



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 黄美芷

学生学号 201930361029

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

题目：智能种植机操作平台	1
一、选题背景	2
二、方案论证	2
2.1 总体方案的选择	2
2.1.1 eye in hand 法	2
2.1.2 eye to hand 法	2
2.2 无线通讯方案的选择	3
2.2.1 Wifi 通讯	3
2.2.2 蓝牙通讯	4
三、过程论述	5
3.1 esp8266 的应用调试	5
3.1.1 esp8266 为 server-上位机为 client	5
3.1.2 esp8266 与上位机透传	6
3.2 esp8266 与单片机的串口通讯	8
3.2.1 硬件连接	8
3.2.2 串口通讯程序代码	8
3.2.3 串口通讯流程图	10
3.3 上位机的无线通讯	11
3.3.1 通过 TCP 协议进行无线通讯	11
3.3.2 编程模型	11
3.3.3 上位机的无线通讯相关函数	12
四、结果分析	13
4.1 esp8266 配置结果	13
4.2 esp8266 与单片机配置	13
4.3 上位机通讯结果	14
4.4 种植机实现结果	15
五、创新实践总结	17

智能种植机操作平台

一、选题背景

我们的课题组直击社会和历史问题的痛点，将机器识别和传统的第一产业农业和种植业相融合，实现全自动化和智能化的种植，解放人们的双手，提高生产的效率，降低人力成本。

预期目标：对目标球和操作臂识别定位并驱动 x , y , z 方向的电机使操作臂位移到目标球的位置，再通过驱动操作臂和 z 轴电机对目标球进行拾取操作，从而模拟实际种植场景中种植转移步骤。同时，除了实现以上全自动的对目标球进行拾取外，也能实现通过手动按键对目标球进行拾取功能。

指导思想为：摄像头和上位机程序对目标球和操作臂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操作臂的移动和进行拾取操作。

二、方案论证

2.1 总体方案的选择

2.1.1 eye in hand 法

该法的操作步骤为：首先，操作臂遍历种植区域的网格，此时摄像头安装在操作臂上，若上位机识别出目标球，则给单片机发出停止指令，单片机收到停止指令，对 z 轴的步进电机发脉冲信号，让操作臂下降、夹其、上升，最终完成操作。

缺点为它是开环控制，当出现扰动时不好控制，而操作臂必须从原点开始遍历，同时设计遍历程序比较复杂，需要测量每移动一格的具体脉冲拍数。优点为上位机的识别程序简单，只需要识别各种球的种类，不需要物体定位，也不需要操作臂识别。

2.1.2 eye to hand 法

该法的操作步骤为：首先，摄像头架在种植区域的正上方较高的位置。通过上位机程序对操作臂和不同颜色的球进行识别，上位机程序通过确认需要拾取的球的位置，与操作臂的位置进行比较，进而通过串口程序给单片机发送信号，单片机通过上位机给他发送的信号决定发给 x , y 轴步进电机的信号，从而使得操作臂逐渐达到目标球位置进而实现拾取或者浇灌操作。

缺点为上位机程序较为复杂，需要用到 YOLOv5 识别算法，同时需要将操作臂纳入识别对象中。此外，远处的目标不好识别，只要当摄像头、操作臂和目标球三点共线上位机就会认为操作臂在目标球正上方，导致操作臂的夹取出现误差。

优点为它是闭环控制，上位机实时监控目标球和操作臂的相对位置，即使出现一些扰动也能迅速进行补偿，系统的稳定性较好，同时操作臂初始位置可以在不遮挡目标球的任何位置。因此本项目采用方案二。

2.2 无线通讯方案的选择

考虑农作物实际应用，控制决策中心和种植机操作执行机构往往分离，因此我们利用无线模块，上位机将控制信息传输到单片机中，单片机串口通讯接收信号并执行相应的操作。

2.2.1 Wifi 通讯

wifi 通讯主要分软硬件两个方面。软件方面是用单片机 c 语言知识建立 wifi 模块和单片机地连接，使得单片机能够接收发信息；此外上位机通过 python 语言也可以与 wifi 模块通讯。

在硬件方面，选择高性能、高性价比、开发环境便利的 ESP8266-01S 模块。并将 esp8266 与单片机串口硬件连接，搭建能够无线通讯的单片机系统。

在本项目中 wifi 通讯考虑了以下两种情况。

方案一：上位机作为客户端，搭载 esp8266 的单片机作为服务器。上位机连接 esp8266 模块发出的局域网进行通讯。连接模式有 TCP 和 UDP 连模。TCP 是面向连接的协议，在收发数据前必须和对方建立可靠的连接，为数据传输打下可靠基础；UDP 是一个面向无连接的协议，数据传输前，源端和终端不建立连接，发送端尽可能快的将数据扔到网络上，接收端从消息队列中读取消息段。

此时，对于单片机模块，需要实现串口调式软件的功能。先用 AT 指令集配置 esp8266 模块为 AP 模式并设置 esp8266 的 WiFi 基本信息。接着 esp8266 连接单片机，编写程序使得 esp8266 与单片机进行串口通讯。

对于上位机模块，需要实现网络调式助手的功能。通过 python 函数来实现上位机的 TCP 客户端方式的连接或者是 UDP 方式的连接。这两种协议连接方式的比较将在 3.3 节阐述。

方案二: 上位机和单片机作为客户端, 连接同一局域网进行通讯。连接模式为 UDP 连模。

此时, 手机设置热点。配置 esp8266 和上位机连接该热点, 为了使得下次上电 esp8266 不需要重新配置, 在该方案中设置为透传模式, 即下次上电 esp8266 接收的串口数据将不改变数据形式全部发送到单片机串口。上述两方案的分析比较将在 3.1 节阐述。

Wifi 通讯优点为: 为局域网通信, 传输速度快, 设备连接能力强。

2.2.2 蓝牙通讯

设备间通信时, 蓝牙功耗低、性价比高、易于开发, 可以很好搭载单片机使用, 被广泛使用在智能家居、远程控制等领域。其中, 最为常见的是选用 HC-05 蓝牙模块与单片机进行通讯。但由于蓝牙模块的软硬件连接方式和 wifi 模块几乎一致, 因此我们选择连接更加稳定, 传输速率更快的 wifi 模块。

三、过程论述

对于本组的项目，本人主要负责无线通讯部分，其中包括无线通讯模块 esp8266 的 wifi 配置，esp8266 与单片机的串口通讯，上位机的无线通讯三个部分。

3.1 esp8266 的应用调试

esp8266 wifi 调试装置图如下，将 esp8266 与 USB 转 TTL 连接，使用安信可串口调试助手进行 esp8266 的调试。

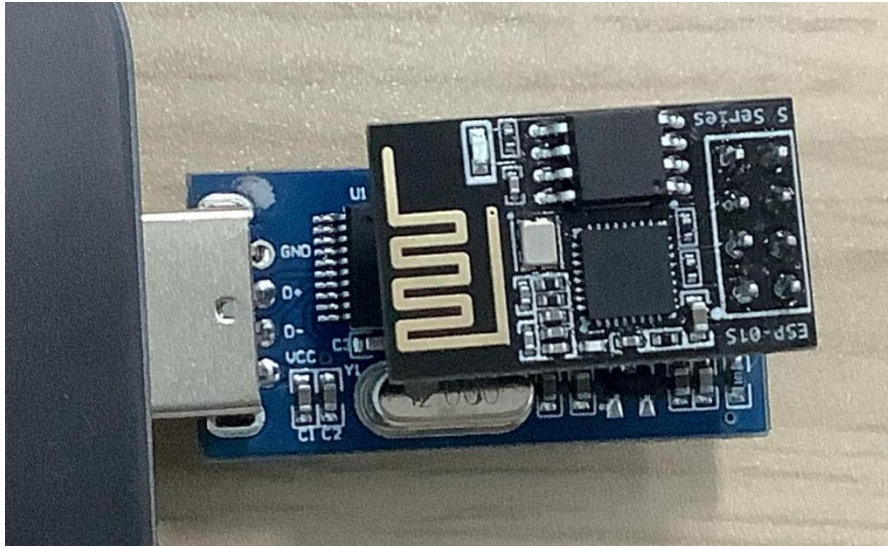


图 3-1 esp8266 wifi 配置装置图

3.1.1 esp8266 为 server-上位机为 client

1) 设置 wifi 模式

AT+CWMODE=2 //此时 8266 为 AP 模式

响应: OK

2) 重启生效

AT+RST

响应: ready

3) 设置局域网

AT+CWSAP="ESP8266","123456789",11,0 //热点名字、密码、通道 11 通道与其他的互不干扰，0 不需要密码接入

响应: OK

4) 查询设备 IP

AT+CIFSR //建立或连接一个网络，分配一个 IP，重新连接新的网络，IP 会改变，

电脑端也是如此

响应: +CIFSR:APIP,"192.168.4.1"

OK

其中该模块 IP 为 192.168.4.1。

5) 启动多连接并建立 server

AT+CIPMUX=1 //多链接

AT+CIPSERVER=1,8080//开启服务器模式,端口号 8080

响应: OK



图 3-2 esp8266 作为服务器调试图

重新上电，只有 wifi 信息保留。

再次上电，需要重新配置连接方式以及是否监听，所以重新上电后，8266 需要：

AT+CIPMUX=1

AT+CIPSERVER=1

3.1.2 esp8266 与上位机透传

1) 设置 wifi 模式及重启生效

AT+CWMODE=1 //此时 8266 为 station 模式

AT+RST

2) 加入设置好的 wifi

AT+CWJAP="test_not_online","1234567890HMZ"

响应: WIFI CONNECTED

3) 查询 IP

AT+CIFSR

响应: +CIFSR:STAIP,"192.168.43.206"

OK

4) 单连接及设置透传

AT+CIPMUX=0 //单连接

AT+CIPMODE=1 //开启透传

AT+SAVETRANSLINK=1,"192.168.43.12",8089,"TCP",2000 2000

//其中, 为 8266 的随便一个端口, IP 为电脑地址, 上电直接传输, 不用发 AT+CIPSEND

响应: OK

5) 重启生效

AT+RST

响应: ready



图 3-2 esp8266 作为客户端透传调试图

重新上电, 模块依旧保持着透传模式。

虽然 esp8266 透明传输模式对于单片机的编程而言更为简洁, 但无法判断该数据是否所需, 在调式阶段发现数据传输错误率较高。esp8266 作为服务器的无线通讯中, 单片机重新上电, 上位机则需要重新连接该局域网, 但一旦连接, 数据传输比较可靠和稳定, 因此, 本项目最终采取方案一。

3.2 esp8266 与单片机的串口通讯

3.2.1 硬件连接

单片机与 esp8266 的接口连接如图 3-3，实际连接效果如图 3-4。

ESP8266-01	TTL-USB
TXD	RXD
RXD	TXD
CH_PD、VCC	3.3V或5V (会很烫，容易烧坏)
GND	GND

图 3-3 硬件连接示意图

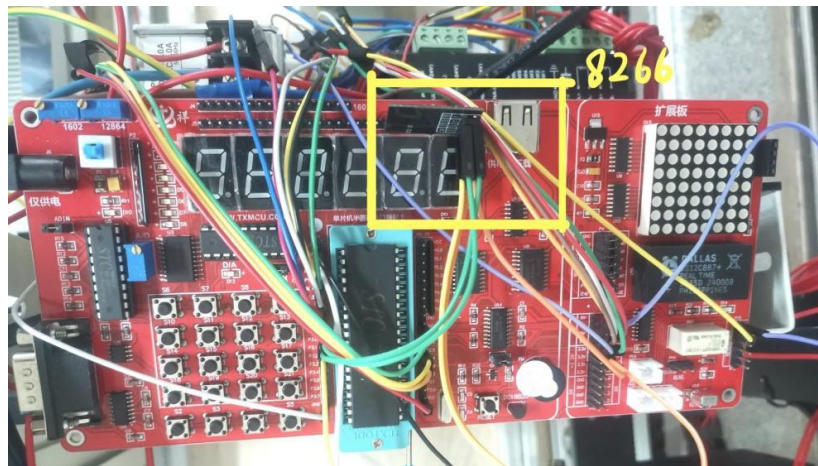


图 3-4 硬件连接实物图

如图，esp8266 共有 5 根线，TXD 和 RXD 分别连接单片机的 P3.0 (RXD)、P3.1(TXD)接口，EN、3.3V 接口连接右边的 3.3V 电源，GND 接口为地线。

3.2.2 串口通讯程序代码

- 1) 串口初始化函数同无线串口初始化一致，在此不加赘述。
- 2) 重新上电后单片机向 esp8266 发送 AT 指令进行配置，其中 AT 指令发送函数如下：

```
void SENT_At(uchar *At_Comd)//指针指向 At 指令
{
    ES = 0;//关闭串口中断
    while(*At_Comd!='\0')
    {
        SBUF = *At_Comd;
        while(!TI);//等待该字节发送完毕，发送完后硬件自动置一
        TI = 0;//硬件置一以后必须软件置零才能进行下一数据传送
        delay_us(5);
        At_Comd++;//指向下一个字节
    }
}
```

```

    }
}

```

3) 单片机对 wifi 模块初始化函数:

```

void WIFI_Init()//通过单片机配置 AT 指令
{
    SENT_At("AT+CIPMUX=1\r\n");
    delayms(1000);
    beeper=0;//蜂鸣器响
    SENT_At("AT+CIPSERVER=1,8080\r\n");
    delayms(1000);
    beeper=1;
    smgdisplay(0,0);//设置成功,蜂鸣器灭,数码管闪烁
    ES = 1;
}

```

4) 当上位机发送 0X30 时,在串口调式助手窗口接收的数据为: +IPD,0,1:0, 因此我们所需要的数据是进入缓存器的第 10 位,前面 4 位可以作为判断数据格式的标志,其中断函数如下。

```

void ser() interrupt 4
{
    if(RI==1)//硬件置一
    {
        RI = 0;//软件清零,防止未接收完成就执行程序
        Receive_table[i] = SBUF;
        if(Receive_table[0]!='+')//判断接收数据格式
            i++;
        else
            i = 0;
        if(i>=10)
        {
            if((Receive_table[0]!='+')&&(Receive_table[1]!='I')&&(Receive_table[2]!='P')
            &&(Receive_table[3]!='D'))
            {
                a=Receive_table[9];//取第 10 位的数据
                smgdisplay(a,1);//数码管显示函数
                delayms(1000);
            }
            i = 0;
        }
    }
    else
        TI = 0;
}

```

3.2.3 串口通讯流程图

主函数流程图:

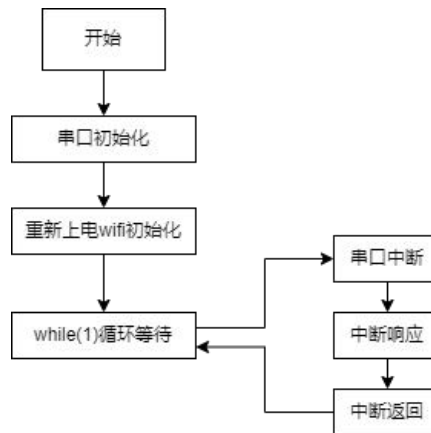


图 3-5 主函数流程图

中断函数流程图:

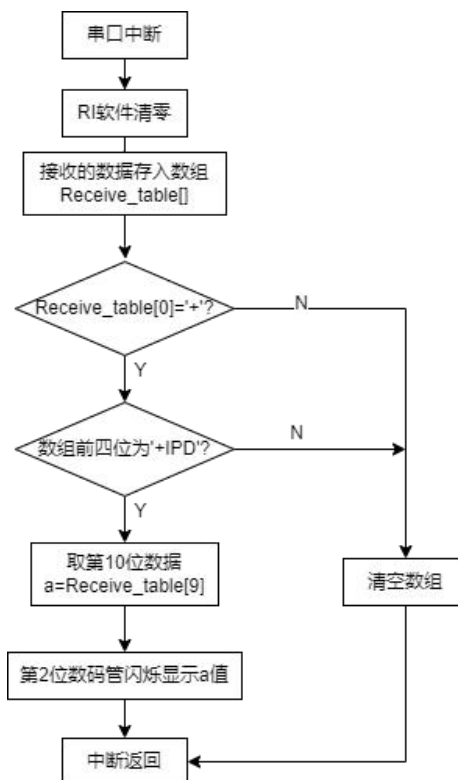


图 3-6 中断函数流程图

3.3 上位机的无线通讯

3.3.1 通过 TCP 协议进行无线通讯

TCP 与 UDP 都是 TCP/IP 协议栈的网络传输层协议。TCP 面向连接的协议，高可靠性为应用层提供可靠的、面向连接的数据服务。而 UDP 与 TCP 协议完全相反，它为应用层提供不可靠、无连接和基于数据报文的服务。“不可靠”意味着 UDP 协议无法保证数据从发送端正确地传送到目的端。在本次项目中，传输的数据是单片机的控制信号，为了减少数据传输过程的不可控因素，本项目采用 TCP 的可靠连接方式。

对于上位机的无线通讯，要解决两端的问题。一是服务器端，二是客户端，即是客户端的上位机和服务端的单片机。并用（IP 地址；端口号）的嵌接字形式表达通信端点。

服务器端，服务器先建立并绑定嵌接字，开启监听状态并允许连接请求，接下来则等待客户端的连接。在本次项目中，esp8266 则为服务器模式，已经先配置了局域网绑定了嵌接字，每次上电则是重新开启监听和允许连接状态。

客户端上位机作为客户端，创建嵌接字并申请与客户端连接，连接成功后用读写函数进行数据的通讯。当通讯完毕，结束连接。

3.3.2 编程模型

根据 3.3.1 节论述，可以得到下面的 TCP 套接字编程模型。

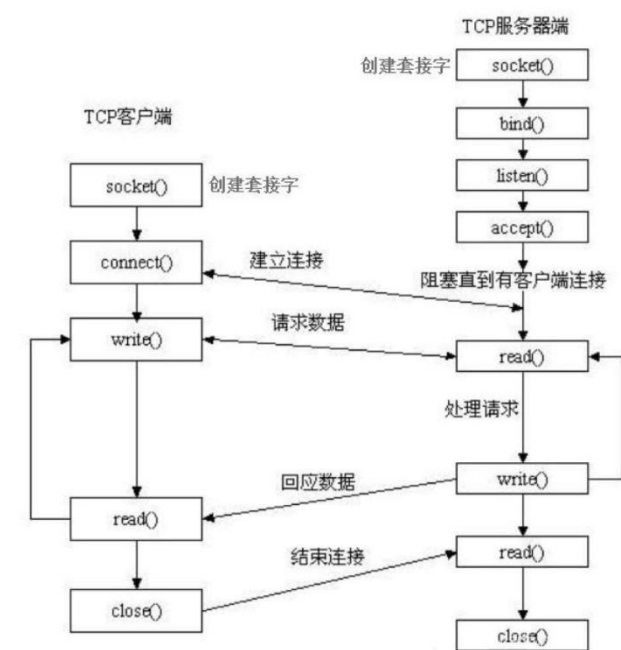


图 3-7 TCP 套接字编程模型

3.3.3 上位机的无线通讯相关函数

1) 建立连接

利用 socket 库，建立一个 Socket 对象，拥有发送和接受网络数据的功能并使用 TCP 协议进行可靠的连接，其 python 函数如下。

```
TCP_client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# socket 此时是一个 Socket 对象
# AF_INET (IPv4 协议用于 Internet 进程间通信), AF_UNIX (用于同一台机器进程间通信)
# 套接字类型: 可以是 SOCK_STREAM (流式套接字, 用于 TCP 协议)
print ('looking for server')
try:
    TCP_client.connect(("192.168.4.1", 8080))
    #这里的 IP 是 8266 作为热点后的 IP 地址
    print ('done.')
except:
    print('server not find')
```

2) 发送函数

本次项目中，和队长交流得知上位机根据 yolov5 算法获得一个包含目标类别和控制操作臂方向的信号，上位机只需要发送一个字节的信号即可，因此设计的数据类型为字节，其发送函数如下。

```
try :
    send_code=0x30
    #以 0x30 为测试数据
    if send_code=="exit":
        break
    TCP_client.send(struct.pack('B',send_code))
    time.sleep(0.2)
    print ('Message send successfully')
except socket.error:
    #Send failed
    print ('Send failed')
```

四、结果分析

4.1 esp8266 配置结果

如图，esp8266 的配置信息为：

名称: ESP8266

密码: 123456789

安全类型: 无密码直接连接

WiFi 通道: 11

并且进行了多连接配置和允许监听功能。



图 4-1 esp8266 配置结果

4.2 esp8266 与单片机配置

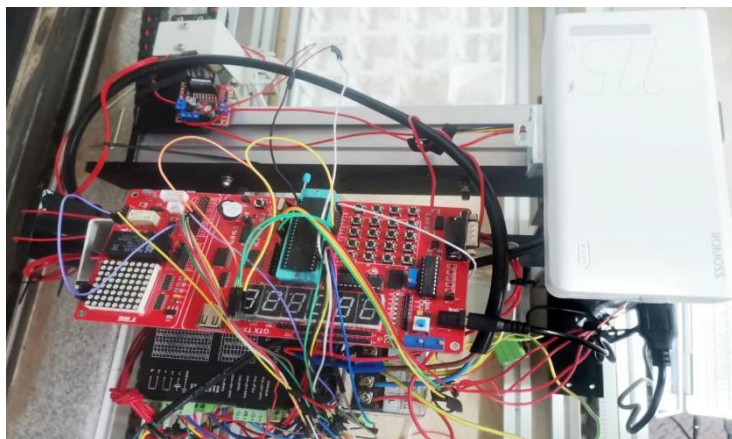


图 4-2 硬件连接效果图

如上图，控制核心最终通过 DC 电源线连接移动电源供电，串口通讯则通过 esp8266 提供的局域网进行通讯。

4.3 上位机通讯结果

1) 上位机连接 eps8266

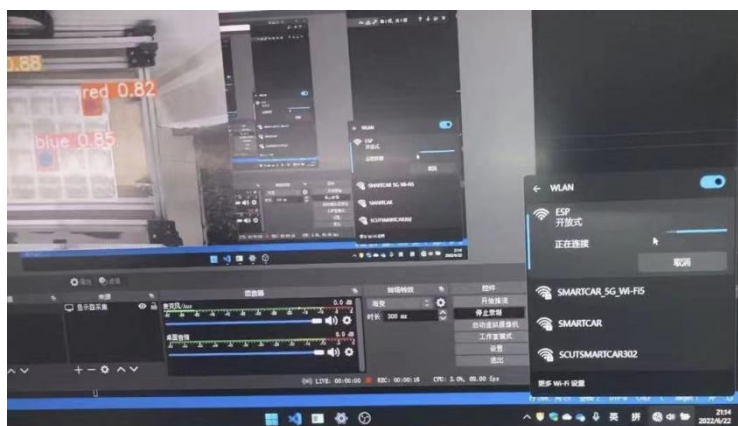


图 4-3 上位机连接

启动种植机电源和单片机的电源后，将出现单片机模块的局域网，上位机连接局域网。

2) 上位机与单片机进行通讯

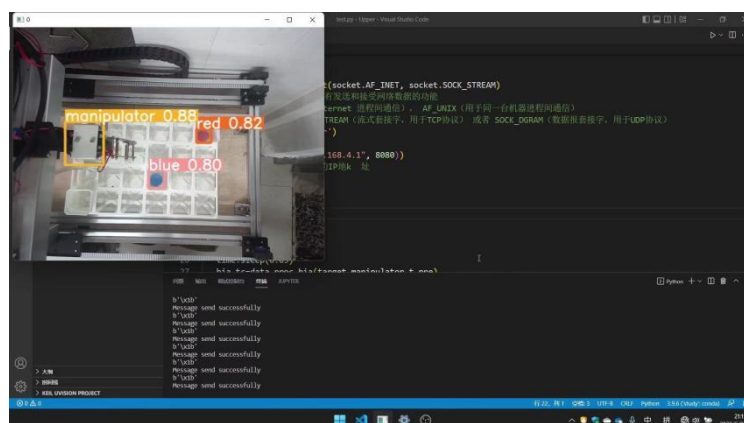


图 4-4 上位机连接

如上图，连接单片机的上位机进行无线通讯。每次通讯结束发送成功信号和打印发送数据。左上方为摄像机的监控以及目标检测算法处理结果。

4.4 种植机实现结果

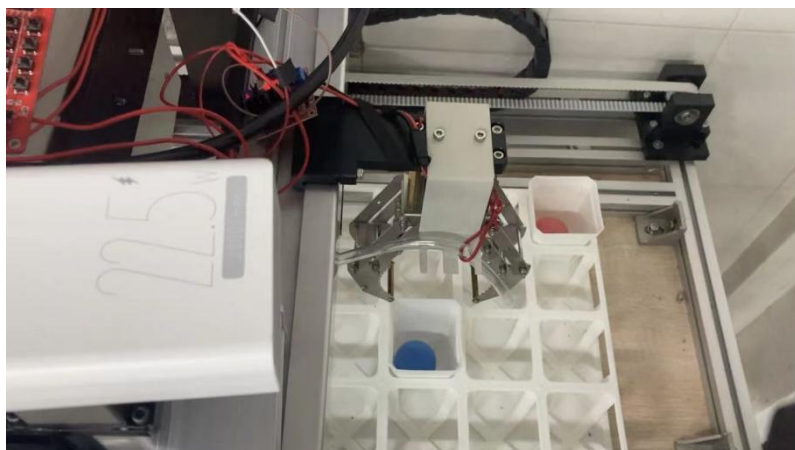


图 4-5 机械臂与目标球定位

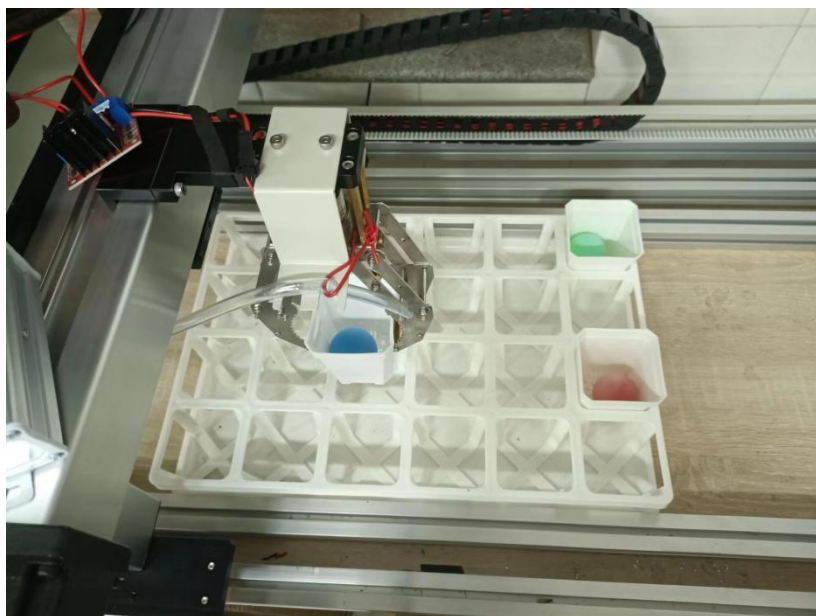


图 4-6 操作臂抓取图 1

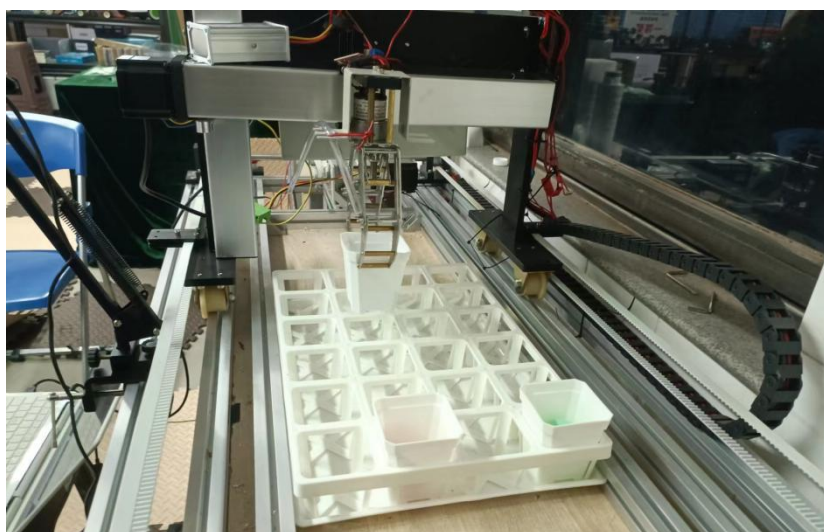


图 4-7 操作臂抓取图 2

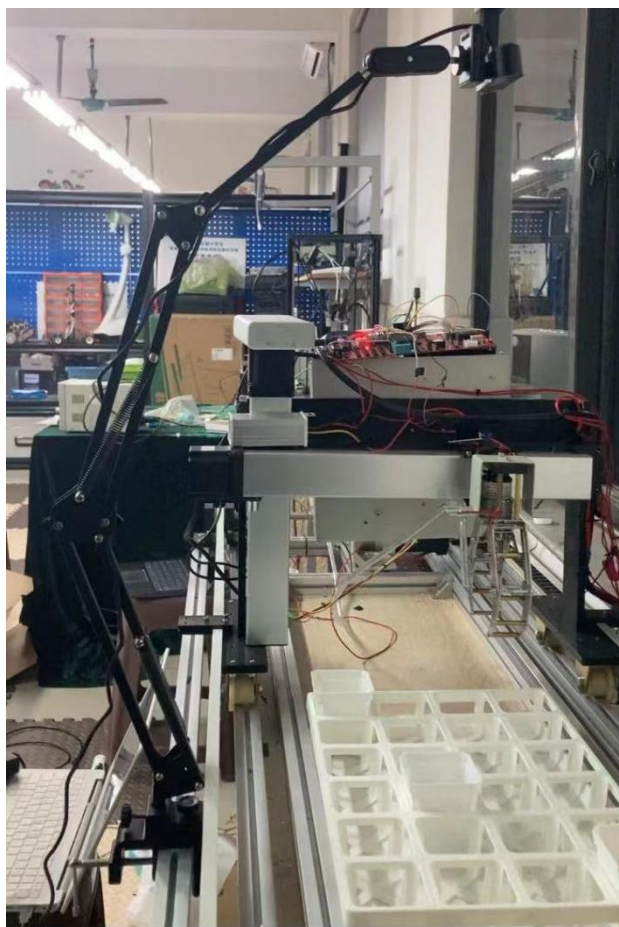


图 4-8 总体种植机图

目标为将篮球所在的盆栽夹起。将操作臂移动初始位置，打开摄像头，启动上位机程序，对小球和操作臂识别和定位，两者未重合之前，上位机程序不断通过串口发送目标球和操作臂的相对位置信息给单片机。单片机通过串行驱动程序接受该信息并向 x, y 轴步进电机发送脉冲信号，使得操作臂和目标球位置重合。

然后单片机运行夹起目标球程序，在操作臂沿 z 轴下降过程中，操作臂的位置随高度的下降而变化，此时操作臂和目标球 x, y 方向的位置又会出现一点偏差，但此时上位机仍然会发出串口信号让单片机驱动 x, y 轴步进电机进行位置偏差的补偿。将目标球夹到距离底部 15cm 处后，操作完成，如图 4-5，图 4-6，图 4-7，图 4-8 所示。

五、创新实践总结

一开始，当我看到这次的选题种植机还是一个个零部件的时候，确实是感到了无从下手，单单种植机，包括了软硬件地结合，又包括了被控对象、执行机构等简单闭环系统所有内容，除了种植机本身的组装，其余的内容全由我们小组进行。现在课程设计已经接近尾声，我们的预期目标也基本达成，回顾过去还是觉得不可思议、团队合作的不可思议。

对于我自己负责的无线通讯部分，一开始我特别的迷糊，对于这个领域可以说是一片空白，本科所学习的知识似乎也没有提及如何进行 WiFi 通讯实例，幸运的是，本次选题的智能家具给了我启发，从赛元开发板的 WiFi 通讯实例开始摸索，知道了 esp8266 模块，逐渐了解 esp8266 的 AT 指令以及其含义；对于上位机，我们组长提出要用 python 函数实现这个功能，在搜索相关方面知识时发现大家都是用网络调试助手进行调试，很少涉及到自己开发，因此我追本溯源，找到了开发这个软件作者编写的用户手册，根据开发原理自己用 python 程序编写。

这是我第一次觉得课程设计是如此新鲜，从上课所学的基础，延伸到了很多领域。不可避免地遇到了很多困难，有硬件选型错误、硬件电源供电问题等硬件问题，也有数据无法识别、上位机数据类型传输等软件问题，还有预计期望不理想等调试问题，最终都在组员和队长地陪伴下一一解决。

对于本次任务，我们小组的每个组员都非常认真甚至是刻苦地完成自己的目标，其中，队长更是付出了巨大心血，对于每个部分都有涉略，但是又非常地信任我们组员，相信我们能够完成，当我们出现问题时，捋清并提供一些新的思路。而且队长又有自己需要负责地部分，目标检测、超声波传感，甚至是不同队员之间的成果融合、优化都是他来完成。若是没有队长和我们地不断调试，从早上到晚上，是无法达到如今的成果的。因此，我觉得队长值得满分。