

Final Project: Controlled Manipulator Based on Speech Recognition

1854116 Mingzhi Zhu

1851788 Laiyu Yang

1853539 Qixuan Yang

July 4, 2021

Contents

1 Brief Description	3
1.1 Profile	3
1.2 Architecture Design	3
1.2.1 Hardware Design	3
1.2.2 Software Design	4
1.3 Module List and Module Function Description	4
1.3.1 Hardware Modules	4
1.3.2 Software Modules	7
2 Implemented Requirements	7
2.1 Move	7
2.2 Path Planning	7
2.3 GUI	9
2.4 One Click Return	10
2.5 Writing	10
2.6 Palletizing	10
2.7 Speech Recognition	10
3 Advantages and Disadvantages	11
3.1 Advantages	11

3.2 Disadvantages	12
4 Improvement	12
5 Development Environment	12
5.1 Operating System	12
5.2 Development Tool	13
5.3 Development Language	13
6 Core Code	13
6.1 Change Audio File Sampling Rate	13
6.2 Move Detailed Data Computing	13
6.3 User Interface Design	14
6.4 Realization of Speech Recognition	14
6.5 Realization of Serial Communication	14
7 Preview Video	14

1 Brief Description

1.1 Profile

Manipulator is a kind of mechanical and electronic equipment that simulates the function of human arm and wrist. It can move any object according to the requirements. It can not only move in a simple direction, but also complete a certain combination action. According to people's different needs, different combinations of moving direction and distance can be designed to meet the needs of users.

Intelligent manufacturing with industrial manipulator as the core is an important means to promote the transformation and upgrading of China's manufacturing industry. In the future, with the continuous expansion of industrial manipulator applications and the transformation of intelligent manufacturing, China's industrial manipulator has great development potential. Therefore, independent development and design of an industrial manipulator is necessary and has broad prospects.

The general manipulator is divided into software part and hardware part. As the overall control part of the whole system, the software part is mainly responsible for the user interaction part, receiving user input and controlling the hardware part to respond.

This project is a writing education manipulator, through the use of a combination of software and hardware, as well as the application of raspberry pie, to complete the basic mobile, path planning and writing operations.

1.2 Architecture Design

1.2.1 Hardware Design

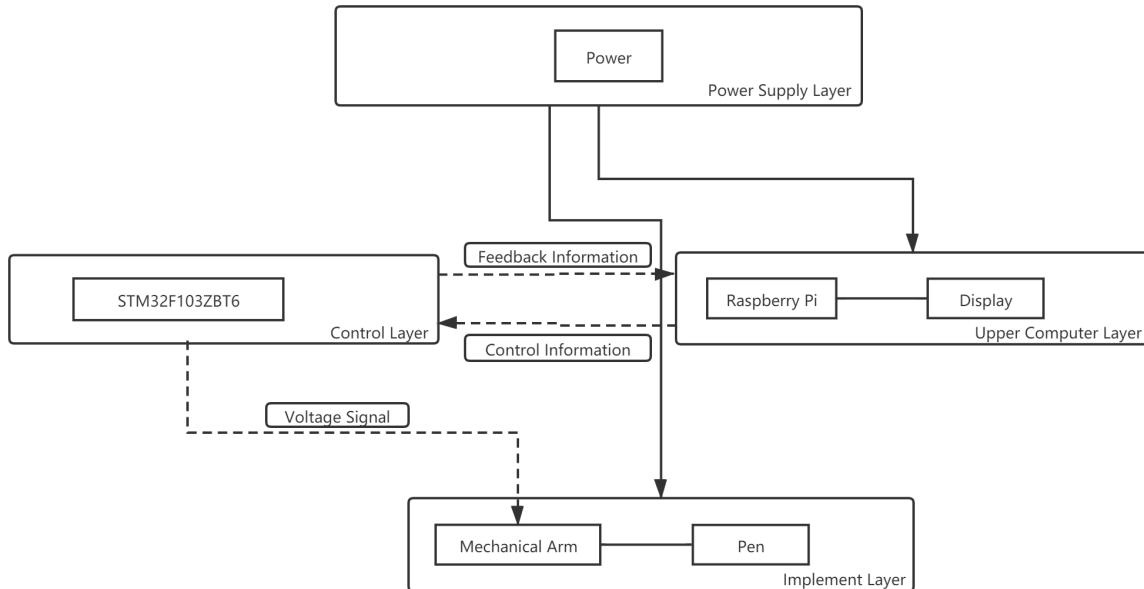


Figure 1: Hardware Architecture Diagram

The hardware structure of the system is shown in the figure1. The system consists of four layers: power supply layer, control layer, upper computer layer and execution layer.

The power supply layer selects the knob power supply as the power supply equipment for the whole system, which supplies the three motors on the manipulator and also supplies power to the raspberry pie.

The upper computer layer is composed of raspberry pie. In this system, the upper computer can send the end target of the end effector of the manipulator. Through the kinematics calculation of the upper computer, the related operations of the three motors are obtained. This information is used as the control information and transmitted to the lower computer through the serial port, that is, the control layer. At the same time, raspberry pie can be connected to the display screen to design gui interface for intuitive operation visualization.

The control layer is composed of motion control board stm32f103zbt6, which connects three motors. The control layer receives the relevant control information (including motor number and rotation angle) sent by the upper computer layer through the serial port, and outputs the pulse signal to control the motor movement. At the same time, the control layer can also send feedback information to the upper computer layer to explain the control effect.

The execution layer is composed of a mechanical arm and a paint barrel. The executive layer receives the voltage signal from the control layer to enable the motor. Under the linkage movement of the three motors, it realizes the control of the paint barrel operated by the end actuator, and realizes the target requirements of the upper computer.

1.2.2 Software Design

The software structure diagram of the system is shown in the figure2. In the system, the upper computer layer and control layer need to carry out software programming.

In the upper computer layer, the system operator inputs the relevant operation information of the manipulator to the upper computer layer in the speech recognition interface or GUI interface. In the upper computer layer, the control information is calculated by kinematics, and the simple control effect is translated into the specific key motion (corresponding to the specific angle of each motor) of the manipulator in order to achieve the motion effect. The specific motion control information is transmitted to the control layer through the serial port and the designed communication protocol. Among them, the voice information is collected through the USB sound card, and can also be input into the voice file prepared locally.

After receiving the motion control information at the serial port, the control layer understands the case according to the designed communication protocol, and then controls the motor motion according to the information.

Through the timer and interrupt software, the PWM pulse signal required for motor motion is output at the GPIO pin. At the same time, the control layer can also feed back the control effect to the upper computer layer through the serial port for GUI display of the control effect.

1.3 Module List and Module Function Description

This project is divided into hardware part and software part.

1.3.1 Hardware Modules

- **42 series two-phase motor:** Connected with A4988 drive board by motor line. Two 42BYH60 two-phase stepping motors and one 42BYH47 two-phase stepping motor are used as the driving device of the manipulator. The required signal and power supply are

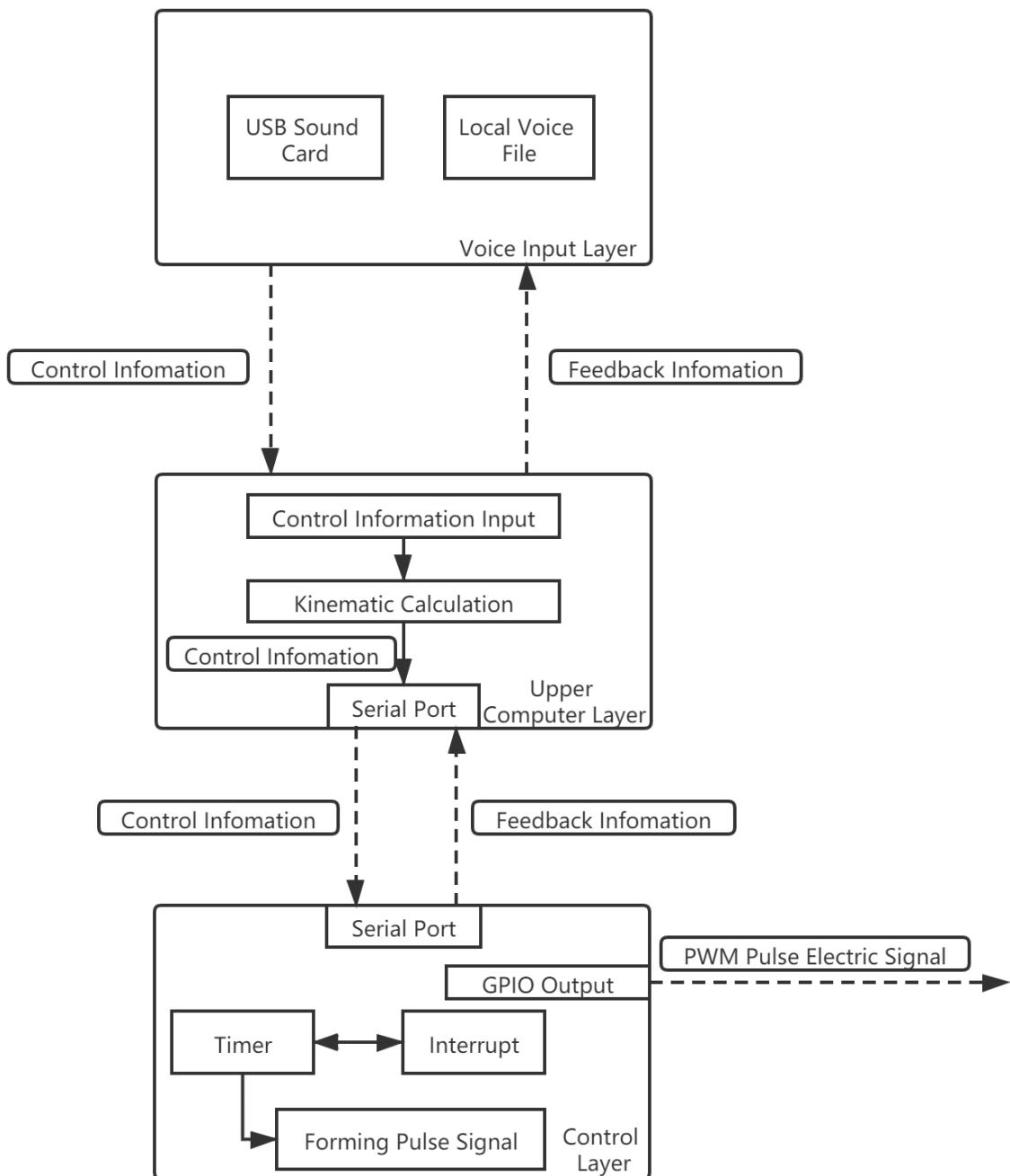


Figure 2: Software Architecture Diagram

Table 1: Hardware Core Device Selection

Module	Selection
Motor Driver	A4988 Driver Module
Electric Machinery	42BYH60/47 Two Phase Stepping Motor
Motion Control Panel	F103ZBT6Stm32 Board
Independent Smart Board	Raspberry Pie 4B
USB Sound Card	Taobao Unknown 19.9yuan Free Shipping

provided by A4988 driving board.

- **A4988 drive board:** It is connected with three 42 series two-phase stepper motors through motor lines, and three A4988 drivers are connected with GPIO port of STM32. There are three A4988 drivers on A4988 drive board, corresponding to three stepper motors. Its functions are power supply, set subdivision, STM32 square wave, control direction and speed. In fact, the control speed is to control the pulse frequency, that is, to control the delay time between high and low. When each pulse motor rotates one step, the more pulses per unit time, the greater the rotation angle per unit time, and the faster the speed is. There is a potentiometer on A4988 module, which can adjust the current by turning it.
- **F103ZBT6Stm32:** GPIO port is connected with A4988 driver board, UART serial port is connected with raspberry pie. As a motion control board, three A4988 drivers are connected through the output of GPIO port, and step and dir control signals are generated to control the movement of stepper motor.
- **Raspberry Pie 4B:** As an independent intelligent board, it is connected with the motion control board, namely F103ZBT6Stm32, through the serial port, and completes the tasks of command analysis, path planning and kinematics inverse solution through calculation.

The core module selection of the hardware part of this project is shown in Table1.



Figure 3: Raspberry Pie 4B

1.3.2 Software Modules

The core module selection of the software part of this project is shown in Table2.

Table 2: Software Core Function Description

Module Name	Module Function
Upper Computer UI Module	This part is the upper computer of the manipulator, which is completed by the raspberry pie development board with Ubuntu. It mainly realizes the GUI of human-computer interaction, displays the current position of the manipulator, and realizes the interaction with the lower computer.
Trajectory Planning Module	This part is located in the upper computer, which is mainly responsible for trajectory planning. It can realize the interpolation algorithm of straight line, cubic function interpolation of curve and Bessel curve fitting. It is called by the main function of GUI.
Kinematics Calculation Module	This part is located in the upper computer, which is mainly used as the bridge between GUI and lower computer. Through the kinematics model of the manipulator, the position information needed by the user is transformed into the operation parameters of each joint motor.
Communication Module	In this part, the communication between the upper computer and the lower computer is realized. The upper computer transmits the operation instructions of the motor to the lower computer, and the lower computer feeds back the state of each joint of the manipulator to the upper computer.
Motor Drive Module	This part is located in the lower computer, which converts the instructions transmitted by the upper computer into the status of each IO port of the drive motor, and then drives the motor drive board to realize the motor drive.

2 Implemented Requirements

2.1 Move

It can move according to the input path within the allowable error range. The software part receives the user's input and performs simple operation in six directions, and the user can adjust the moving speed and single moving distance.

2.2 Path Planning

- **MoveJ:**It is the operation of moving one unit, which is the differential unit of the following moveL function.
- **MoveL:**It is a linear moving function. By specifying the coordinates of the end point

in the established coordinate system, the manipulator moves linearly from the current position to the target position.

- **MoveP:** It is a function that simulates the function of the palletizer. It simulates a system that grabs all items from one location and moves them to another location for automatic stacking. It can be stacked in multiple layers and is generally used for forklift transportation to warehouse storage.



Figure 4: Single Chip Microcomputer and Control Board



Figure 5: Manipulator

2.3 GUI

The system controls the movement of the whole manipulator through the interface. The main functions of the interface are as follows:

- Input the coordinates and moving speed of the end effector, and click the end coordinate control button and start button to realize the function of moving the end effector to the input coordinates.
- Input the movement direction(x, y, z), the coordinates of the movement distance and the movement speed, and click the single joint control button and start button to realize the function of the manipulator moving a certain distance in the input movement direction.
- Input the moving direction (x, y, z), the coordinate of the moving distance and the moving speed, and click the simulate palletizing button and start button to realize the function of simulating palletizing through the curve movement of the manipulator.
- Click the up, down, left, right, front, back button and start button to realize the function of the manipulator moving a fixed distance in a certain direction.
- Realize the display function of input information of upper computer and feedback information of lower computer in the display text box.

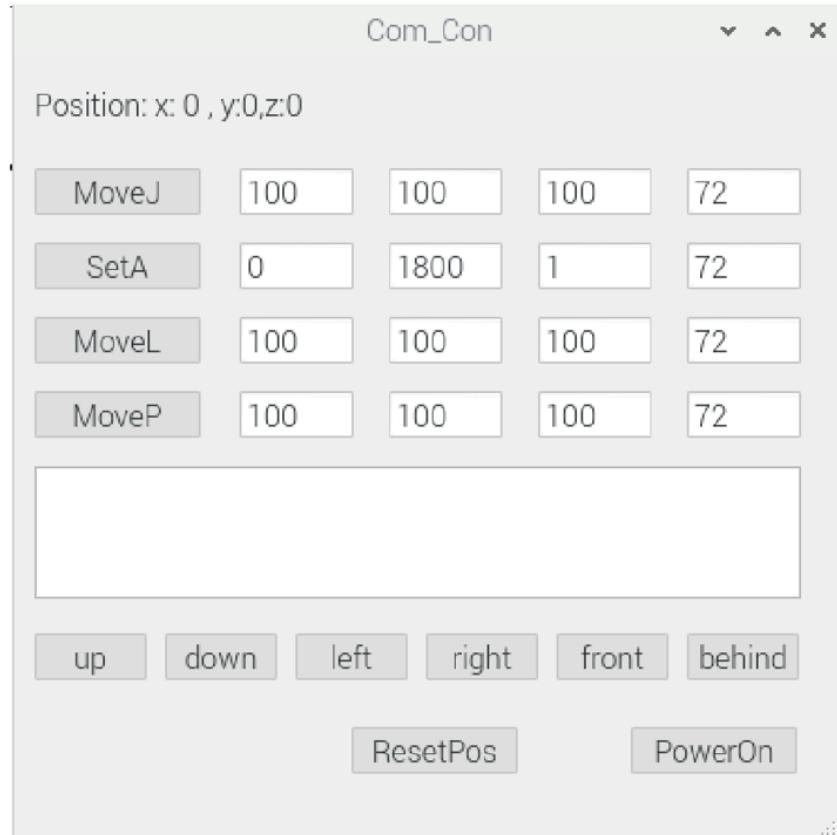


Figure 6: Direct Control GUI

2.4 One Click Return

In some cases, it may be necessary to return to the original position, so we designed a one button return function. By clicking reset on the GUI, we can return the manipulator to its original position.

2.5 Writing

In the system, the combined movement mode of numbers and letters is stored, and the robot arm can be controlled to move and write numbers and letters through speech recognition. Detailed examples can be viewed in the submitted video.

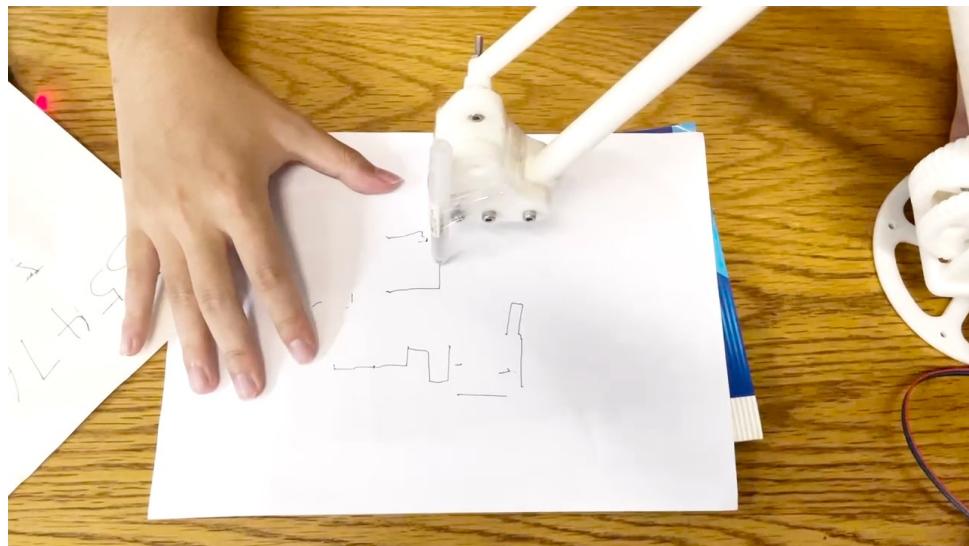


Figure 7: Writing

2.6 Palletizing

After entering the target location, it will plan a curve that does not wipe the ground, then slow down to the highest position, and then move to the target location in the same way. Simulate the process of moving an object by an excavator.

2.7 Speech Recognition

The system uses Baidu voice API, which can recognize the user's real-time voice and audio files, and convert the voice into text, so as to control the movement of the manipulator. This project supports the use of speech recognition function to operate the manipulator movement, the main operations include:

- Choose up, down, left, right, front and back six directions to move, you can adjust the moving distance.
- Simulate palletizing machine function and plan reasonable path. Complete the simulation of palletizing operation.
- According to the user's voice input, write numbers or letters to meet the user's needs.



Figure 8: Speech Recognition GUI

We also provide a GUI for speech recognition system, but you need to connect raspberry pi to a monitor to see it. You can't use SSH to connect raspberry pi to see the GUI. A screenshot of the GUI is shown in Figure 7.

3 Advantages and Disadvantages

3.1 Advantages

- Using manipulator instead of manual operation can not only improve the efficiency, but also has a high fault tolerance rate.
- The mechanical arm can achieve large-scale mass production, reduce labor costs and technical training costs, optimize cost control, and the product price has market competitiveness.
- Professional and technical team can quickly tailor automation solutions according to different product and process requirements of users.
- The robot arm has universal design characteristics, which can not only meet the automatic production of "small variety and large batch" workpieces, but also meet the application

requirements of "multi variety and small batch".

- The mechanical arm is composed of software and hardware, which is highly integrated, high fault tolerance, few fault points and low maintenance cost.
- The software system not only develops a simple and beautiful interface, but also provides the function of input speech recognition. The user interaction is very rich.

3.2 Disadvantages

- There is no autonomous mobile capability, and user input is needed to trigger the mobile.
- At present, it only supports writing simple numbers and letters, and it is still difficult to write Chinese characters.
- The weight of the equipment is not fully considered when making the manipulator. The base of the manipulator is unstable and may fall down.
- The wiring of the device is more complicated, and some peripherals such as a monitor, sound card, microphone, mouse, keyboard, etc. are also required.

4 Improvement

According to the analysis of our advantages and disadvantages, there are many things we can do to improve this program:

- The precision error of the X, Y, Z three-axis motion of the manipulator motor is about 0.5cm. After that, experiments and tests can be carried out continuously to optimize the software system, improve the precision, reduce the error and achieve more accurate movement.
- At present, the system only supports the writing of letters and numbers, but does not support the writing of complex Chinese characters in mobile programs. This system will continuously improve the software system, and input Chinese characters into the system thesaurus, so that the manipulator can support writing Chinese characters.
- At present, the system supports writing on the plane, and then the writing function can be extended to three-dimensional space to complete the writing and description of three-dimensional patterns.
- At present, the system supports fixed-point movement, that is, six directions of movement in three-dimensional space, and then can add rotation, expansion and other mobile functions according to the actual needs.

5 Development Environment

5.1 Operating System

- Windows 10
- MacOS
- Linux based on Raspberry Pi

5.2 Development Tool

Table 3: Hardware Core Device Selection

Name	Function
VSCode	For integrated development for raspberry pi
Baidu Voice API	Speech recognition support
STM32CUBEIDE	For developing embedded C program
Qt Designer	For drawing graphical interface
SolidWorks	For kinematics simulation for manipulator
ROS	For kinematics simulation and data output for manipulator

5.3 Development Language

- Python3.7
- C

6 Core Code

The software system part of this project includes several Python scripts. The core code of some scripts is shown below.

6.1 Change Audio File Sampling Rate

Using different recording equipment to record the audio file, the file format and sampling rate will be different, and Baidu voice API recognition of the audio file requires the sampling rate must be 16000, this project uses wav audio format, so we need to first use the audio file to WAV format, and the sampling rate to 16000.

```
def resample_rate(path,new_sample_rate):
    signal, sr = librosa.load(path, sr=None)
    wavfile = path.split('/')[-1]
    wavfile = wavfile.split('.')[0]
    file_name = wavfile + '_new.wav'
    new_signal = librosa.resample(signal, sr, new_sample_rate) #
    librosa.output.write_wav(file_name, new_signal , new_sample_rate)
    #命令行执行sox 1_new.wav -b 16 -e signed-integer 1_new.wav
```

Figure 9: Change Audio File Sampling Rate

6.2 Move Detailed Data Computing

According to the needs of users, we can get the distance, angle, speed and terminal position, and convert these data into the data that can be recognized and transmitted by serial communication for serial communication. The manipulator can actually move according to these data to meet the needs of users.

```

def inverse_kin(sx, sy, sz, s):
    x1 = float(sx)
    y = float(sy)
    z1 = float(sz)
    L13 = 84.72
    L2 = 41.04
    L5 = 200
    L6 = 214.8
    L7 = 45
    DX = 300
    DZ=50

    q1 = 17.657917590920615
    q2 = 152.68267192378207
    q3 = 0.0

    x = x1 + DX
    z=z1+DZ
    jj3 = math.atan2(y, x)
    a = (x ** 2 + y ** 2) ** 0.5 - L2 - L7
    b = z - L13
    c = (a ** 2 + b ** 2) ** 0.5
    sita2 = math.atan2(b, a)
    sita1 = math.acos((L5 ** 2 + c ** 2 - L6 ** 2) / (2 * L5 * c))
    jj2 = math.pi / 2 - sita1 - sita2
    sita3 = math.acos((L5 ** 2 + L6 ** 2 - c ** 2) / (2 * L5 * L6))
    jj1 = sita3 - jj2-math.atan2(28,72)
    j1d = jj1 * 180 / math.pi
    j2d = jj2 * 180 / math.pi
    j3d = jj3 * 180 / math.pi
    return -10 * (j1d * 45 / 11 - q1), -10 * (j2d * 45 / 11 - q2), 10 * (j3d * 45 / 21 - q3),s

```

Figure 10: Move Detailed Data Computing

6.3 User Interface Design

According to the requirements, we use pyqt to design a user interface, users can choose up, down, left, right, front and back six directions of movement and moving distance, can also select the location of the end point and moving speed to move.

6.4 Realization of Speech Recognition

Users can input voice commands, and the system can recognize voice and make corresponding operations.

6.5 Realization of Serial Communication

Serial communication is the communication function between the software system and the hardware part of the manipulator. The software function transfers a series of mobile data to the hardware, and the hardware makes corresponding mobile operations according to the data.

7 Preview Video

The system running preview video can be seen in the video folder of the submitted file.

References

- [1] Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th edition,Ben Shneiderman,Catherine Plaisant,Maxine Cohen

```
def up_clicked():
    print('up')
    t1,t2,t3,sp=inverse_kin(posx,posy,posz+18,speed)
    tri_move(t1,t2,t3,sp)
def down_clicked():
    print('down')
    t1,t2,t3,sp=inverse_kin(posx,posy,posz-18,speed)
    tri_move(t1,t2,t3,sp)
def left_clicked():
    print('left')
    t1,t2,t3,sp=inverse_kin(posx,posy-18,posz,speed)
    tri_move(t1,t2,t3,sp)
def right_clicked():
    print('right')
    t1,t2,t3,sp=inverse_kin(posx,posy+18,posz,speed)
    tri_move(t1,t2,t3,sp)
def behind_clicked():
    print('behind')
    t1,t2,t3,sp=inverse_kin(posx-18,posy,posz,speed)
    tri_move(t1,t2,t3,sp)
def front_clicked():
    print('front')
    t1,t2,t3,sp=inverse_kin(posx+18,posy,posz,speed)
    tri_move(t1,t2,t3,sp)
```

Figure 11: User Interface Design

```
def btn_mop_clicked():
    global posx,posy,posz
    startpos = [posx,posy,posz]
    #print(startpos )
    endpos = [float(ui.lineEdit_mp1.text()),\
              float(ui.lineEdit_mp2.text()),\
              float(ui.lineEdit_mp3.text())]
    tgts = []
    midpos = [(startpos[0]+endpos[0])/2,(startpos[1]+endpos[1])/2,posz+50]

    tgt = [startpos,midpos,endpos]
    tgts = []
    for row in tgt:
        tpr1 ,tpr2,tpr3 ,yyyy= inverse_kin(row[0],row[1],row[2],0)
        tgts.append([tpr1,tpr2,tpr3])

    #print("gggg = {}".format(tgts))
    for row in tgts:
        flag = 0
        tri_move2(row[0],row[1],row[2],ui.lineEdit_mp4.text())

        #th = Thread_3()
        #th.run(row)
```

Figure 12: User Interface Design

- [2] Li Hongcheng, Zhang Shuli, Hao Xin, Liu Shenghui. Intelligent control method of desktop manipulator [J]. Technological innovation and application, 2020 (05): 42-43
- [3] Baidu Speech Recognition Technology Document,<https://cloud.baidu.com/doc/SPEECH/index.html>

```
def do_sth(self, text):
    self.ui.movie.stop()
    self.ui.movie.jumpToFrame(0)
    self.ui.label_6.setText(text or '小思没听清能再说一遍嘛? ')
    print(text)
    try:
        for i in text:
            if '前' in i:
                front_clicked()
            elif '后' in i:
                behind_clicked()
            elif '上' in i:
                up_clicked()
            elif '下' in i:
                down_clicked()
            elif '左' in i:
                left_clicked()
            elif '右' in i:
                right_clicked()
        if "马驮" in text:
            btn_mop_clicked()
        if "写" in text and "一" in text:
            up_one()
        if "写" in text and "二" in text:
            up_two()
    except:
        self.ui.label_6.setText('小思没听清能再说一遍嘛? ')
    self.timer.setInterval(5000)
    self.timer.start()
```

Figure 13: Realization of Speech Recognition

```
def serial_send_data(data):
    if not serial0.isOpen():
        serial0.open()
    if serial0.isOpen():
        #print ('串口已打开')
        # data = b'ML(-1800-1800+1800,018)\r\n'      #发送的数据
        serial0.write(data)      #串口写数据
        #print ('You Send Data:',data)
        print('You Send Data:{}' .format(data.decode('utf8')))
        while True:
            data = serial0.read(40)      #串口读40位数据
            if data != b'':
                break
        print('receive data is :{}' .format(data.decode('utf8')) )
        #print ('receive data is :',data)
        data = data.decode('utf8')
        global posx,posy,posz
        posx,posy,posz = transpos_kin (data)
        str1 = "Position x={:.1f};y={:.1f};z={:.1f}" .format(posx,posy,posz)
        global flag
        flag = 1
        #print(flag)
    else:
        print('串口未打开')
        # ('串口未打开')

    #关闭串口
    serial0.close()

    if serial0.isOpen():
        print('串口未关闭')
        #print ('串口未关闭')
    else:
        pass
        #ui.textEdit_cons.append ('串口已关闭')
        #print ('串口已关闭')
```

Figure 14: Realization of Serial Communication