the transmission and display of the camera video stream, Socket connection and data transfer, operation of the MySQL database through Python, as well as the design and implementation of the server-side thread. Generally speaking, the knowledge involved is extensive and comprehensive, and the design duration of the whole project is long. In the process of implementation, students continue to introduce new technologies, abandon infeasible technology, and make innovations in some details. For example, reading and filtering of MySQL database information. The only downside to the tried part is that neither USB camera's video or PiCamera's can be displayed on the Android Client side, even though they can already be displayed on a PC. The project always follows the principle of low-cost.

Appendices

Appendix A

List of project hardware cost

Table A.1: List of project hardware cost

Name of hardware	Number	Price (GBP)
Raspberry Pi 3 model B	1	28.79
L298N DC motor driver module	1	3.49
DC motor	4	14.99
Rubber wheel	4	
4 AA battery holder	1	
Servo motor	3	7.77
MQ-2 gas sensor	1	3.99
AM2302 temperature and humidity sensor	1	6.99
HC-SR04 Ultrasonic sensor	1	3.99
Tracking sensor	1	2.28
Usb camera	1	8.21
Pi camera and holder	1	29.99
Car model chassis and supports	2	5.99
Half breadboard	1	6.84
Jump wires	70	4.99
Power bank	1	22.99
Recharge pro charge and 4 rechargeable AA batteries	1	13.99
HC-SR501	1	3.15
LED indicator	1	1.25
Summary		169.69

Notice: The table only lists the name, the price and the quantity of the hardware used in the final product. The general hardware used in actual development process, such as keyboard and display, and the abandoned sensors and components, such as the development board of Raspberry Pi and MCP3208 are not be counted.

Appendix B

Real Figure of the Car Model

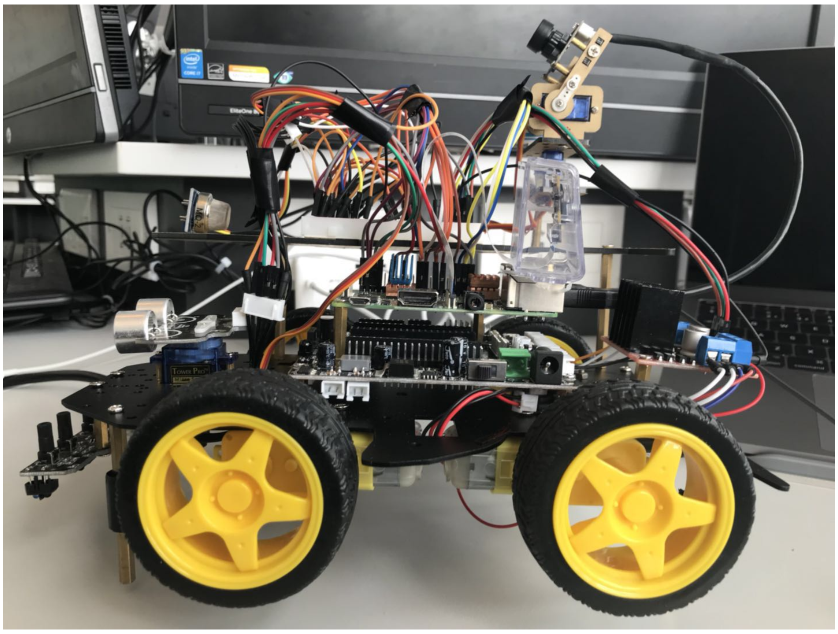


Figure B.1: profile view of the car model

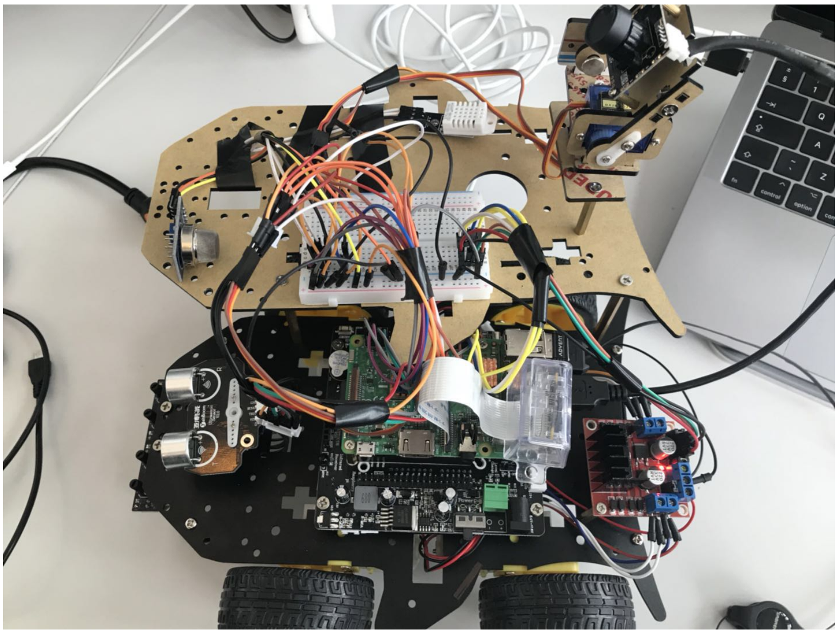


Figure B.2: front view of the car model

Appendix C

Reference

[**1**] Feizhou, Y. and Dongkai, Y. (2010). A review of the Internet of Things. Beijing. Electronic industry press.

[**2**] ZhiHu,. (2012). What's the difference between RFID and NFC? [online]. Zhi Hu. [Viewed 27 April 2018]. Available from: https://www.zhihu.com/question/19931182/answer/14796454.

[**3**] Agarwal,T.,(2015). Electronics, Projects, Focus. Wireless Sensor Networks and their Applications [online].[Viewed 26 April 2018]. Available from: https://www.elprocus.com/introduction-to-wire less-sensor-networks-types -and-applications/

[**4**] International Electrotechnical Commission (2014). Internet of Things: Wireless Sensor Networks. Switzerland: IEC, Geneva(24-25).

[**5**] David, T.,(2015). My Posting Bubble.How Many IPv6 Addresses are there? [online]. 5 December 2015. [Viewed 27 April 2018]. Available from: https://www.myhostingbubble.com/blog/2015/12/09/how-many-ipv6-addresses-are-there/

[**6**] Swati, K.,(2016). Analytics Vidhya. 10 Real World Applications of Internet of Things (IoT)- Explained in videos[online]. 26 August 2016. [Viewed 28 April 2018]. Available from: https://www.analyticsvidhya .com/blog/2016/08/10-youtube-videos-explaining-the-real-world-applications-of-internet-of-things-iot/

[**7**] Nicola,K. (2003). Smart Home - a definition[online]. Available from: http://www.housingcare.org/downloads/kbase/2545.pdf

[**8**] Molly Edmonds "How Smart Homes Work - Setting Up a Smart Home"[online]. available from: http://home.howstuffworks.com/home-improvement/energy-efficiency/smart-home1.html

[**9**] Erec, "Home automation systems" in Reporter, February, 2002, Vol. 19

[**10**] Innovation Department. (2008). ZigBee - WiFi Coexistence. 38000 Grenoble, France: Schneider Electric.

[**11**] ZhiHu,.(2017). Painless Machine Learning.[online]. 22 november 2017.[Viewed 1 May 2018]. Available from: https://zhuanlan.zhihu.com/p/22464594

[**12**] Jatin Parekh. (2017). Network World. WiFis evolving role in IoT[online]. 11 May 2017.[Viewed 30 April 2018]. Available from: https://www.networkworld.com/article/3196191/lan-wan/wifi-s-evolving-role-in-iot.html

[**13**] WayBack Machine. (2017). How Bluetooth technology work[online]. 17 January 2008 to 20 December 2017. [Viewed 6 April 2018]. Available from: https://web.archive.org/web/20080220032544/http ://www.bluetooth.com:80/Bluetooth/Technology/Works/

[**14**] Guo, B. et al.,(2013). Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things. Journal of Network and Computer Application[online]. Vol.36(6), pp.1531-1539. [Viewed 27 April 2018]. Available from: DOI: 10.1016/j.jnca.2012.12.028

[**15**] Gubbi,J. Et al.,(2013). Internet of thing (IoT): A vision, architectural elements, and future directions. Future Generation Computer System[online]. Vol.29(7), pp.1645-1660. [Viewed 3 May 2018]. Available from: DOI: 10.1016/j.future.2013.01.010

[**16**] Addimulam, S. C. (2015). Smart home control using raspberry pi in internet of things environment (Order No. 1603977). Available from ProQuest Dissertations & Theses A&I. (1738629611). Retrieved from https://search.proquest.com/docview/1738629611?accountid=13828

[**17**] Android Overtakes Windows for First Time StatCounter [online] Available from: https://www.businesswire.com/news/home/20170403005635 /en/Android-Overtakes-Windows-Time--StatCounter%E3%80%82

[**18**] Dave, N. (2017). The Raspberry Pi is suddenly the third best-selling computer ever. The Inquirer[online]. 20 March 2017 [viewed 9 Apirl 2018]. Available from: https://www.theinquirer.net/inquirer/new s/3006780/the-raspberry-pi-is-suddenly-the-third-best-selling-computer-ever

[**19**] Patrick, D,Justo (2015). Raspberry Pi or Arduino Uno? One Simple Rule to Choose the Right Board. Make:[online]. 4 December 2015 [view 18 March 2018] Available from: https://makezine.co m/2015/12/04/admittedly -simplistic-guide-raspberry-pi-vs-arduino/

[**20**] Using Python with Arduino - Controlling an LED [online]. Available from: https://circuit digest.com/microcontroller-projects/arduino-python-tutorial

[**21**] Jason, K. (2016). WHY IS BEAGLEBONE OPEN HARDWAR E?[online].Available at: https://beagleboard.org/static/presentations/ makerfaire-WhyIsBeagleBoneOpenHardware.pdf

[**22**] Steven Barrett; Jason Kridner, "Bad to the Bone:Crafting Electronic Systems with BeagleBone Black, Second Edition," in Bad to the Bone:Crafting Electronic Systems with BeagleBone Black, Second Edition , 1, Morgan & Claypool, 2015, pp.417- doi: 10.2200/S00675ED1V01Y201509DCS046

[**23**] Alasdair,A.(2016). Meet the New Raspberry Pi 3 A 64-bit Pi with Built-in Wireless and Bluetooth LE. Make:[online]. 28 February 2016 [viewed 3 Apirl 2018]. Available from: https://makezine.com/2016/02/28/m eet-the-new-raspberry-pi-3/

[**24**] Raspberry PI GPIO.FAQS. (2018). Available from: https://www.raspberry pi.org/help/faqs/#softwareLanguages

[**25**] Simon Monk. 2014. Raspberry Pi Cookbook (1st ed.). O'Reilly Media, Inc..

[**26**] BST-4WD the Raspberry Pi car development board.(2017).[online] Available from: https://detail.1688.com/offer/558555534154.html

[**27**] Sushant, NJune. (2016). Raspberry Pi 3 Tutorial 12 GPIO DHT11 Digital Temperature and Humidity Sensor. Module143 [online]. 12 June 2016. [view 28 February 2018]. Available from: http: //invent.module143.com/daskal_tutorial/raspberry-pi-3-gpiodht11-digital-temperature-humidity-sensor/

[**28**] HOW TO SET UP THE DHT11 HUMIDITY SENSOR ON THE RASPBERRY PI. (2016).[online] Available from: http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/

[**29**] HC-SR04 User Guide. (2017).[online]. Available from: https://elecfreaks.com/estore/download/EF03085-HC-SR04_Ultrasonic_Module_User_Guide.pdf

[**30**] Sensorkit,W. (2017). KY033 Tracking sensor module. Sensor Kit X30.(2017) [view 15 May 2018]. Available from: http://sensorkit.en.joy-it.net/index.php?title=KY-033_Tracking_sensor_module

[**31**] Configure and read out the Raspberry Pi gas sensor (MQ-X).(2017).[online]. Available from: https://tutorials-raspberrypi.com/configure-and-read-out-the-raspberry-pi-gas-sensor-mq-x/

[**32**] The working principle of MQ-2.[online]. Available from: http://www.doc88.com/p6921575829543.html

[**33**] Carla,S.(2014).How to Operate Linux Spycams With Motion.LINUX.COM[online]. 10 July 2014.[View 10 Apirl 2018]. Available from: https://www.linux.com/learn/how-operate-linux-spycams-motion

[**34**] DC Motor or Direct Current Motor[online]. Available from: https://www.electrical4u.com/ dc-motor-or-direct-current-motor/

[**35**] Dejan,N.(2014).Arduino DC Motor Control Tutorial L298N — PWM — H-Bridge. HOW SERVO MOTORS WORK & HOW TO CONTROL SERVOS USING ARDUINO[online] Avaliable from: https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge.

[**36**] The comprehensive GPIO Pinout guide for the Raspberry Pi.,(2015)[online]. Available from: https://pinout.xyz/pinout/ground

[**37**] Michal,.Miroslav,M.Pavel,T.,(2017).DC motors and servo-motors controlled by Raspberry Pi 2B. 21st International Conference on Circuits, Systems, Communications and Computers (CSCC 2017).125(5),2-4. DOI:10.1051/matecconf/201712502025

[**38**] Controlling Servo Motors(2013).[online]. Available from: http://razzpisampler.oreilly.com/ch05.html

[**39**] Xiren,X.(2008).Computer Network.Electronic industry press.402(5),343-350. ISBN: 9787121053863

[**40**] Computer blog for Nepal students(2012).advantage and disadvantage of modular programming[online].15 May 2012.[View 2 May 2018].Available from: http://pnagila.blogspot.co.uk/2012/05/advantages-and-di sadvantages-of-modular_15.html

[**41**] Karaca, S. et al,.(2016). A low cost smart security and home automation system employing an embedded server and a wireless sensor network. IEEE International Conference on Consumer Electronics. DOI: 10.1109ICCEBer lin.2016.7684721

[**42**] Comparing Raspberry Pi and BeagleBone Black.Available from: https:// makezine.com/2014/02/25/how-to-choose-the-right-platform-raspberry-pi -or-beaglebone-black/

[**43**] BTS-4WD development board. Available from: https://item.taobao.comi tem.htm?spm=a1z10.5-c-s.w4002-14871668706.15.279c78f4qkaq Ej&id=559029701723

[**44**] PiCam. Available from: https://pi.gate.ac.uk/pages/snowcam.html

[**45**] The Raspberry Pi USB camera. Available from: https:https://item.taoba o.com/item.htm?spm=2013.1.20141003.9.30bf0c599arTb9&scm=1007.1001 1.70203.100200300000001&id=561420113647&pvid=afc9b 0f0-9e75-410d-9663-ae1961b5ca7f

[**46**] L298N DC motor driver. Available from: https://www.raspberrypi.org/fo rums/viewtopic.php?t=152981

[**47**] Servo Motor PWM mode. Available from:http://razzpisampler.oreilly.com /ch05.html

[**48**] Raspberry Pi 3 Model B. Available from: http://shumeipai.nxez.com/2018/0 3/14/raspberry-pi-3-model-bplus-sale-now.html

[**49**] Denis,D.(2010). Is ZigBee or Wi-Fi More Cost-Effective and Efficient for Smart Grid Communications in the Home?[online]. [Viewed 12 May 2018]Available from: https://energypriorities.com/entries/2010/1 2/ge_zigbee_vs_wi-fi_han.php

[**50**] Juan,L.(2015). Testing Accuracy of Long-Range Ultrasonic Sensors for Olive Tree Canopy Measurements[online].[Viewed 18 July 2018]Available from:https://doi.org/10.3390/s150202902

[**51**] W-E.(2015).DHT init sequence wrongP[online].[Viewed 18 July 2018]Available from: https://github.com/adafruit/DHT-sensor-library/issues/48

[**52**] Send SMS Messages from Raspberry Pi. Available from: https://www.rasp berrypi.org/forums/viewtopic.php?t=69286

[**53**] Send e-mail by SMTP. Available from: https://www.liaoxuefeng.com/wiki /001374738125095c955c1e6d8bb493182103fac9270762a000/001386832745 198026a685614e7462fb57dbf733cc9f3ad000

[**54**] human motion infrared sensor HC-SR501. Available from: https://blog.csd n.net/sinat_33523805/article/details/78701878

[**55**] Basic knowledge analysis of human proximity sensors. Avaiable from: http://www.elecfans.com/yuanqijian/sensor/20180207631907.html

[**56**] What is System Testing? Avaiable from: https://www.youtube.com/watch?v=CC0He-SS-Do&frags=pl%2Cwn