



华南理工大学
South China University of Technology

创新实践报告书

题目：智能种植机操作平台

学 院 自动化科学与工程学院

专 业 自动化

学生姓名刘纯富

学生学号201830611316

指导教师陈安

课程编号 046100401

课程学分 1

起始日期 2022. 4. 30——2022. 6. 24

| | |
|------|---------------------------------|
| 教师评语 | <div>教师签名：</div> <div>日期：</div> |
| 成绩评定 | |
| 备注 | |

目录

| | |
|---------------------------|----|
| 一、选题背景..... | 1 |
| 二、设备简介..... | 2 |
| 三、设计理念..... | 5 |
| 3.1 总体方案的选择 | 5 |
| 3.1.1 eye in hand 法 | 5 |
| 3.1.2 eye to hand 法 | 6 |
| 四、过程论述..... | 8 |
| 4.1 人员分工 | 8 |
| 4.2 单片机程序 | 9 |
| 4.2.1 主程序流程图 | 9 |
| 4.2.2 水泵控制 | 9 |
| 4.2.3 直流电机抓手控制 | 10 |
| 4.2.4 矩阵键盘控制 | 11 |
| 五、结果分析..... | 12 |
| 六、创新实践总结 | 13 |
| 6.1 心得体会 | 13 |
| 6.2 组长评分 | 13 |
| 附录..... | 14 |

智能种植机操作平台

一、选题背景

在科技迅速发展，模式识别和人工智能快速进步的社会背景下，如何将先进的科学技术融入到传统的劳动密集型产业中去，以提高生产力，带动社会发展，成为科研人员需要解决的问题。

而从历史角度看，农业种植业一直是人类生产生活的核心所在，如何改进生产工具和方法，提高生产力一直是人类在绞尽脑汁想解决的。本次智能种植机操作平台设计旨在实现全自动化和智能化的种植，让种植可控且更加精细化，解放人们的双手，提高生产的效率，降低人力成本。

在设计中我们将目标球当作农作物，以不同颜色的球表示不同的农作物。我们所想达到的目标是：对目标球和抓手进行识别定位并通过驱动 x 、 y 、 z 轴方向的电机（PPP 结构机械臂）来使抓手位移到目标球的位置，再通过驱动抓手和 z 轴电机对目标球进行拾取操作，同时可根据需求通过水泵实现液体的喷洒，从而模拟实际种植场景中种植物的转移和灌溉步骤。同时，除了实现全自动的对目标球进行拾取外，也能实现通过手动按键驱动电机和水泵对目标球进行拾取和灌溉功能。

智能种植机操作平台设计的指导思想为：摄像头和上位机程序对目标球和抓手进行精确的识别定位，上位机给单片机发送准确位置信号，单片机通过接收到的信号给步进电机和直流电机驱动器发送的脉冲信号进而准确地控制抓手的移动和拾取操作，当然，根据需求可控制水泵灌溉。

二、设备简介

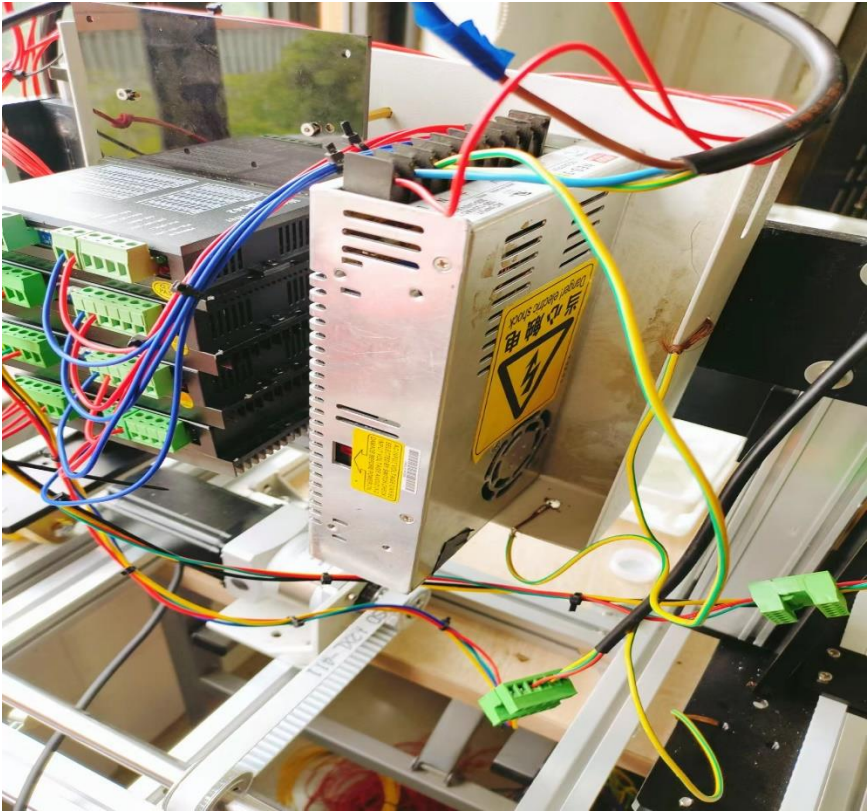


图 2-1

如图 2-1 为整流装置，可将 220v 交流电转换为 24v 的直流电，然后为各个模块供电。

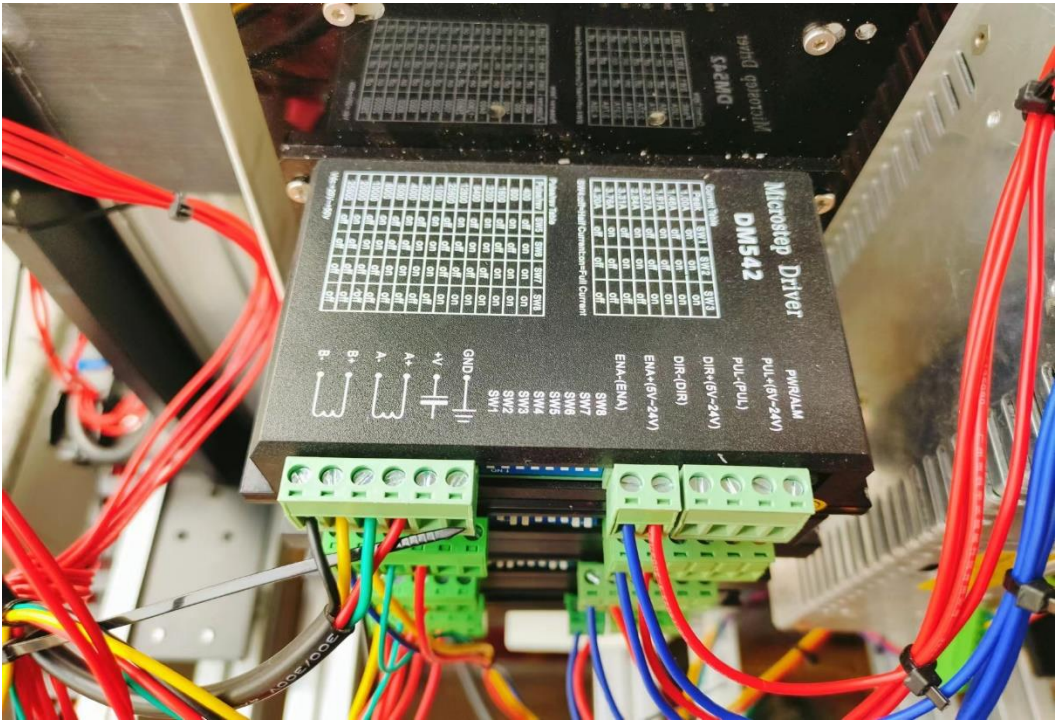


图 2-2

如图 2-2 为 DM542 数字式两相步进电机驱动器，目的是驱动和控制 x、y 和 z 轴的步进电机的运动。

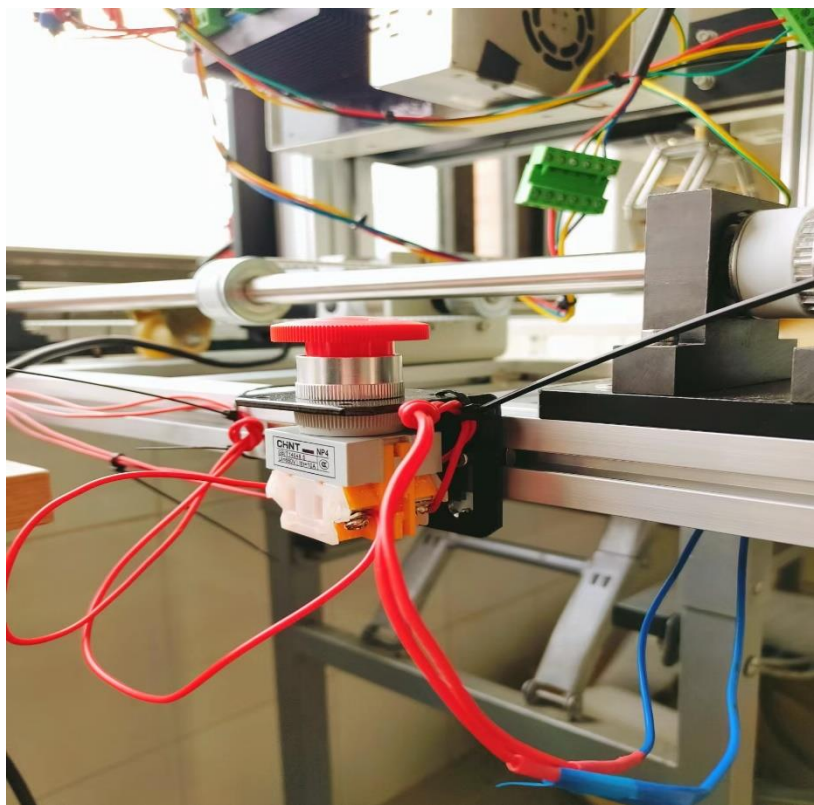


图 2-3

如图 2-3 红色按钮为设备的总电源开关。

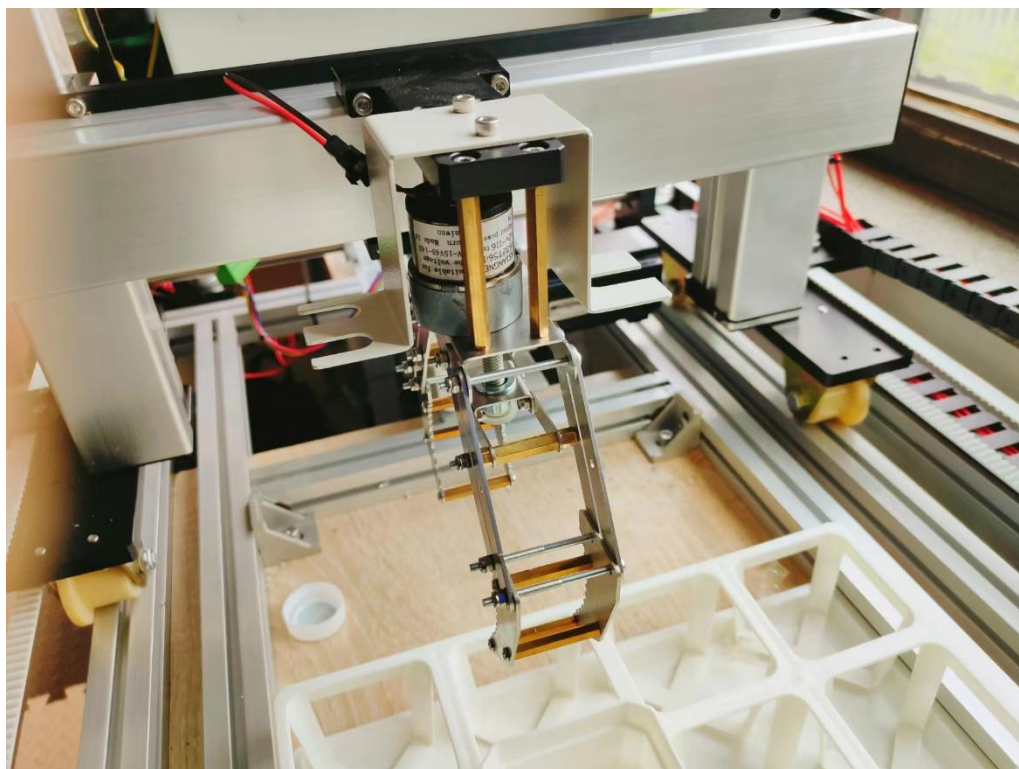


图 2-4

如图 2-4 为抓手，由直流电机驱动，直流电机正转，抓手夹紧，直流电机反转，抓手则松开。

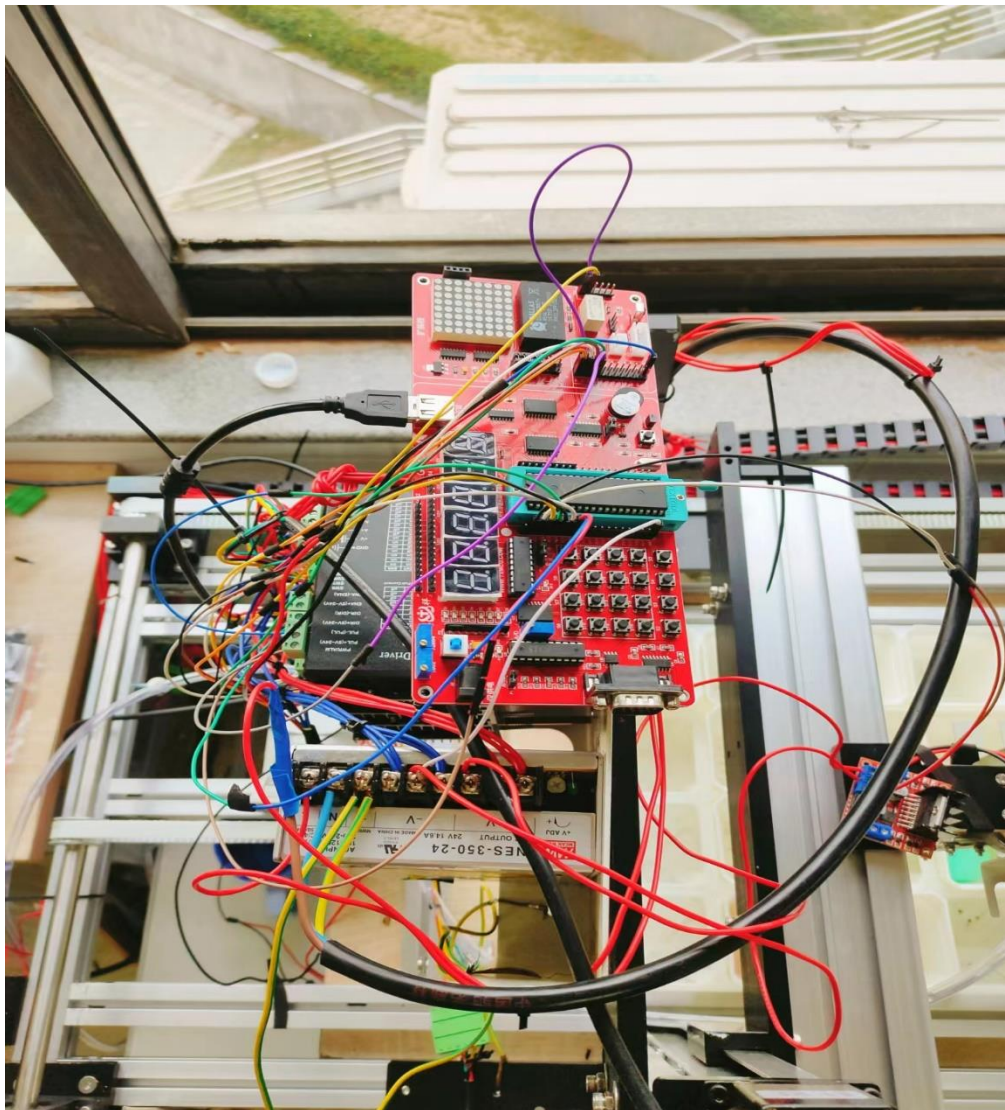


图 2-5

如图 2-5 为单片机，即下位机控制器，我们采用的是 51 单片机。



图 2-6

如图 2-6 位直流电机驱动模块，即是抓手驱动模块。

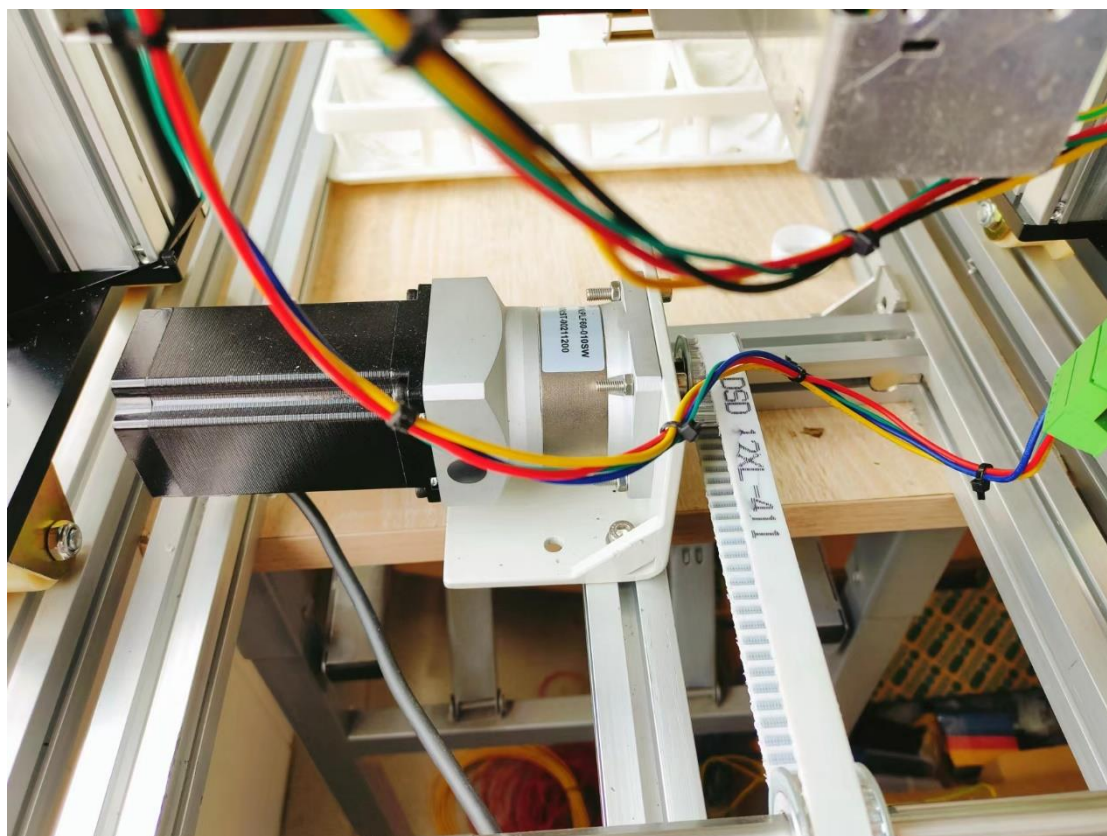


图 2-7

如图 2-7 为驱动 x 轴机械臂的步进电机，y 轴和 z 轴步进电机类似。

三、设计理念

3.1 总体方案的选择

3.1.1 eye in hand 法

这种方案是将摄像头安装在抓手上，x、y 轴机械臂带动抓手遍历种植区域的每一个位置，摄像头跟随抓手一起移动，摄像头拍到的图像为抓手正下方的图像，计算机（笔记本电脑）处理摄像头拍摄到的图像，图像中出现目标，则 z 轴机械臂下降一定高度（超声波测距），然后驱动抓手抓住目标，同时可驱动水泵喷水，然后 z 轴上升，完成对目标的抓取，然后通过 x、y 轴机械臂将目标移动到指定位置。

该方案的优点是上位机识别的工作任务少，仅需识别抓手下方是否为目标，然后将识别到目标的信号发送给单片机，驱动相应设备运动。缺点是整个

系统是开环控制，抓手的运动轨迹固定，不能适应目标的偏差，且下位机程序复杂。如图 3-1 为该方案的操作流程图。

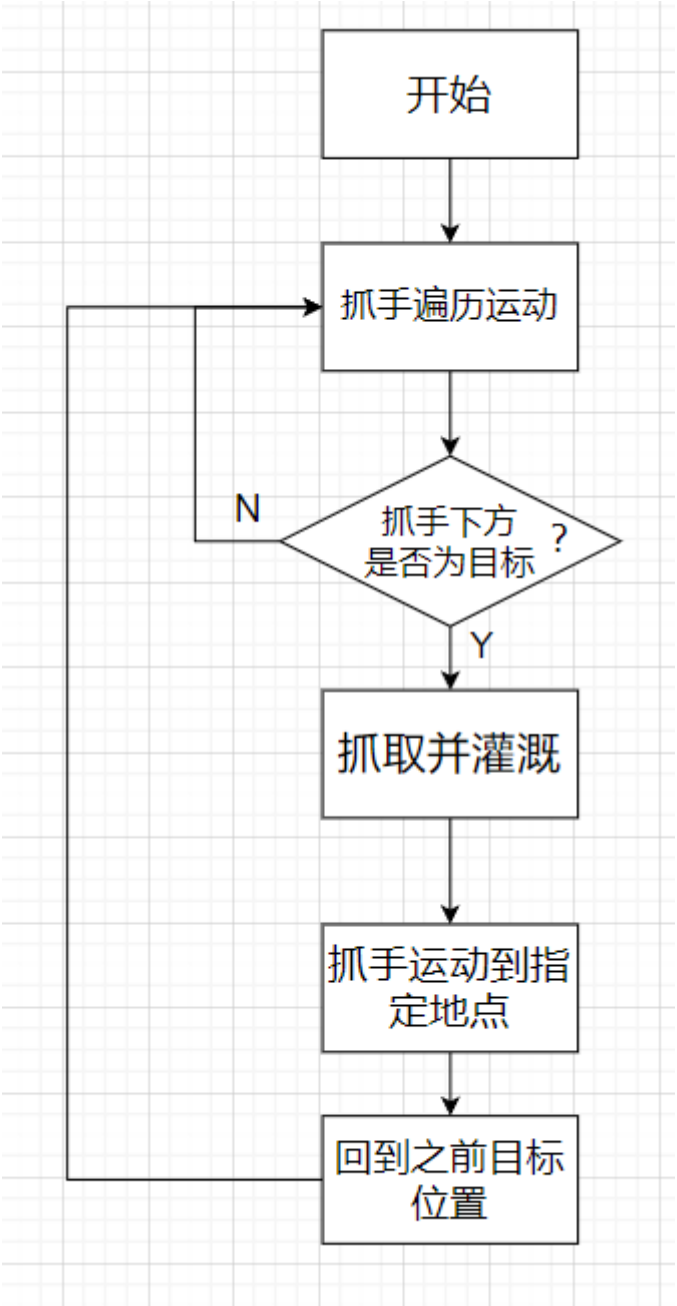


图 3-1

3.1.2 eye to hand 法

这种方案是将摄像头安放在整个种植区域的上方，摄像头拍到的图像为整个种植区域的图像，计算机（笔记本电脑）处理拍摄的图像，识别抓手和目标位置的偏差，只要有偏差，就给单片机发送 x、y 轴电机运动信号，单片机收到该信

号就驱动 x 、 y 轴电机运动，抓手和目标位置重合，计算机就发送 x 、 y 轴电机停止运动信号，单片机收到该信号， x 、 y 轴电机停止运动，同时单片机驱动 z 轴电机运动使 z 轴机械臂下降一定高度，然后驱动抓手抓住目标，然后 z 轴电机运动使 z 轴机械臂上升一定高度，完成对目标的抓取。

该方案的缺点是上位机识别程序会比较复杂，需要根据拍摄的图像判断是否有偏差同时发送电机运动信号。优点是抓手运动是根据抓手与目标之间的偏差运动的，具有偏差适应性，更加智能化，且下位机程序简单。如图 3-2 为该方案的操作流程图。

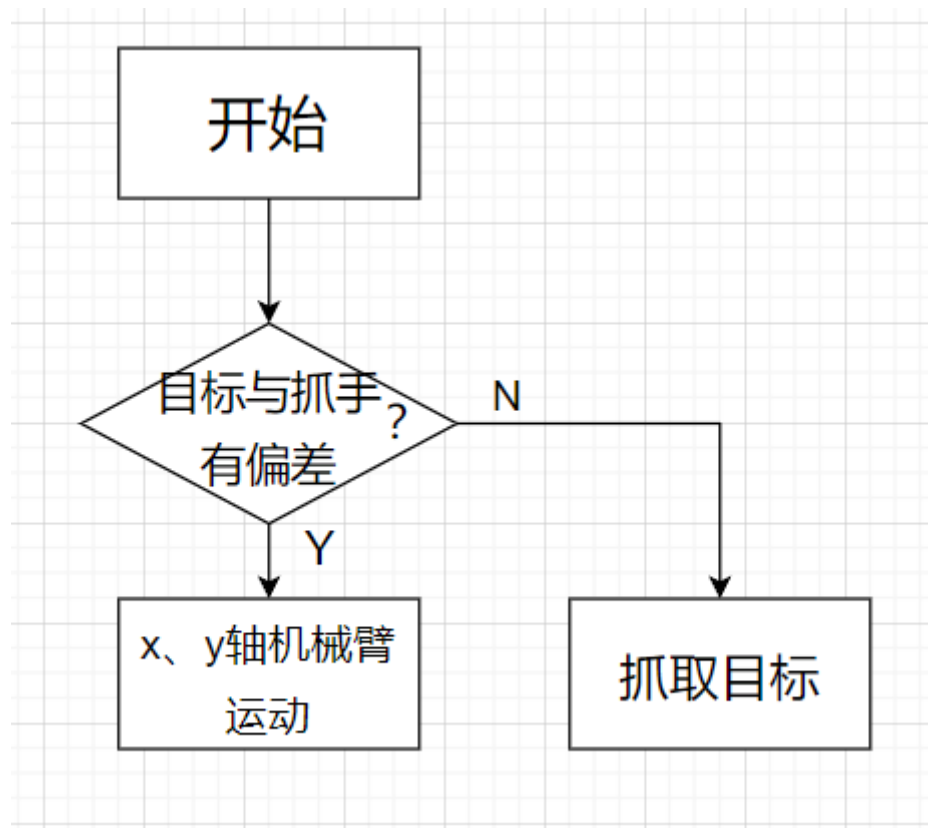


图 3-2

最终我们选择了 eye to hand 法。

四、过程论述

4.1 人员分工

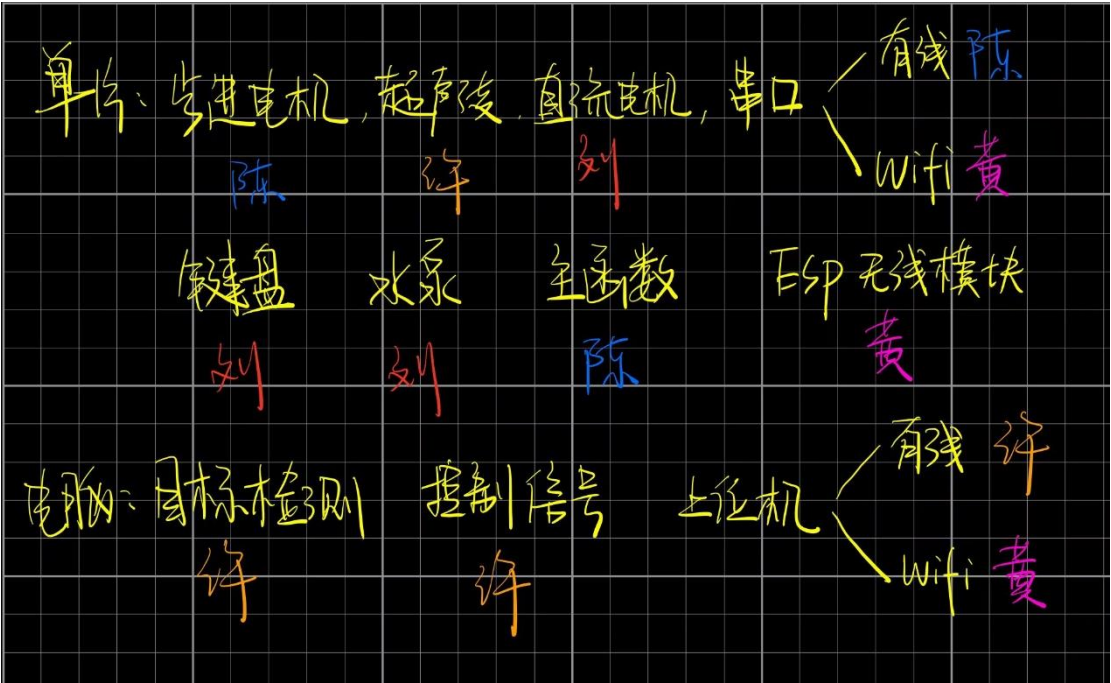


图 4-1

4.2 单片机程序

4.2.1 主程序流程图

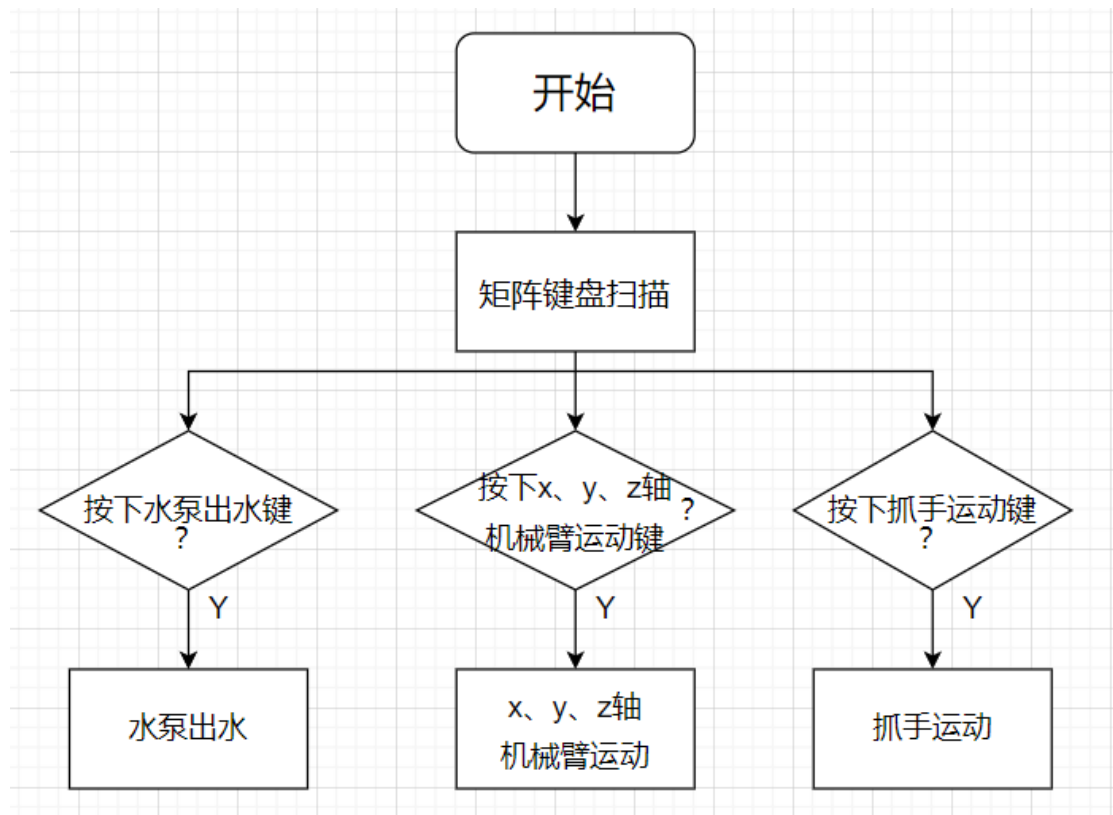


图 4-2

4.2.2 水泵控制

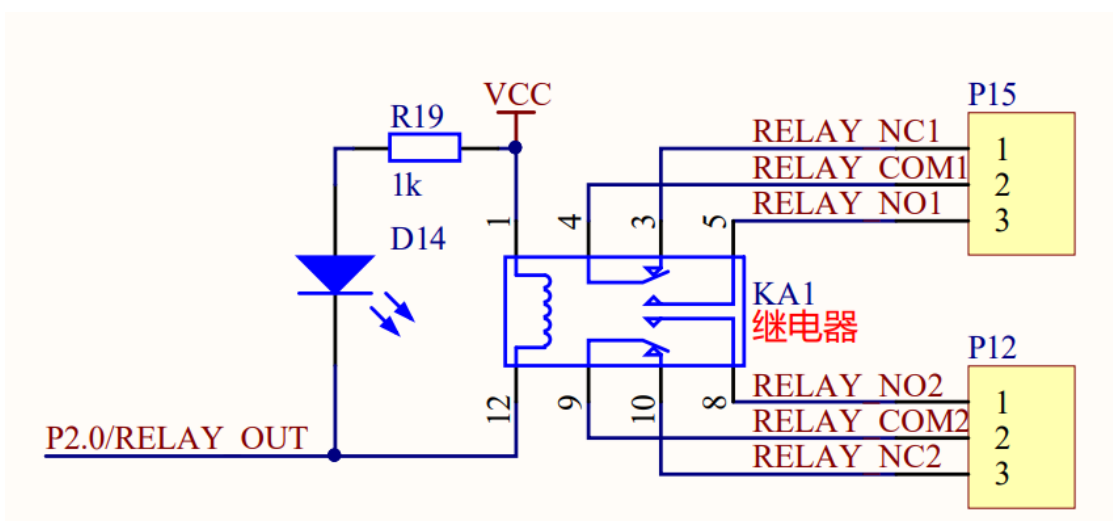


图 4-3

如图 4-3 为继电器原理图，继电器实际由三极管驱动，水泵直流电机回路由继电器的常开端和公共端控制通断，当图示 P2.0 口给低电平信号，常开端和公共端

闭合，水泵通电开始工作。

4.2.3 直流电机抓手控制

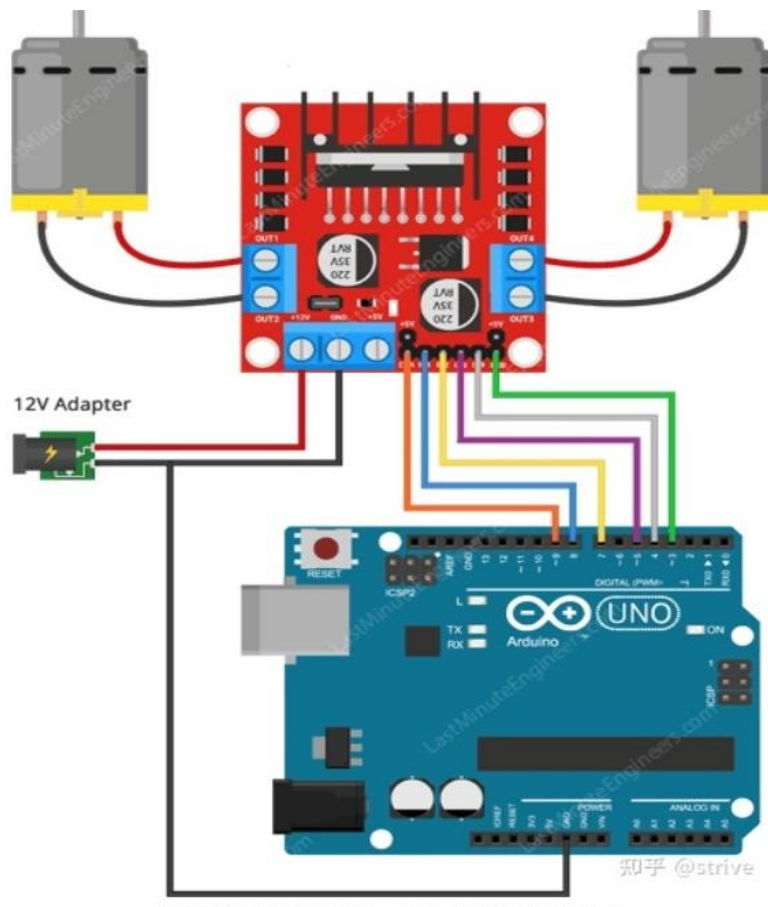


图 4-4

如图 4-4 为 L298N 电机驱动板 + 直流电机控制接线图，电机驱动板通过 H 桥控制电机旋转方向（前进、后退、停止），如图 4-5 为 H 桥电路图。单片机给驱动板对应的信号来控制直流电机运动。

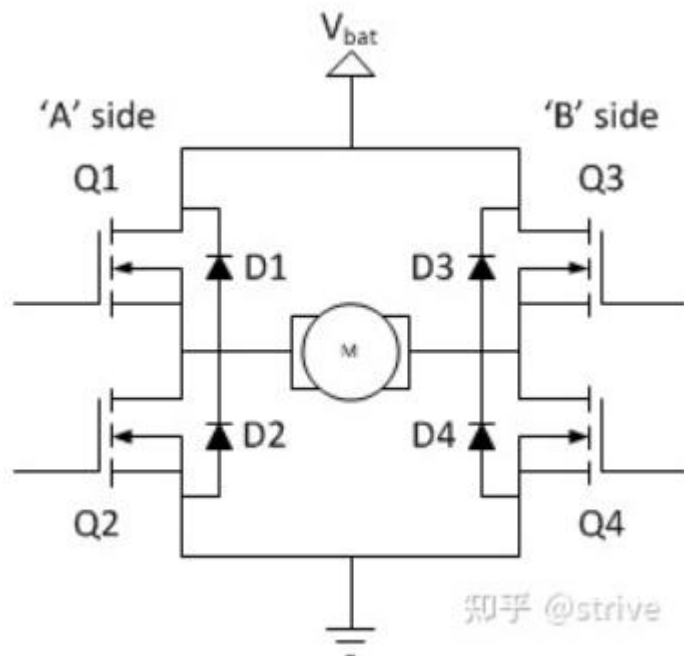


图 4-5

4. 2. 4 矩阵键盘控制

如图 4-2，编写矩阵键盘扫描程序，按下对应的按键，程序返回特定值，判断返回的值来运行对应程序，以此来达成对按键对设备的手动控制。

五、结果分析

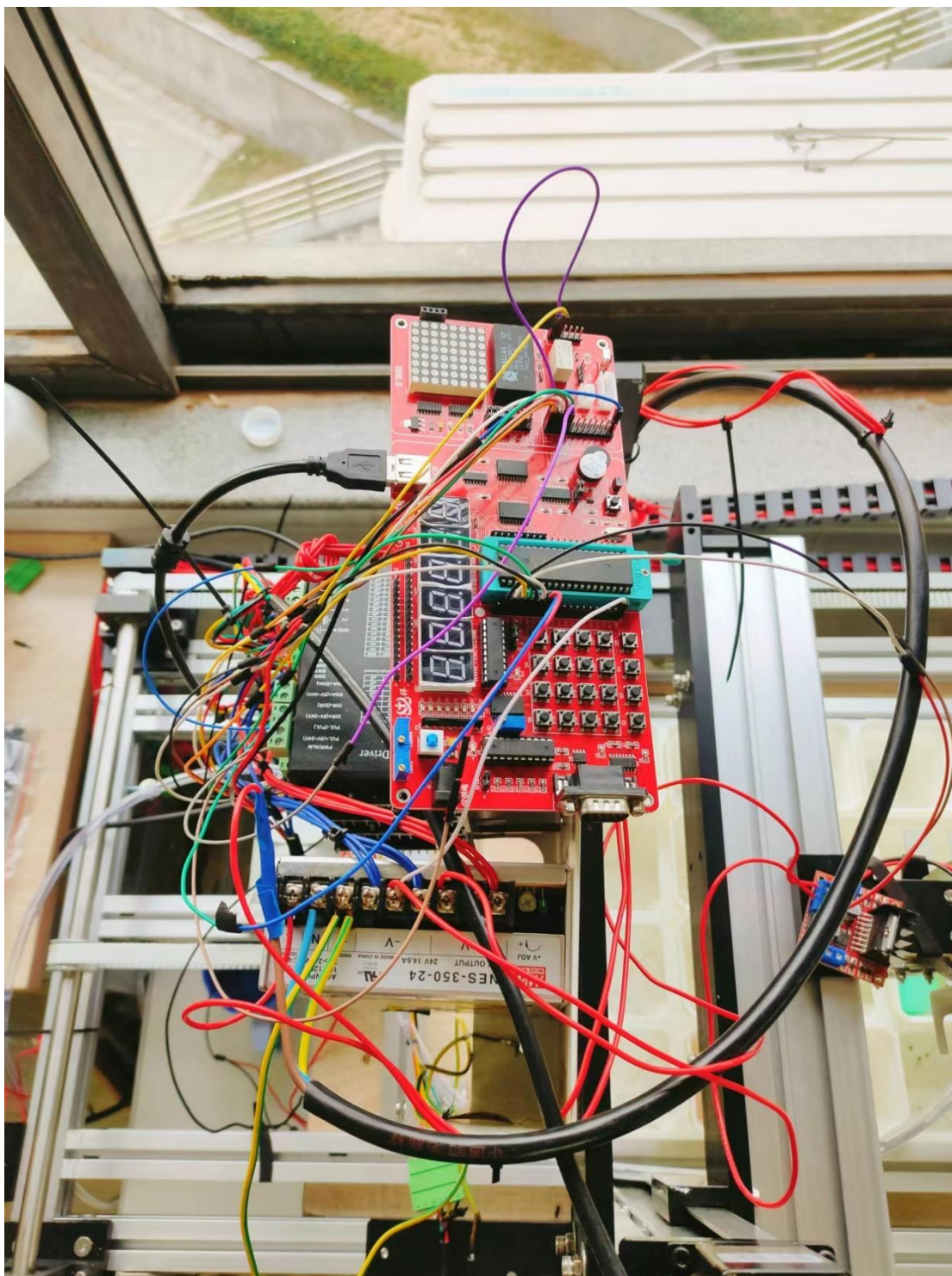


图 5-1

如图 5-1 为单片机和各设备接线图。

最终运行时，按键都能有效控制 x 、 y 、 z 轴机械臂，水泵和抓手，缺点是速度不够快和不能变速，比较好地完成了分工任务。

六、创新实践总结

6.1 心得体会

第一次去实验室看到设备时，设备上一堆连接线，看着还觉得挺复杂，仔细观察了解各个部件的功能后，发现整个种植机的工作原理还是比较简单的，关键还是程序的设计，让各个部件按照预想的方案协调工作非常重要。

在实践过程中，要将各科专业知识结合起来运用，比如机器人技术基础、单片机原理和电机拖动等。丰富和加强自己的专业知识对科学设计将大有裨益，在以后的学习中，应当针对性的学习各类知识。

种植机操作平台设计工作量还是比较大的，合理的分工合作才能让设计进展地快而好，像这种工作量偏大的设计，需要有持之以恒的心态。大家协商并合理地安排了各个小组成员的工作内容，这样我们的设计才能有序进行，并且在设计过程中组长和成员都十分认真负责，只有大家认真负责的态度，才能有好的成果。

经过此次的创新实践，我体会到了设计的艰辛，加深了对专业知识的实际运用能力，也体会到做出成果的喜悦。

6.2 组长评分

许文晋：100

附录

```
#include<reg52.h>
#define uchar unsigned char
#define uint unsigned int
sbit relay=P2^0;    //继电器控制口
sbit in1=P1^0;      //抓手控制脉冲输入
sbit in2=P1^1;      //同上
sbit pulse_x=P1^2;  //x dirver
sbit dir_x=P1^3;    //同上
sbit pulse_y=P1^4;  //y dirver
sbit dir_y=P1^5;    //同上
sbit pulse_z=P1^6;  //z dirver
sbit dir_z=P1^7;    //同上
sbit dula=P2^6;     //数码管锁存端
sbit wela=P2^7;     //数码管锁存端
uchar flag,a;
uchar code table[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,
                    0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};
void delayms(uint xms); //长延时
void delay(uint x);     //短延时
uchar matrixkeyscan(); //按键控制
void pump();            //水泵控制
void grab();            //抓手控制
void init();            //串口通信初始化
void manipulator();     //ppp 结构操作臂控制
void main()
{
    P0=0;
    dula=1;
    dula=0;
    P0=0xc0;
    wela=1;
    wela=0;
```

```

    init();
    while(1)
    {
        manipulator();
        pump();
        grab();
    }
}

void delayms(uint xms)
{
    uint i, j;
    for(i=xms; i>0; i--)
        for(j=110; j>0; j--);
}

void delay(uint x)
{
    uint i, j;
    for(i=x; i>0; i--)
        for(j=10; j>0; j--);
}

void display(uchar num)
{
    P0=table[num];
    dula=1;
    dula=0;
}

uchar matrixkeyscan()
{
    uchar temp, key;
    P3=0xfe;
    temp=P3;
    temp=temp&0xf0;
    if(temp!=0xf0)
    {

```

```

    delayms(10);
    temp=P3;
    temp=temp&0xf0;
    if(temp!=0xf0)
    {
        temp=P3;
        switch(temp)
        {
            case 0xee:
                //key=0;
                break;

            case 0xde:
                key=1;
                break;

            case 0xbe:
                //key=2;
                break;

            case 0x7e:
                //key=3;
                key=7;
                break;
        }
        while(temp!=0xf0)
        {
            temp=P3;
            temp=temp&0xf0;
        }
        display(key);
    }
}
P3=0xfd;

```

```

temp=P3;
temp=temp&0xf0;
if(temp!=0xf0)
{
    delays(10);
    temp=P3;
    temp=temp&0xf0;
    if(temp!=0xf0)
    {
        temp=P3;
        switch(temp)
        {
            case 0xed:
                //key=4;
                key=3;
                break;

            case 0xdd:
                //key=5;
                key=2;
                break;

            case 0xbd:
                //key=6;
                key=4;
                break;

            case 0x7d:
                //key=7;
                key=8;
                break;
        }
        while(temp!=0xf0)
        {

```



```

        temp=P3;
        temp=temp&0xf0;
    }
    display(key);
}
}
P3=0xfb;
temp=P3;
temp=temp&0xf0;
if(temp!=0xf0)
{
    delayms(10);
    temp=P3;
    temp=temp&0xf0;
    if(temp!=0xf0)
    {
        temp=P3;
        switch(temp)
        {
            case 0xeb:
                //key=8;
                break;

            case 0xdb:
                //key=9;
                key=5;
                break;

            case 0xbb:
                //key=10;
                break;

            case 0x7b:
                //key=11;

```

```

    key=9;
    break;
}
while(temp!=0xf0)
{
    temp=P3;
    temp=temp&0xf0;
}
display(key);
}
}
P3=0xf7;
temp=P3;
temp=temp&0xf0;
if(temp!=0xf0)
{
    delays(10);
    temp=P3;
    temp=temp&0xf0;
    if(temp!=0xf0)
    {
        temp=P3;
        switch(temp)
        {
            case 0xe7:
                //key=12;
                break;

            case 0xd7:
                //key=13;
                key=6;
                break;

            case 0xb7:

```

```

        //key=14;
        break;

        case 0x77:
            //key=15;
            key=10;
            break;
    }
    while(temp!=0xf0)
    {
        temp=P3;
        temp=temp&0xf0;
    }
    display(key);
}
}
return key;
}
void pump()
{
    if(matrixkeyscan()==10)
    {
        delayms(1);
        if(matrixkeyscan()==10)
        {
            relay=0;
        }
    }
    if(matrixkeyscan()!=10)
    {
        delayms(1);
        if(matrixkeyscan()!=10)
        {
            relay=1;

```

```

    }
}
}
void grab()
{
    if (matrixkeyscan()==7)
    {
        delayms(10);
        if (matrixkeyscan()==7)
        {
            in1=1, in2=0;
        }
    }
    if (matrixkeyscan()==8)
    {
        delayms(10);
        if (matrixkeyscan()==8)
        {
            in1=0, in2=1;
        }
    }
    if (matrixkeyscan()==9)
    {
        delayms(10);
        if (matrixkeyscan()==9)
        {
            in1=0, in2=0;
        }
    }
}
void init()
{
    TMOD=0x20;
    TH1=0xfd;

```

```

    TL1=0xfd;
    TR1=1;
    REN=1;
    SM0=0;
    SM1=1;
    EA=1;
    ES=1;
}

void manipulator()
{
    if(matrixkeyscan()==1)//x 轴正转
    {
        delayms(1);
        if(matrixkeyscan()==1)
        {
            dir_x = 1;
            while(matrixkeyscan()==1)
            {
                delay(1);
                pulse_x=0;
                delay(1);
                pulse_x=1;
                delay(1);
            }
        }
    }
    if(matrixkeyscan()==2)//x 轴反转
    {
        delayms(1);
        if(matrixkeyscan()==2)
        {
            dir_x = 0;
            while(matrixkeyscan()==2)
            {

```

```

        delay(1);
        pulse_x=0;
        delay(1);
        pulse_x=1;
        delay(1);
    }
}
}
if(matrixkeyscan()==3)//y 轴正转
{
    delayms(1);
    if(matrixkeyscan()==3)
    {
        dir_y = 1;
        while(matrixkeyscan()==3)
        {
            delay(1);
            pulse_y=0;
            delay(1);
            pulse_y=1;
            delay(1);
        }
    }
}
if(matrixkeyscan()==4)//y 轴反转
{
    delayms(1);
    if(matrixkeyscan()==4)
    {
        dir_y = 0;
        while(matrixkeyscan()==4)
        {
            delay(1);
            pulse_y=0;

```



```

        delay(1);
        pulse_y=1;
        delay(1);
    }
}
}
if(matrixkeyscan()==5)//z 轴正转
{
    delayms(1);
    if(matrixkeyscan()==5)
    {
        dir_z = 1;
        while(matrixkeyscan()==5)
        {
            delay(1);
            pulse_z=0;
            delay(1);
            pulse_z=1;
            delay(1);
        }
    }
}
if(matrixkeyscan()==6)//z 轴反转
{
    delayms(1);
    if(matrixkeyscan()==6)
    {
        dir_z = 0;
        while(matrixkeyscan()==6)
        {
            delay(1);
            pulse_z=0;
            delay(1);
            pulse_z=1;

```

```
        delay(1);  
    }  
}  
}  
}
```