



基于 Hi3861 的家庭保健系统

林翔；杨旖；陈泳吉

第一部分 设计概述

1.1 设计目的

本项目旨在设计和开发一款基于华为海思Hi3861V100 芯片的智慧家庭保健物联网系统，以实现对用户健康状况的实时监测与管理。在满足小巧便利的同时，确保测量健康指标的准确，在当代快节奏生活模式下让用户也能随时了解和管理自己的身体健康指标状态，为生活忙碌和生活不变的用户提供更好的安全保障，提高家庭健康管理的便捷性和智能化水平。

1.2 应用领域

个人健康监护：系统能够实时监测个人的生理指标，如体温、心率、血氧饱和度、血压等，适用于需要长期健康监测的个人，尤其是老年人、慢性病患者以及康复期病人。通过集成智能药盒、联网体重秤等功能，系统能够帮助家庭成员管理日常用药、体重控制等，提高家庭健康管理的效率和质量。

公共卫生监测：在公共健康事件（如传染病爆发）期间，系统可以帮助政府和卫生部门收集和分析居民的健康数据，为公共卫生决策提供依据。

1.3 主要技术特点

系统由前端用户交互界面、智能药盒、语言控制系统、红外测温、心率血氧测量、联网体重秤、屏幕显示部分构成。通过 3861 联网功能实现主控与微信小程序的信息交互。微信小程序可以为药盒设定闹钟，并显示在前端界面，方便用户实时查看和更改闹钟，到达设定时间语言提醒服药用药，同时药盒会自动出药。通过连接的外设采集体温、体重、心率、血氧、血压等重要健康指标，并通过 3861 联网功能将数据上传到云端，智能分析家庭成员的健康状态后进行语音播报，同时会将各项健康指标显示在外设显示屏和前端界面上，直观反映出用户健康状态和给出针对性的建议。

1.4 关键性能指标

数据采集精度：系统所使用的各种传感器（如红外体温计、心率血氧仪、体重秤等）的数据采集精度较高，确保监测数据的准确性和可靠性。

响应时间：系统从接收用户指令到执行相应操作（如药盒出药、语音提醒、本地云端数据交互等）的响应时间在 0.5 秒左右，可以及时提醒用户吃药，及时观察数据，提供即时的用户体验。

数据传输安全性：系统在将健康数据上传至云端的过程中，采用用户名密码验证模式，保证MQTT传输数据时的安全性，防止隐私泄露。

语音识别准确率：语音控制系统对不同语调和语速的识别准确率高，以提供流畅的语音

交互体验。

系统可扩展性：系统架构应具有一定的扩展性，便于未来为家庭保健系统添加新的功能和升级。

1.5 主要创新点

（1）数据互通，综合多种智能传感器，将每项健康指标上传到主控，通过对这些传感器数据的综合分析，可以全面了解用户的健康状态，为用户提供更全面的健康管理服务。

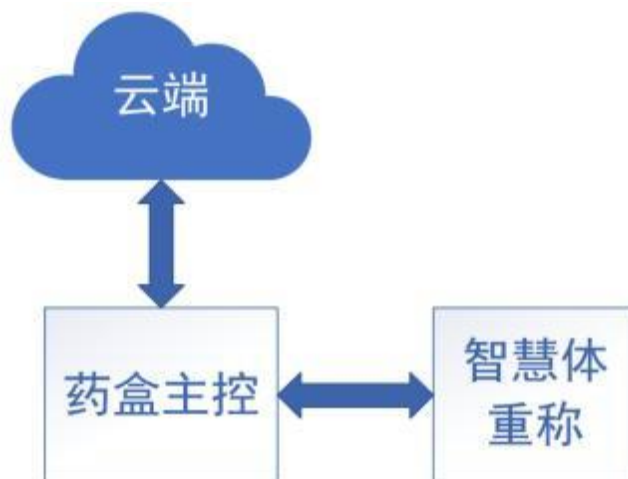
（2）远程监控和控制，可远程设置闹钟提醒用户吃药，为用户提供更舒适便捷的家居体验。

（3）智慧体重称具有检测BMI（身体质量指数）的功能，用户可以及时直观地了解自己的肥胖程度。

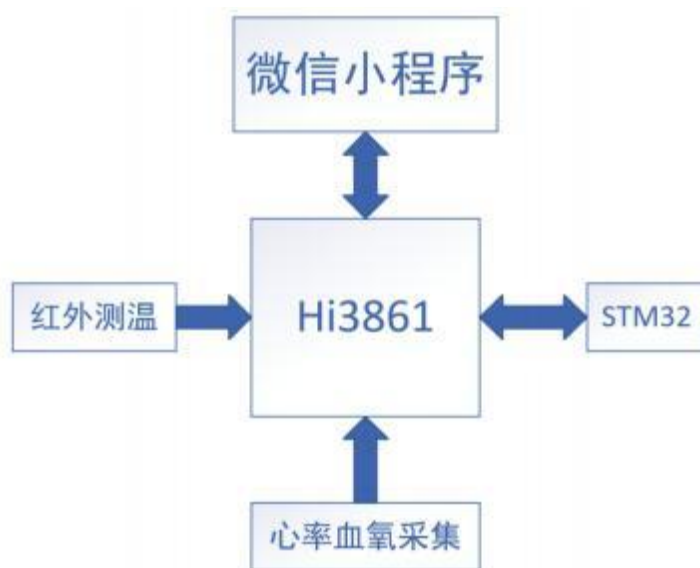
（4）使用语音和简易的触屏控制，没有复杂的按键，便于用户上手使用。

第二部分 系统组成及功能说明

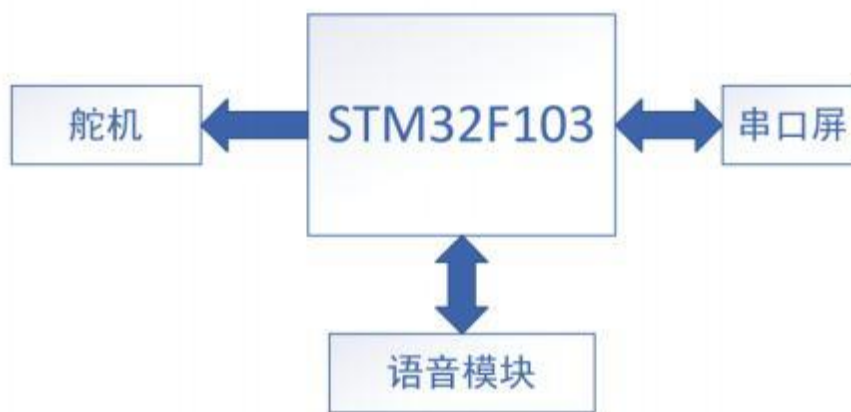
2.1 整体介绍



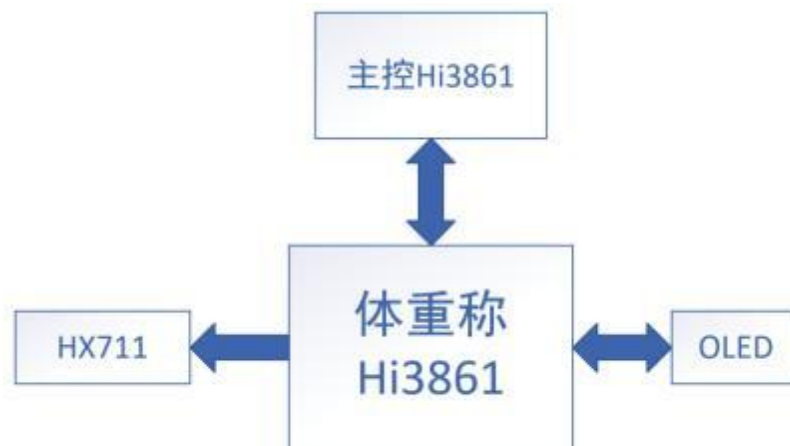
智慧家庭保健由药盒主控，智慧体重称和微信小程序构成。智慧体重称可以将采集到的用户体重数据上报给药盒主控，药盒主控收到云端下发身高数据后再发送给智慧体重称，体重称根据身高数据计算出用户的BMI并显示。药盒主控采集健康指标数据后上报给微信小程序。小程序可以设置用户的身高信息，为药盒设定闹钟。



主控Hi3861 可以与微信小程序进行数据交互，与红外测温传感器、心率血氧采集传感器和STM32F103 相连。Hi3861 与微信小程序通过MQTT协议通信可以将采集到的各类健康指标数据实时显示在微信小程序上，小程序可以设置用户的身高信息，为药盒设定闹钟；主控与红外测温传感器通过I2C协议通信，可以检测用户的体表温度数据；与心率血氧采集传感器和STM32F103 通过UART通信，采集用户的心率血氧数据，可以发送数据给 32，将数据在串口屏上显示。



STM32F103 与舵机、语音模块和串口屏相连。通过PWM波来控制舵机，当药盒定时闹钟时间到时，舵机控制药盒打开。可以通过语音控制药盒打开和关闭；通过USART与语音模块和串口屏通信，语音模块提醒用户吃药，播报用户健康指标状态。串口屏显示用户的各种健康指标数据。



智慧体重称的主控芯片也是Hi3861。与主控芯片通过Socket(TCP)进行通信，与OLED屏幕通过I2C协议通信，与HX711通过串行接口协议进行通信。智慧体重称可以将采集到的用户体重数据上报给药盒主控，药盒主控收到云端下发身高数据后再发送给智慧体重称，体重称根据身高数据计算出用户的BMI并通过OLED屏幕显示。

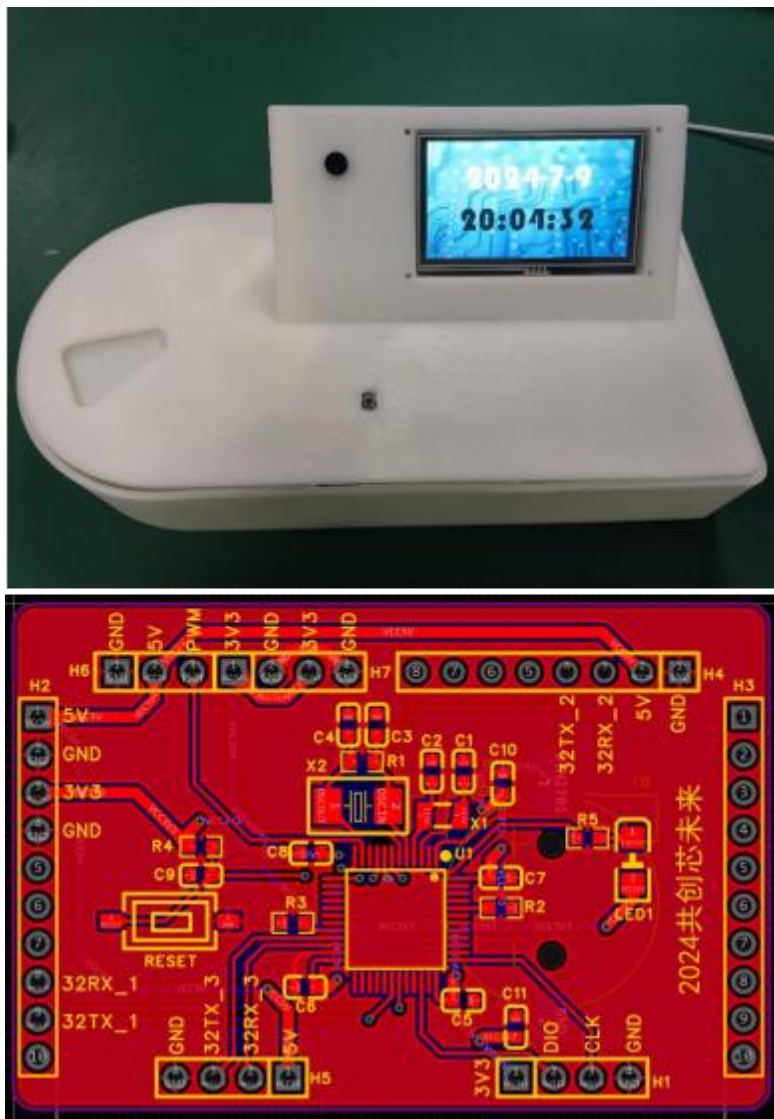
2.2 各模块介绍

(1) 无线通信：

利用华为海的高集成度Hi3861V100-2.4GHz Wi-Fi-SoC芯片，实现主控芯片与云端和体重称的无线通信。通过将健康数据上传到云端并进行智能分析，系统可以更加全面地了解家庭成员的健康状况，及时发现异常并提供相应的预警和建议。连接指定WIFI热点，通过DHCP协议动态分配服务器可用的IP地址，主控连接热点后通过MQTT协议实现主控与华为云的数据交互。主控设备与联网体重称使用TCP协议通信，主控设备作为服务端，联网体重称作为客户端，通过SOCKET接口将二者相连，实现数据互通。

(2) 主控部分：

包括药盒，温度采集，心率血氧采集、语音控制的功能。在主控程序中，使用CMSIS-RTOS2任务调度机制，合理安排各任务次序、任务间的互相调度，实现多复杂任务有条不紊的运行。主控实现驱动各传感器采集心率、血氧、体温数据，将接收到的数据处理；与云端和智慧体重称WIFI通信交互数据，将采集到的健康指标数据上报云端，将身高信息传递给智慧体重秤，接收传来的体重数据。其中stm32f103部分我们制作了PCB，以接插件的形式稳定地连接在Hi3861开发板上。



(3) 健康指标数据采集：

系统通过连接外设采集多项健康指标，实现了全面的健康监测。这些数据不仅可以实时上传到云端，还可以进行智能分析，为用户提供个性化的健康建议。通过连接多种高精度智能传感器准确采集用户的心率、血氧、温度数据，将采集到的心率、血氧、温度数据在串口屏幕上实时显示，方便用户及时查看。主控可以将传感器采集的数据上报给云端，便于在外家庭成员远程了解家中成员的健康状况。应用JF103 模组采集心率血氧、红外测温传感器JX90614 采集体温。我们采用的心率血氧模块JF103相较于传统的心率血氧模块具有测量更精准，反应速度更快等优点，能够精准及时反应人体的健康状况。

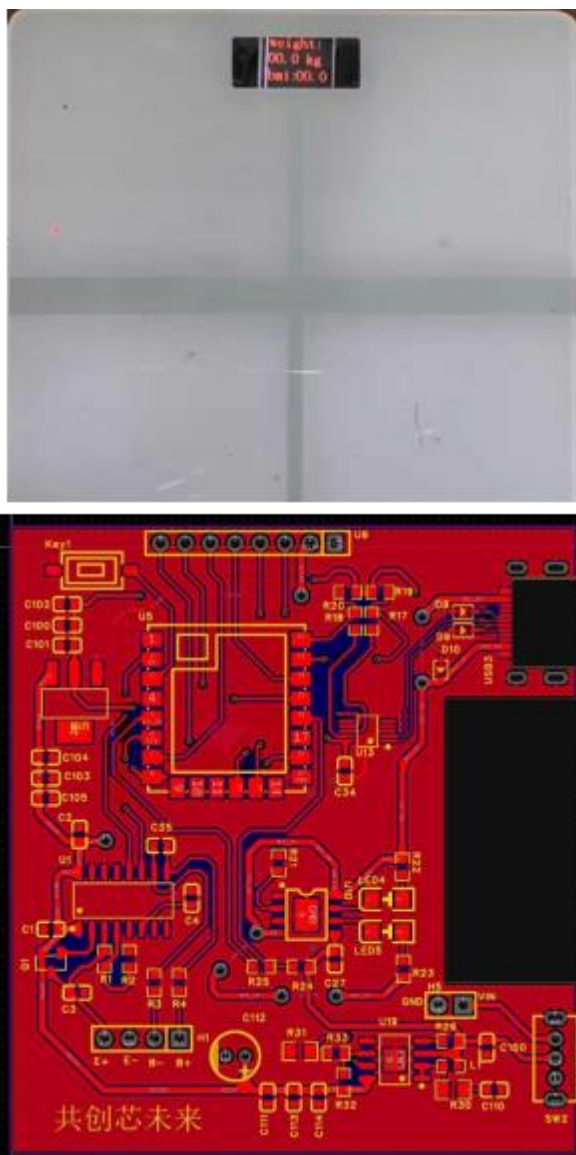
(4) 语音模块：

语音模块可以语音输出提醒用户吃药；语音播报用户各健康指标状态；语音接收相应语音指令并发送提示信号，主控收到信息后做出相应反应。药盒时间到，语音播报“主人该吃药了”提醒用户吃药。当用户说“播报我的健康指标状态”，语音模块识别相应语音给STM32发送提示信号，STM32收到信号后发送各健康指标的数据状态，语音模块接收后播报心率

、血氧、体温、BMI的健康信息。例如，红外测温传感器测得人体体温为 38°C，用户说“播报我的健康指标状态”后，语音播报“您的体温偏高”，若健康指标一切正常，则播报“您的健康指标一切正常”。

（5）智慧体重秤

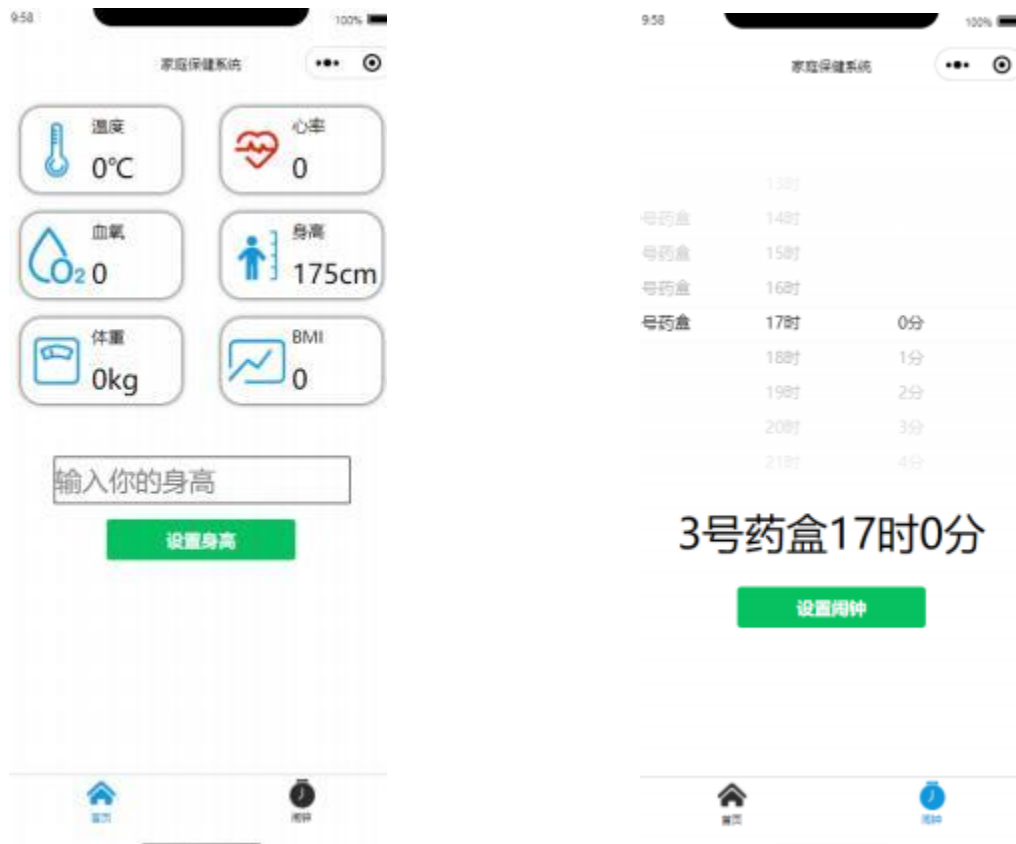
体重秤利用电阻应变半桥电路，差分放大，ADC转换，精确测量人体体重，将采集到的体重数据显示在OLED屏幕上。通过WIFI通信将采集到的体重数据上报给主控芯片，当主控发送用户设定的身高信息时，体重秤采集身高数据，并计算用户的BMI（身体质量指数），并将计算的结果显示在OLED屏幕上。体重秤由锂电池供电，可用TypeC接口为体重秤充电，体重秤PCB如下。



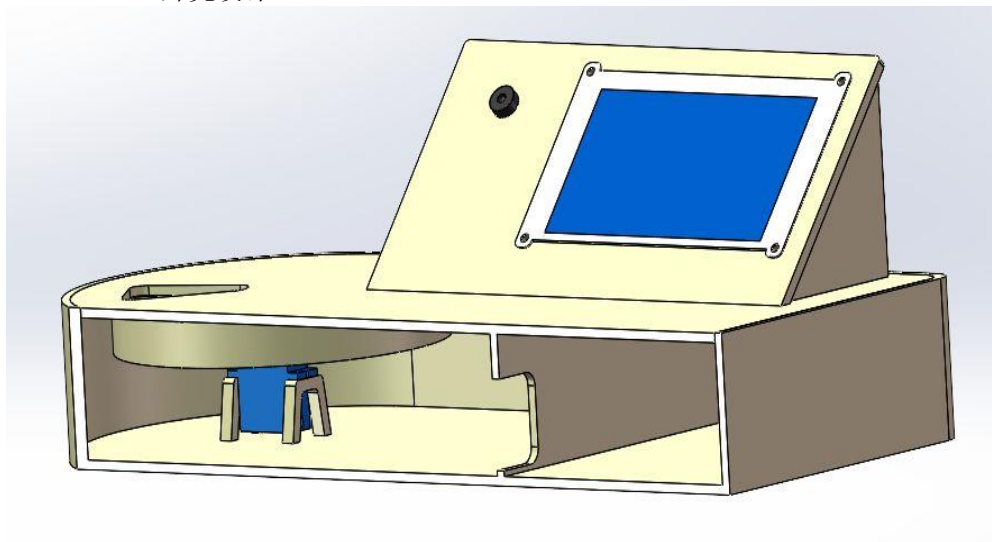
（6）微信小程序：

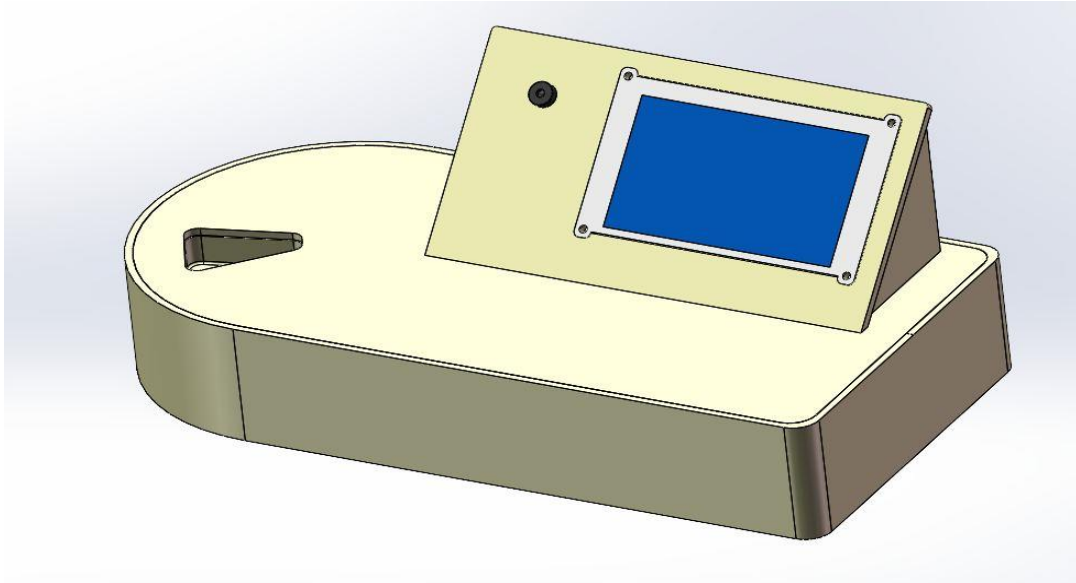
我们为本项目制作了微信小程序，进入小程序后可以查看采集到的各类健康指标数据，如温度、心率、血氧、体重、BMI。小程序输入用户的身高信息，下发给主控作为计算BMI

的依据。可以在闹钟页面为指定的药盒设定闹钟，提醒用户吃药。



(7) 3D外壳设计:





第三部分 完成情况及性能参数

该系统设置了主控药盒和体重秤两个设备,药盒设置了红外体温测量,血氧心率测量,智能语音助手,定时给药,屏幕显示等功能,体重秤设置了体重测量以及与主控通信等功能,上述功能在不断调试修改下均已完成。其中,红外测温模块JX90614 测量温度与医用体温枪测量温度基本一致,误差范围 0.1~0.3 摄氏度左右;在测试过程中,原本采用的MAX30102 模块并不能满足我们原本的设计需求,于是便选用了精度更高更为集成并且测量参数更多的JFC103 模块,心率与血氧数据在和专用医疗设备的数据对比下,精确度高达 92%,此外,该模块还能测量血压值、微循环、疲劳状态和心率变异性,具有很大的开发空间,可后续继续完善。智能语音助手选型为ASRPRO-4M,响应时间在 0.5 秒左右,高速精确。药盒给药通过SG92R舵机带动转盘式药舱完成,能使对应位置的药品精确抵达出药口。体重秤部分利用电阻应变半桥电路,差分放大、ADC转换,精确测量人体体重,并发送回主控药盒进行数据汇总。药盒外观采用了 3D 建模打印的方式,使用建模软件 3ds MAX进行药盒结构设计,药盒采用高密度光敏树脂材料打印,美观性强且结实牢靠,为智慧药盒各项功能的实现提供了重要的保障。

第四部分 总结

4.1 可扩展之处

（1）提升传感器检测精度，为用户提供更精准的健康管理。

（2）降低体重称功耗，未检测到体重时芯片处于休眠状态，检测到体重时，通过硬件电路给MOS管开关唤醒芯片，以降低功耗。

（3）给IOT系统增加更多设备，如测穿戴式手环，可以检测血压，血氧，心率等关键健康指标参数。满足更多用户的多样需求。

4.2 心得体会

在这个充满挑战与机遇的竞赛中，我不仅收获了宝贵的知识与技能，更深刻地体会到了团队合作的力量、创新思维的重要性。

（1）知识与实践的深度融合

嵌入式设计是一门理论与实践紧密结合的学科。在竞赛准备过程中，我深刻意识到仅仅掌握书本上的理论知识是远远不够的。通过亲手搭建硬件平台、编写驱动程序、调试软件逻辑，我真正地将理论知识应用到了实际项目中。这种“学中做，做中学”的方式，极大地提升了我的动手能力和问题解决能力。

（2）团队合作的力量

竞赛不是一个人的战斗，而是一个团队的协同作战。在团队中，我们各自发挥所长，又相互学习，共同进步。面对复杂的问题，我们集思广益，共同寻找解决方案。这种团队精神不仅帮助我们克服了技术上的难关，也让我们在紧张的竞赛氛围中感受到了温暖和支持。我深刻体会到，一个优秀的团队能够激发出每个人最大的潜能，实现个人与团队的共同成长。

（3）创新思维的激发

竞赛鼓励创新，这要求我们不仅要掌握现有的技术，更要敢于尝试新的思路和方法。在竞赛过程中，我们不断思考如何优化设计方案、提高系统性能、降低成本等。这些思考和实践不仅锻炼了我的创新思维，也让我学会了如何在有限的资源下寻求最优解。我认识到，创新是推动技术进步和社会发展的重要动力。

总之，参加此次竞赛是一次非常宝贵的经历。它不仅让我收获了知识和技能，更让我学会了团队合作、创新思维和面对困难的勇气与坚持。我将以这次竞赛为新的起点，继续努力学习与实践，为成为一名优秀的工程师而不懈奋斗。

第五部分 附录

数据上报:

```
1. static void ReportMsg(report_t *report)//上传 json 数据的拼装
2. {
3.     oc_mqtt_profile_service_t service;
4.     oc_mqtt_profile_kv_t Temperature;
```

```

5.    oc_mqtt_profile_kv_t Weight;
6.    oc_mqtt_profile_kv_t HeartRate;
7.    oc_mqtt_profile_kv_t BloodOxygen;
8.    oc_mqtt_profile_kv_t BMI;
9.
10.   //json 数据的拼装
11.   service.event_time = NULL;
12.   service.service_id = "Health_System";//服务 id, 与华为云上要一致
13.   service.service_property = &BMI;
14.   service.nxt = NULL;
15.
16.   char BMIStr[10];
17.   char WeightStr[10];
18.   E53_Float_To_String(rdata.BMI, BMIStr);
19.   BMI.key = "BMI";
20.   BMI.value = BMIStr;
21.   BMI.type = EN_OC_MQTT_PROFILE_VALUE_STRING;
22.   BMI.nxt = &BloodOxygen;
23.
24.   BloodOxygen.key = "BloodOxygen";
25.   BloodOxygen.value = &rdata.BloodOxygen;
26.   BloodOxygen.type = EN_OC_MQTT_PROFILE_VALUE_INT;
27.   BloodOxygen.nxt = &HeartRate;
28.
29.   HeartRate.key = "HeartRate";
30.   HeartRate.value = &rdata.HeartRate;
31.   HeartRate.type = EN_OC_MQTT_PROFILE_VALUE_INT;
32.   HeartRate.nxt = &Weight;
33.
34.   E53_Float_To_String(rdata.Weight, WeightStr);
35.   Weight.key = "Weight";
36.   Weight.value = WeightStr;
37.   Weight.type = EN_OC_MQTT_PROFILE_VALUE_STRING;
38.   Weight.nxt = &Temperature;
39.   //上下对应连接
40.   Temperature.key = "Temperature";
41.   Temperature.value = &report->Temperature;//&report->Temperature;
42.   Temperature.type = EN_OC_MQTT_PROFILE_VALUE_INT;
43.   Temperature.nxt = NULL;
44.
45.   int statue = oc_mqtt_profile_propertyreport(USERNAME, &service);/
    /上报数据

```

```

46.     if(statue == en_oc_mqtt_err_ok)
47.     {
48.         printf("上报成功\n");
49.     }
50.     else{
51.         printf("上报失败\n");
52.     }
53.     return;
54. }

```

命令解析:

```

1. static void DealCmdMsg(cmd_t *cmd)//命令 json 解析
2. {
3.     cJSON *obj_root, *obj_cmdname, *obj_paras, *obj_para;
4.
5.     int cmdret = 1;
6.
7.     obj_root = cJSON_Parse(cmd->payload);
8.     if (obj_root == NULL) {
9.         oc_cmdresp(cmd, cmdret);
10.    }
11.
12.    obj_cmdname = cJSON_GetObjectItem(obj_root, "command_name");
13.    if (obj_cmdname == NULL) {
14.        cJSON_Delete(obj_root);
15.    }
16.    //定闹钟
17.    if (strcmp(cJSON_GetStringValue(obj_cmdname), "Medicine_Clock") =
        = 0)
18.    {
19.        obj_paras = cJSON_GetObjectItem(obj_root, "Paras");
20.        if (obj_paras == NULL) {
21.            cJSON_Delete(obj_root);
22.        }
23.        obj_para = cJSON_GetObjectItem(obj_paras, "Clock");
24.        if (obj_para == NULL) {
25.            cJSON_Delete(obj_root);
26.        }
27.        char* command = cJSON_GetStringValue(obj_para);//指令赋值
28.        printf("s = %s\n", command);//收到命令下发信息
29.        MY_USART1_SendString(command);//发送命令给 32
30.
31.        medicine[command[0] - '0'].clock_status = 1;//药盒时钟开启

```

```

32.         strcpy(medicine[command[0] - '0'].command, command); //记录药盒
    指令
33.         printf("command = %s\n", medicine[command[0] - '0'].command);

34.
35.         cmdret = 0;
36.     }
37.     else if(strcmp(cJSON_GetStringValue(obj_cmdname), "Height") == 0)
38.     {
39.         obj_paras = cJSON_GetObjectItem(obj_root, "Paras");
40.         if (obj_paras == NULL) {
41.             cJSON_Delete(obj_root);
42.         }
43.         obj_para = cJSON_GetObjectItem(obj_paras, "height");
44.         if (obj_para == NULL) {
45.             cJSON_Delete(obj_root);
46.         }
47.         rdata.Height = cJSON_GetNumberValue(obj_para);
48.         printf("height = %d\n", rdata.Height);
49.         cmdret = 0;
50.     }
51.
52.     cJSON_Delete(obj_root);
53. }

```

与体重称通信:

```

1. static void TCPServerTask(void)
2. {
3.     // 在 sock_fd 进行监听, 在 new_fd 接收新的链接
4.     int sock_fd, new_fd;
5.     // 服务端地址信息
6.     struct sockaddr_in server_sock;
7.     // 客户端地址信息
8.     struct sockaddr_in client_sock, *cli_addr;
9.     int sin_size;
10.    // 创建 socket
11.    if ((sock_fd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
12.        perror("socket is error\n");
13.        return;
14.    }
15.    //填入 bind 的信息
16.    bzero(&server_sock, sizeof(server_sock));
17.    server_sock.sin_family = AF_INET;

```

```

18.     server_sock.sin_addr.s_addr = htonl(INADDR_ANY); //ip 地址为
        0.0.0.0, 等待分配
19.     server_sock.sin_port = htons(CONFIG_CLIENT_PORT); //设置端口号
20.     // 调用 bind 函数绑定 socket 和地址
21.     if (bind(sock_fd, (struct sockaddr *)&server_sock, sizeof(struct
        sockaddr)) == -1) {
22.         return;
23.     }
24.     // 调用 listen 函数监听(指定 port 监听)
25.     if (listen(sock_fd, TCP_BACKLOG) == -1) {
26.         return;
27.     }
28.     printf("start accept\n");
29.     // 调用 accept 函数从队列中
30.     while (1) {
31.         sin_size = sizeof(struct sockaddr_in);
32.         if ((new_fd = accept(sock_fd, (struct sockaddr *)&client_sock
            , (socklen_t *)&sin_size)) == -1) {
33.             perror("accept");
34.             continue;
35.         }
36.         cli_addr = malloc(sizeof(struct sockaddr));
37.         printf("accept addr\r\n");
38.         if (cli_addr != NULL) {
39.             int ret;
40.             if (ret = memcpy_s(cli_addr, sizeof(struct sockaddr), &c
                lient_sock, sizeof(struct sockaddr)) != 0) {
41.                 perror("memcpy is error\r\n");
42.                 return;
43.             }
44.         }
45.         // 处理目标
46.         ssize_t ret;
47.         char recvbuf[512]; //接收的数据
48.         char buf[10]; //上报的数据
49.         while (1) {
50.             memset_s(recvbuf, sizeof(recvbuf), 0, sizeof(recvbuf)); //
                数据清零
51.             printf("ready receive\n");
52.             if ((ret = recv(new_fd, recvbuf, sizeof(recvbuf), 0)) ==
                -1) {
53.                 printf("recv error \r\n");
54.             }

```

```

55.         printf("recv :%s\r\n", recvbuf); //收到体重数据
56.         strcpy(w, recvbuf);
57.         rdata.Weight = E53_Weight(w);
58.         E53_Height_To_String(rdata.Height, buf);
59.         if ((ret = send(new_fd, buf, strlen(buf) + 1, 0)) == -1)
        {
60.             perror("send : ");
61.         }
62.         printf("ret = %d, send_buf = %s\n", ret, buf);
63.     }
64.     close(new_fd);
65. }
66. }

```

微信小程序连接 mqtt:

```

1. connectmqtt:function(){
2.   var that=this
3.   const options={
4.     // connectTimeout:4000,
5.     clientId:'wxmp'+Math.ceil(Math.random()*10),
6.     port:8084,
7.     username:'fc70ac3fa5e8792e875f23c7a88b5f4b',
8.     password:'123123'
9.   }
10.  client=mqtt.connect('wxs://t.yoyolife.fun/mqtt',options)
11.  client.on('connect',(e)=>{
12.    console.log('服务器连接成功 ')
13.    client.subscribe('/iot/5898/pub/1', {qos: 0}, function(err){
14.      if (!err) {
15.        console.log('subscribe message successful');
16.      }
17.    })
18.  })
19.  //信息监听
20.  client.on('message',function(topic,message){
21.    let recv={}
22.    recv=JSON.parse(message)
23.    console.log(recv)
24.    that.setData({
25.      temp:recv.services[0].properties.Temperature,
26.      heart:recv.services[0].properties.HeartRate,
27.      blood:recv.services[0].properties.BloodOxygen,
28.      //height:recv.services[0].properties.Height,
29.      BMI:recv.services[0].properties.BMI,

```



```
30.     weight:recv.services[0].properties.Weight
31.   })
32.   console.log('temp='+that.data.temp)
33.   console.log('heart='+that.data.heart)
34.   console.log('blood='+that.data.blood)
35.   console.log('weight='+that.data.weight)
36.   console.log('BMI='+that.data.BMI)
37.   console.log('收到 '+message.toString())
38. })
39. client.on('reconnect',(error)=>{
40.   console.log('正在重连 ',error)
41. })
42. client.on('error',(error)=>{
43.   console.log('连接失败 ',error)
44. })
45. },
```