

单片机应用系统 创意设计报告



设计题目： 智能工具传递与放置系统

学 院： 电子工程学院

时 间： 2014 年 7 月

成 绩：

指导教师： 余柏生

西安电子科技大学

单片机应用系统创作课程

项目报告

题 目： 智能工具传递与放置系统设计

学 院： 电子工程学院

指导教师： 余柏生

组 别： 本科组

平 台： Arduino Nano3/Atmega328

参赛队成员名单：

| 姓名 | 学号 | 学院 | 专业 | 邮箱 |
|-----|----------|--------|-----------|-------------------|
| 沈云彬 | 02116026 | 电子工程学院 | 电磁场与无线技术 | 461372076@qq.com |
| 姜震华 | 02116002 | 电子工程学院 | 电磁场与无线技术 | 1194119617@qq.com |
| 余春海 | 02114018 | 电子工程学院 | 探测制导与控制技术 | 2293301355@qq.com |
| 马良基 | 02116027 | 电子工程学院 | 电磁场与无线技术 | 1797183619@qq.com |
| 费秋嫻 | 02116033 | 电子工程学院 | 电磁场与无线技术 | 2503801332@qq.com |

题 目： 智能工具传递与放置系统设计

关键词： **Arduino**、智能、机器人、传递、检测

摘要

我们主要利用 **arduino** 单片机完成机器人的物品拿放、工具感知等一系列功能。需要实现的功能大致包含以下几个：

1. 系统可以通过测力传感器通过物品的重量来感知工具的类型，并放入相应位置。
2. 操作者通过按键或者声音识别命令机器人取相应的物品。
3. 系统可以通过人体远红外传感器来感应人所在的方位把工具传递到自己所在的地方。
4. 当我们把工具放到机械手上时，机器人会自动识别工具并放到工具应该放的位置。

另外，系统所包含的传感器大概有：超声波传感器（用来测与人的距离）、远红外传感器（用来感知人所在方位）、红外对管（用来感知机械手所处的位置）、测力传感器（用来感知工具类型）。

Abstract

We mainly use **arduino** single chip microcomputer to finish a serious of functions for the things to pick up and lay down,the tools to sense.The functions in need to realize roughly include as the following:

- 1.The system can perceive the tools' types via the weight of things by the load cell,and then put to the corresponding positions.
- 2.The operator is able to press the key or identify the voice to order the robot to fetch relevant goods.
- 3.The system can sense the person's location via far infrared sensor and transfer the tools to the position.

4.As we put the tool on the machinery hand,the robot can automatic identify the tool and put it to the corresponding positions.

In addition,the system include mainly:ultrasonic sensors(in use of measuring the distance to the human), far infrared sensor(in use of sensing human's position),infrared tube(in use of sensing machinery hand's location),the load cell(in use of sensing the tool's types).

1. 引言

当前在很多关于物品的维修、装配行业中，工人需要用到很多种工具，这样，工具的拿来和放置浪费了很多宝贵的时间。在现在的仓库里面，想要用人工去找一样东西也不是一件容易的事情。为了减少工人寻找物品、拿来物品的时间，并且同时降低工人的劳动强度，我们设计了智能物品传递与放置系统。

系统主要功能是可以根据物品的特征自动识别物品，并放入物品应在的位置，并根据人的指令来拿取相应的物品。

2. 系统硬件模块介绍

2.1.红外接收模块(VS1838):

2.1.1.主要功能

小型设计

内置专用 IC;

宽角度及长距离接收;

抗干扰能力强;

能抵挡环境干扰光线;

低电压工作;

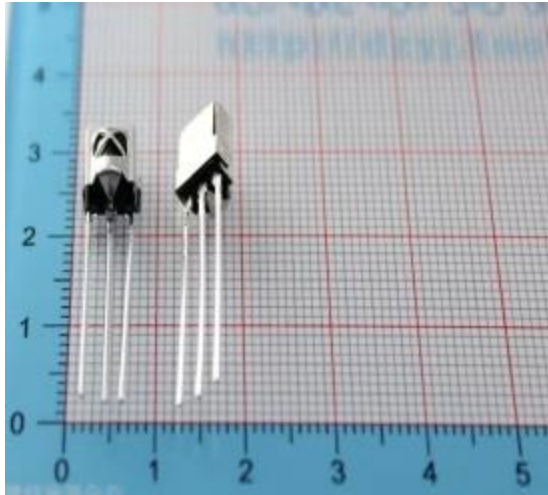
2.1.2.基本参数

光电参数($T=25^{\circ}\text{C}$ $V_{cc}=5\text{V}$ $f_0=38\text{KHZ}$):

| 参 数 | 符号 | 测试条件 | Min | Type | Max | 单 位 |
|-------------|----------------|---------------------------------------|--------------|------|----------|---------------|
| 工作电压 | V_{cc} | | 2.7 | | 5.5 | V |
| 工作电流 | I_{cc} | | 0.6 | 0.8 | — | mA |
| 静态电流 | I_{ce} | 无信号输入时 | 0.1 | | 0.5 | mA |
| 接收距离 | L | ※ | 20 | 23 | | M |
| 接收角度 | $\theta_{1/2}$ | | +/-35 | | | Deg |
| 载波频率 | F_0 | | | 38 | | KHZ |
| BMP 宽度 | f_{BW} | -3Db Bandwidth | — | 8 | — | kHz |
| 低电平输出 | V_{OL} | $V_{in}=0\text{V}$ $V_{cc}=5\text{V}$ | | | 0.4 | V |
| 高电平输出 | V_{OH} | $V_{cc}=5\text{V}$ | $V_{cc}-0.3$ | | V_{cc} | V |
| 输出脉冲 宽 度 | T_{PWL} | $V_{in}=50\text{mVp-p}$ | 500 | 600 | 700 | μS |
| | T_{PWH} | $V_{in}=50\text{mVp-p}$ | 540 | 640 | 740 | μS |

※ 光轴上测试, 以宽度 600/900 μs 为发射脉冲, 在 5CM 之接收范围内, 取 50 次接收脉冲之平均值。

2.1.3.模块图片



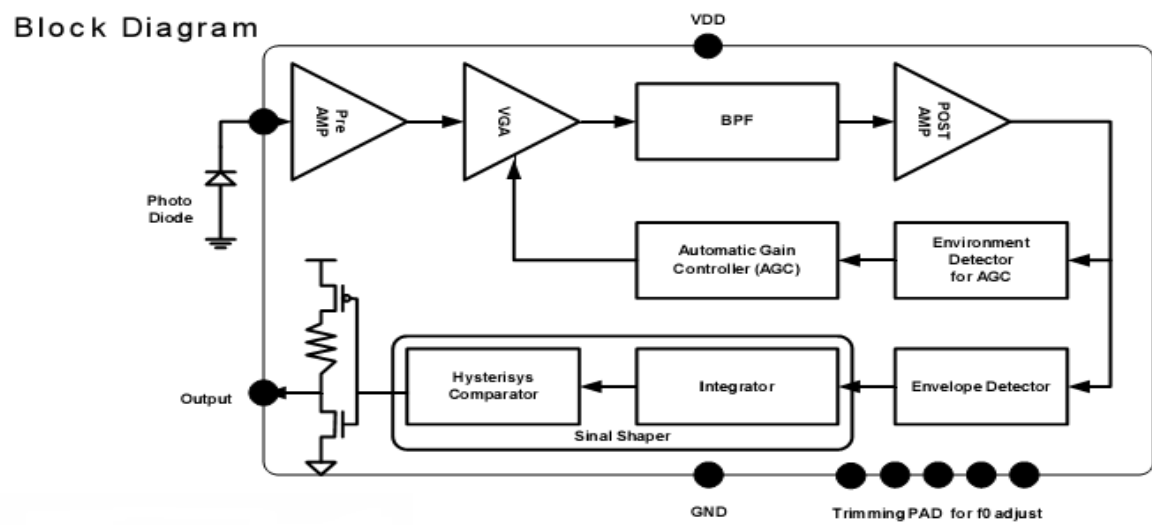
2.1.4.接口定义:

Vcc——电源

Gnd——接地

OUT——输出

2.1.5.模块工作原理:



2.1.6 注意事项:

- 在无任何外加压力及影响品质的环境下储存及使用;
- 在无污染气体或海风(含盐分)的环境下储存及使用;
- 在低湿度环境下储存及使用;
- 在规定的条件下焊接引线管脚,焊接后,请勿施加外力
- 请勿清洗本产品,使用前,请先用静电带将作业员及电烙铁连接落地线
- 请注意保护红外线接收器的接收面,沾污或磨损后会影响接收效果,同时不要触碰表面

2.2.MG995 模拟舵机:

2.2.1.主要功能

1:10 和 1:8 平跑车、越野车、卡车、大脚车、攀爬车、双足机器人、机械手、遥控船,适合 50 级-90 级甲醇固定翼飞机以及 26cc-50cc 汽油固定翼飞机等模型。

2.2.2.基本参数

- 产品型号 MG995
- 尺寸 40.7*19.7*42.9mm
- 重量 55kg
- 工作扭矩 13kg/cm
- 反应转速 53—62R/M
- 使用温度 -30~+60°
- 死区设定 4 微秒
- 插头类型 JR、FUTABA 通用
- 转动角度 最大 180 度
- 工作电流 100mA
- 使用电压 3—7.2V

2.2.3.模块图片



2.2.4.接口定义:

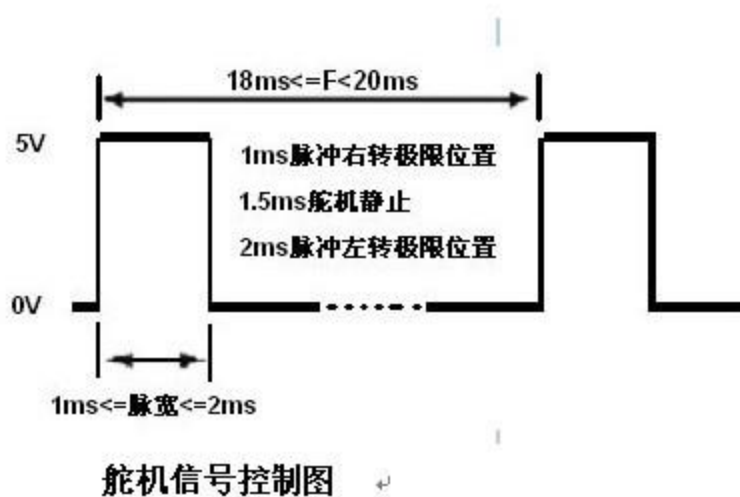
Vcc——电源

Trig——控制端

Echo——接收端

Gnd——接地

2.2.5.模块工作原理:



- 采用 IO 触发测距，给至少 10us 的高电平信号；
- 模块自动发送 8 个 40kHz 的方波，自动检测是否有信号返回；
- 有信号返回，通过 IO 输出一高电平，高电平持续的时间就是
- 超声波从发射到返回的时间。测试距离=(高电平时间*声速(340M/S))/2;

2.2.6.注意事项:

13 千克扭力

使用温度-30 到 60 度

2.3.EEPROM 储存模块(arduino 自带):

2.3.1.主要功能

- 读
- 写
- 一种掉电后数据不丢失的存储芯片。

2.3.2.基本参数

在各型号的 arduino 控制器上的 AVR 芯片均带有 EEPROM, 也有外接的 EEPROM 芯片, 常见 arduino 控制器的 EEPROM 大小:

Arduino UNO、Arduino duemilanove-m328、Zduino m328 均使用 ATmega328 芯片, EEPROM 都为 1K

Arduino duemilanove-m168 的 EEPROM 为 512bytes

Arduino 2560 的 EEPROM 为 4K 在各型号的 arduino 控制器上的 AVR 芯片均带有 EEPROM, 也有外接的 EEPROM 芯片, 常见 arduino 控制器的 EEPROM 大小:

Arduino UNO、Arduino duemilanove-m328、Zduino m328 均使用 ATmega328 芯片, EEPROM 都为 1K

Arduino duemilanove-m168 的 EEPROM 为 512bytes

Arduino 2560 的 EEPROM 为 4K

2.3.3.注意事项:

EEPROM 是几种存取次数有限的半导体存储器之一, 它可以读无数次, 但编程次数却有限, 这种受限的原因在于它的结构。EEPROM 单元的预期寿命取决于浮动栅和基片之间的隧道氧化层的类型、厚度和质量。

2.4.超声波测距模块(HC-SR04):

2.4.1.主要功能

- 距离测量

2.4.2.基本参数

- 典型工作电压：5V。
- 超小静态工作电流：小于 2mA。
- 感应角度：不大于 15 度。
- 探测距离：2cm-400cm
- 高精度：可达 0.3cm。
- 盲区（2cm）超近。
- 完全兼容 GH-311 防盗模块。

2.4.3.模块图片



2.4.4.接口定义：

Vcc——电源

Trig——控制端

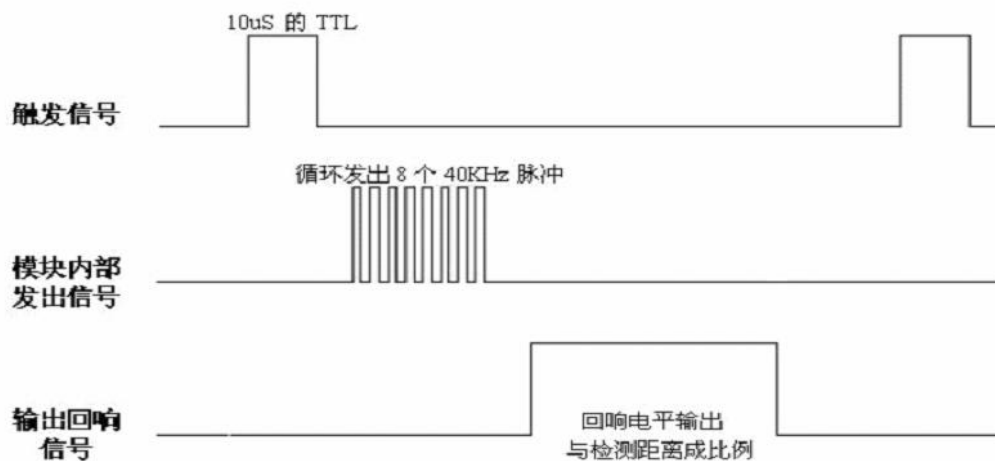
Echo——接收端

Gnd——接地

2.4.5.模块工作原理：

- 采用 IO 触发测距，给至少 10us 的高电平信号；
- 模块自动发送 8 个 40kHz 的方波，自动检测是否有信号返回；
- 有信号返回，通过 IO 输出一高电平，高电平持续的时间就是
- 超声波从发射到返回的时间。测试距离= $(\text{高电平时间} \times \text{声速}(340\text{M/S}))/2$;

超声波时序图：



图二、 超声波时序图

2.4.6 注意事项：

- 此模块不宜带电连接，如果要带电连接，则先让模块的 Gnd 端先连接。否则会影响模块工作。
- 测距时，被测物体的面积不少于 0.5 平方米且要尽量平整。否则会影响测试结果。

2.5.人体远红外检测模块(HC-SR501):

2.5.1.主要功能

- 人体红外感应

2.5.2.基本参数

- 工作电压：DC5V 至 20V
- 静态功耗： 65uA
- 电平输出： 高 3.3v，低 0V
- 延时时间：可调（0.3 秒~10 分钟）
- 封锁时间：0.2 秒
- 触发方式：L 不可重复，H 可重复，默认值为 H
- 感应范围：小于 120 度锥角，7 米以内
- 工作温度：-15~+70 度
- PCB 外形尺寸：32*24mm,螺丝孔距 28mm，螺丝孔径 2mm，感应透镜尺寸：（直径）：23mm（默认）。

2.5.3.模块图片





2.5.4.接口定义:



2.5.5.模块工作原理:

热释电效应同压电效应类似，是指由于温度的变化而引起晶体表面荷电的现象。热释电传感器是对温度敏感的传感器。它由陶瓷氧化物或压电晶体元件组成，在元件两个表面做成电极，在传感器监测范围内温度有 ΔT 的变化时，热释电效应会在两个电极上会产生电荷 ΔQ ，即在两电极之间产生一微弱的电压 ΔV 。由于它的输出阻抗极高，在传感器中有一个场效应管进行阻抗变换。热释电效应所产生的电荷 ΔQ 会被空气中的离子所结合而消失，即当环境温度稳定不变时， $\Delta T=0$ ，则传感器无输出。当人体进入检测区，因人体温度与环境温度有差别，产生 ΔT ，则有 ΔT 输出；若人体进入检测区后不动，则温度没有变化，传感器也没有输出了。所以这种传感器检测人体或

者动物的活动传感。由实验证明，传感器不加光学透镜(也称菲涅尔透镜)，其检测距离小于 2m，而加上光学透镜后，其检测距离可大于 7m。

2.5.6 注意事项：

一、直流工作电压必须符合我们要求的数值，过高和过低都会影响模块性能，而且要求电源必须经过良好的稳压滤波，例如电脑 USB 电源、手机充电器电源、比较旧的9V 的层叠电池都无法满足模块工作要求，建议客户用变压器的电源并经过三端稳压芯片稳压后再通过220UF 和0.1UF 的电容滤波后供电。

二、调试时人体尽量远离感应区域，有时虽然人体不在模块的正前方，但是人体离模块太近时模块也能感应到造成一直有输出，还有调试时人体不要触摸电路部分也会影响模块工作，比较科学的办法是将输出端接一个 LED 或者是万用表，把模块用报纸盖住，人离开这个房间，等2分钟后看看模块是否还是一直有输出？

三、模块不接负载时能正常工作，接上负载后工作紊乱，一种原因是因为电源容量很小负载比较耗电，负载工作时引起的电压波动导致模块误动作，另一种原因是负载得电工作时会产生干扰，例如继电器或者电磁铁等感性负载会产生反向电动势，315M 发射板工作时会有电磁辐射等都会影响模块。解决办法如下：A、电源部分加电感滤波。B、采用负载和模块使用不同的电压的方法，例如：负载使用24V 工作电压，模块使用12V 工作电压，其间用三端稳压器隔离。C：使用更大容量的电源。

四、人体感应模块只能工作在室内并且工作环境应该避免阳光、强烈灯光直接照射，如果工作环境有强大的射频干扰，可以采用屏蔽措施。若遇有强烈气流干扰，关闭门窗或阻止对流。感应区尽量避免正对着发热电器和物体以及容易被风吹动的杂物和衣物。

五、人体感应模块建议安装在密封的盒里，否则可能一直会有输出信号。

六、如果要求人体感应模块的探测角度小于90度时，可以用不透明胶纸遮挡镜片或裁剪缩小镜片来实现。

七、人体感应模块采用双元探头，人体的手脚和头部运动方向与感应灵敏度有着密切的联系，而且红外模块的特性决定了无法精确控制感应距离。

八、模块中的探头（PIR）可以装焊在电路板的另一面。也可将探头用双芯屏蔽线延长，长度应在20厘米以内为好。

2.6.测力模块（应力变阻片+ HX711 24 位模数转换模块）：

2.6.1.主要功能

- 3kg 称重传感器 电子秤 重量 压力应力 测重测力

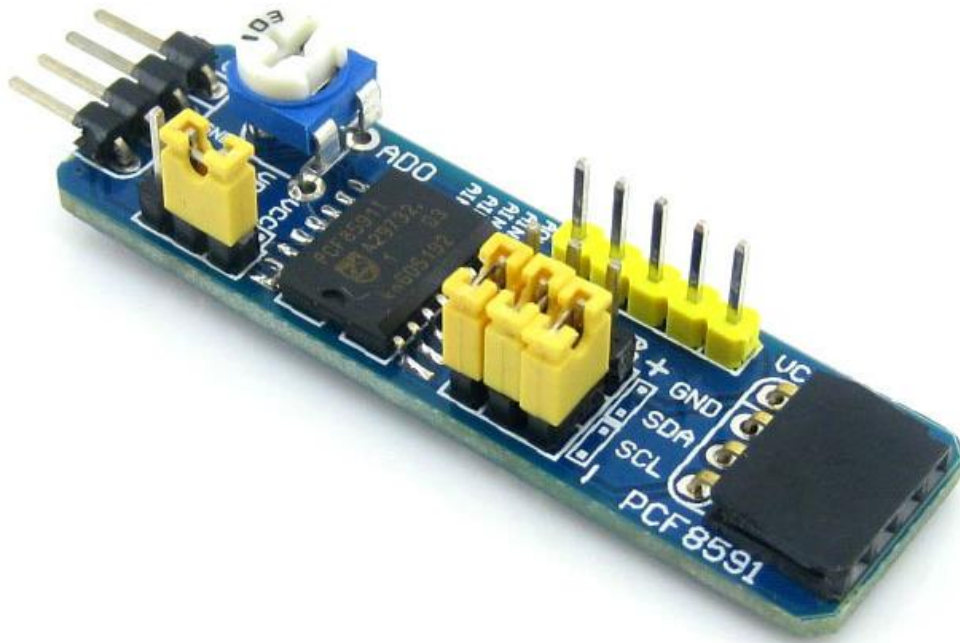
2.6.2.基本参数

传感器尺寸：长 80mm 宽 12.7mm 高 12.7mm

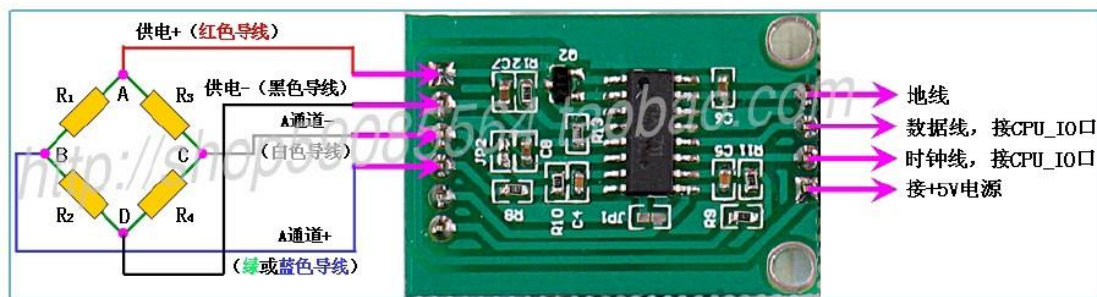
规格：3kg

| | |
|----------|-------------------------------|
| 额定载荷 | 3KG |
| 综合误差: | 0.05%F.S |
| 绝缘电阻: | $\geq 2000M\Omega$ |
| 灵敏度: | 1mv/v |
| 激励电压: | 5~10VDC |
| 非线性: | 0.05%F.S |
| 滞后: | 0.05%F.S |
| 重复性: | 0.05%F.S |
| 零点温度影响: | 0.02F.S/10℃ |
| 蠕变: | 0.05%F.S/3min |
| 灵敏度温度影响: | 0.2F.S/10℃ |
| 零点输出: | $\pm 0.15\%F.S + 0.005(mV/V)$ |
| 安全过载范围: | 150%(F.S) |
| 输入阻抗: | $1000 \pm 50 \Omega$ |
| 输出阻抗: | $1000 \pm 50 \Omega$ |
| 推荐激励电压: | 5--10V |

2.6.3.模块图片



2.6.4.接口定义:



输入电源+：红，输入电源—：黑，输出信号+：绿，输出信号—：白

1. VCC 可以是 2.6-5.5 中的任意值，因为我们使用的是 Arduino，所以直接 5V 供电，GND 接地。
2. SCK 接 Arduino 的 Pin 9, DT 接 Pin 10, 这两个接脚可以在程序中改变。
3. E+、E-、A+ 和 A- 分别接桥式传感器的：激励电压正、负，输出电压正、负

(E+ 接红线；E- 接黑线；A+ 接绿或蓝线；A- 接白线)。

4. B+ 和 B- 接通道 B 的传感器，也可以通过分压电路接电源，用来检测电源电压。不用的话最好接 GND，不过我试验不接也没问题。

PCF8591, I2C 接口排针, I2C 接口排座, 可调电阻, AD 输入口, DA 输出口, 地址跳线端口

2.6.5.模块工作原理：

24 位 ADC 模块（仅电路板模块，不包括传感器）：

ADC 模块采用 24 位高精度的 HX711 芯片,内部集成 128 倍增益可编程放大器,有两路模拟通道输入。输入电路可配置为提供桥压的电桥式（如压力、称重）传感器模式；也可配置为电压表测量模式。

该模块可以 3.3—5.5V 供电,可用于 5V CPU 系统,也可用于 MSP430、LPC900、C8051F 等低压供电的 CPU 系统，是理想的高精度、低成本采样前端模块。

由于 HX711 是低速转换 AD 芯片，实际有效位数为 17 位左右，内部已带 50Hz 抑制电路，外加屏蔽罩并无明显实际效果（客户可以自己测试比较）。要想得到稳定的结果，重要的是做好数字滤波。贴片元件可以机器焊接，但插针必须手工补焊。由于没插针不好测试，所以本店模块都已焊好插针，尽管这样做增加了工时成本，但确保了每个模块都是测试过的。

2.6.6 注意事项：

模块工作电压：4.8-5.5v

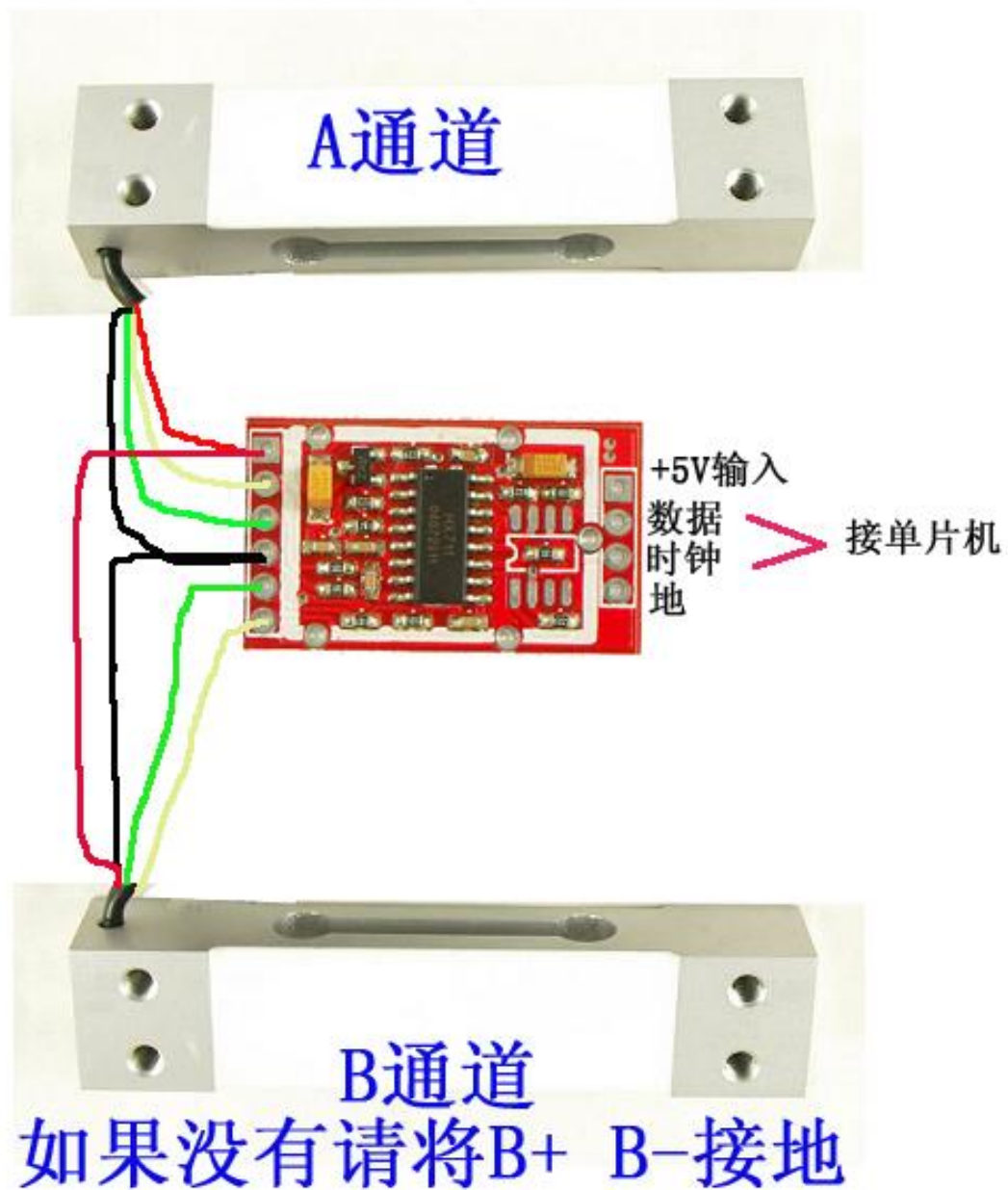
典型电流 1.6mA

体积：长 2.9cm * 宽 1.7cm * 高 0.4cm

- 带金属屏蔽，强抗干扰，预留 MCU(STC15F104)位置，可自行升级二次开发。
- 两路可选择差分输入
- 片内低噪声可编程放大器，可选增益为 32、64 和 128
- 片内稳压电路可直接向外部传感器和芯片内 A/D 转换器提供电源
- 片内时钟振荡器无需任何外接器件，必要时也可使用外接晶振或时钟
- 上电自动复位电路
- 简单的数字控制和串口通讯：所有控制由管脚输入，芯片内寄存器无需编程
- 可选择 10Hz 或 80Hz 的输出数据速率
- 同步抑制 50Hz 和 60Hz 的电源干扰
- 耗电量（含稳压电源电路）：

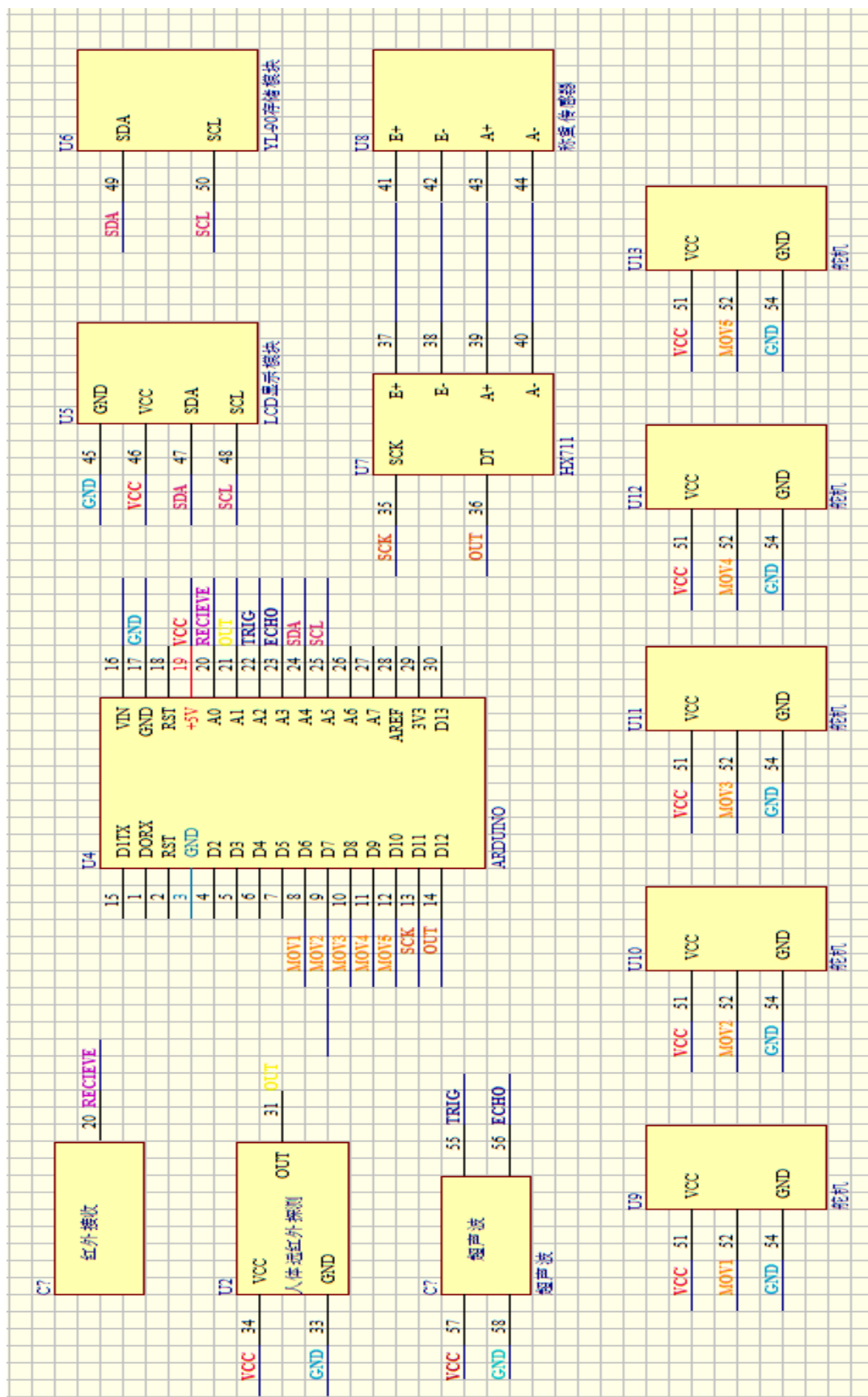
典型工作电流：< 1.7mA，断电电流：< 1 μ A

- 工作电压范围：2.6 ~ 5.5V
- 工作温度范围：-20 ~ +85 $^{\circ}$ C



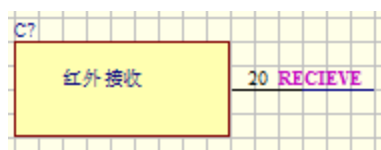
3. 系统硬件电路设计

整体电路原理图图设计如下：



3.1. 红外接收电路:

原理图如下



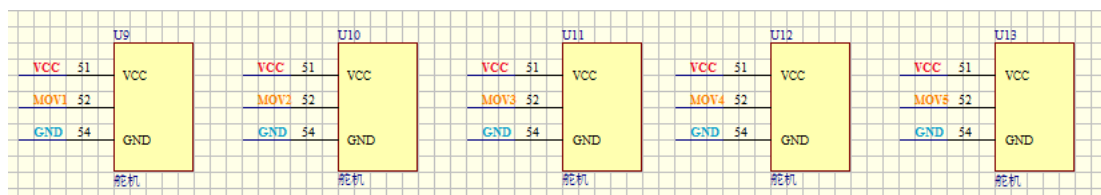
说明:

红外接收器通过红外感光设备对外界物体进行识别，同时在“RECIEVE”管脚与“ARDUINO”单片机上“RECIEVE”管脚相连，从而将外界情况通过红外接收传给主单片机。

因为舵机电磁干扰比较严重，如果红外线接收头与单片机连线过长会产生信号乱码的现象，所以我们直接将红外线接收头焊在了电路板上。

3. 2. 舵机控制电路:

原理图如下:

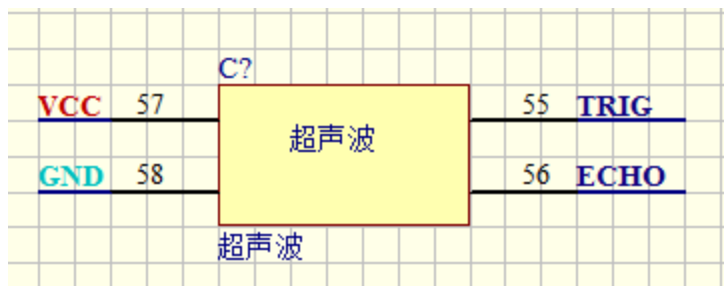


说明:

机械手臂的运作是靠五个舵机的旋转来完成的，每个舵机有三个管脚，“VCC”是正极，接 5V 的电压；“MOV”是信号线，五个舵机的信号线分别与单片机的上对应管脚相连接；GND 是接地线。我们采用的是模拟舵机，需要不断发送控制信号，20ms 定时中断，少于 18ms 无法运行，大于 20ms 会颤抖。

3. 3. 超声波检测模块电路:

原理图如下:

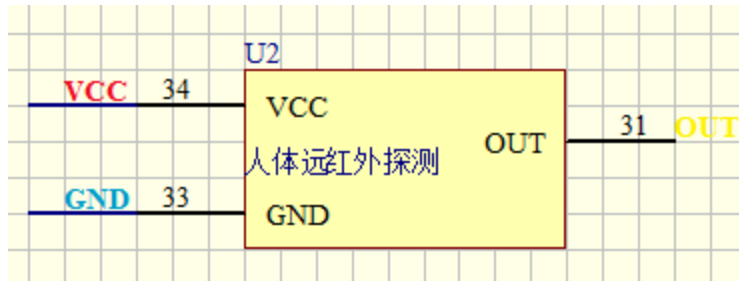


说明：

模块内已经默认取多个数据的平均值，所以在程序中只要计算一次 ECHO 信号脉冲的宽度，就可以达到比较理想的效果。超声波在测量错误时会输出“0”因此在程序中要排除掉这种情况才不至于误认为撞到物体。如果想提高精确度，可以用夏普的红外测距（利用角度测距的原理）。电路图中，“VCC”是正极电源，“TRIG”是控制端，“ECHO”是接收端，“GND”是接地端。

3.4.人体远红外检测模块电路：

原理图如下：

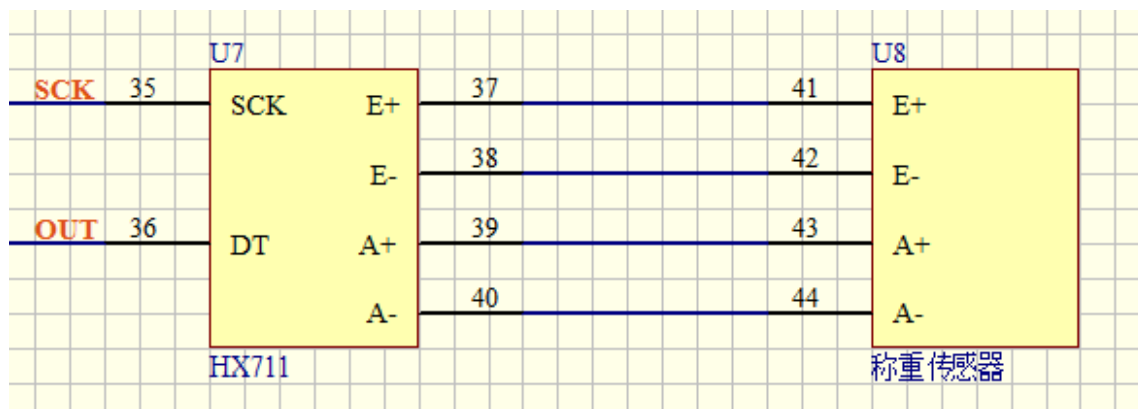


说明：

人体远红外探测是用来对人体进行探测从而发送信号使机械手臂停留在人体感应区域，不过红外探测只能探测到移动的人体，对静止的人体无法感应。分为正极“VCC”，接 5V 电压；输出端“OUT”将信号传输给单片机控制机械臂停留位置；GND 为接地。由于人体红外反应时间比较长，所以在写程序时需要考虑这一点。

3.5.测力模块电路：

原理图如下：



说明:

测力模块分为 HX711 电子称与称重传感器两个部分。测力模块常常会出错输出“0”，因此在程序中要避免这一点导致的误差。HX711 输出的数据是 64 位的，储存会比较麻烦，因此在对精度要求不高的情况下，把它转换成十

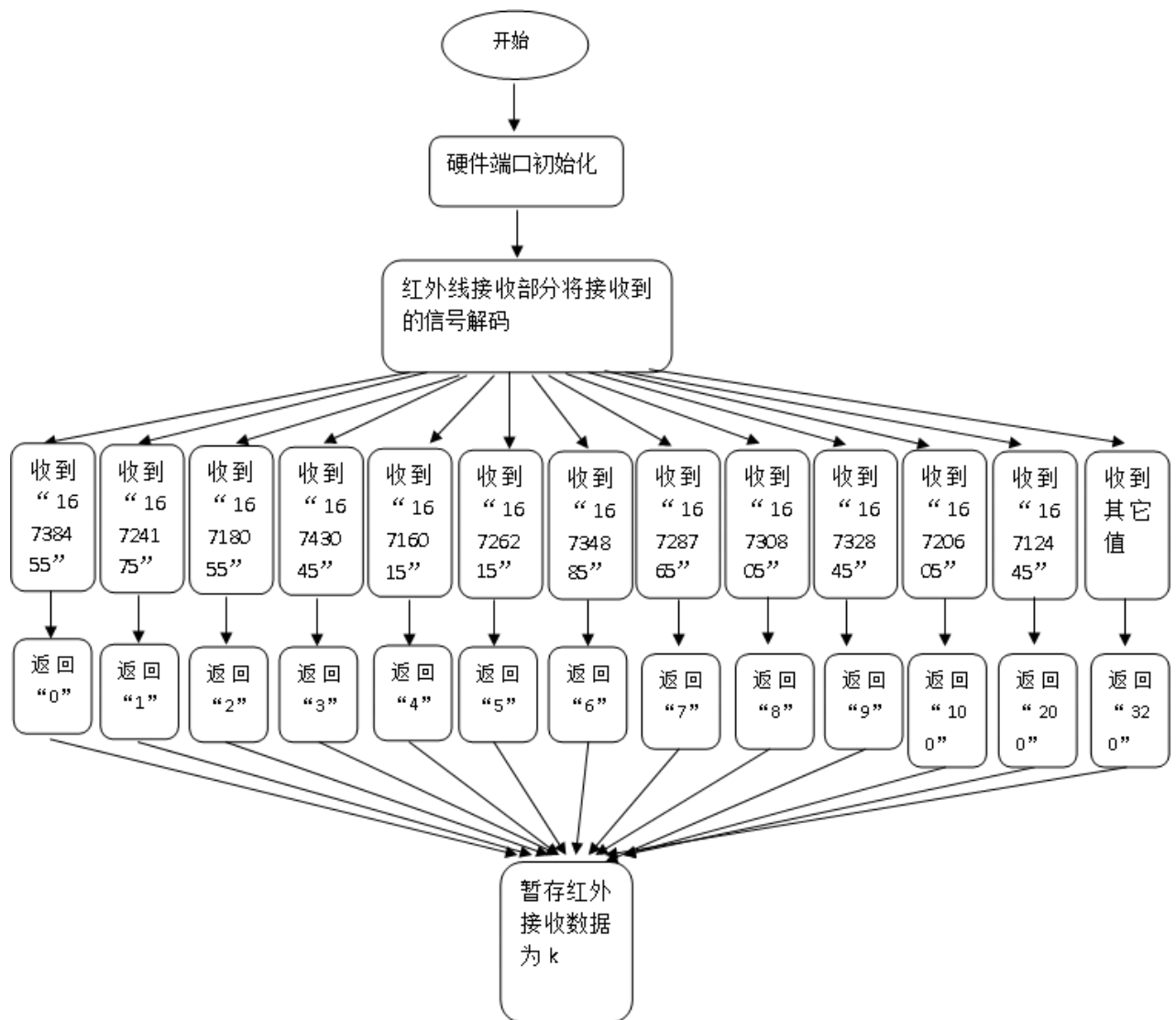
4. 系统各模块软件设计

4.1. 红外接收模块:

程序原理:

红外线遥控系统由接收和发射两部分组成。其中包括发射部分包括键盘矩阵、编码调制、LED 红外发送器等；而接收部分则包括光电转换放大器、解调、解码电路等。

Arduino 中提供了红外线 IRemote 函数库，所以在编程时可以直接调用这个函数库。



主要程序代码：

```
#include <IRremote.h>
```

```
int RECV_PIN = 2; //定义红外接收器的引脚为 2
```

```
unsigned long int rec; //暂存红外接收数据
```

```
IRrecv irrecv(RECV_PIN); //定义红外接收引脚
```

```
decode_results results;
```

```
int HWX(unsigned long int k); //红外转换程序
```

```
void setup() //硬件端口初始化
```

```
{  
  
    Serial.begin(9600);//打开串口  
  
    irrecv.enableIRIn(); // 初始化红外接收器  
  
}  
  
void loop() {  
  
int HWX(unsigned long int k)//红外信号转换程序  
  
    {  
  
        switch (k) {  
  
            case (16738455):                //0  
  
                { return 0;  
  
                    break;  
  
                }  
  
            case (16724175):                //1  
  
                { return 1;  
  
                    break;  
  
                }  
  
            case (16718055):                //2  
  
                { return 2;  
  
                    break;  
  
                }  
  
            case (16743045):                //3  
  
                { return 3;  
  
                    break;  
  
                }  
  
        }  
  
    }  
  
}
```

```
case (16716015):           //4

    {   return 4;

        break;

    }

case (16726215):           //5

    {   return 5;

        break;

    }

case (16734885):           //6

    {   return 6;

        break;

    }

case (16728765):           //7

    {   return 7;

        break;

    }

case (16730805):           //8

    {   return 8;

        break;

    }

case (16732845):           //9

    {   return 9;

        break;

    }
```

```
        case (16720605):                //left

        {   return 100;

            break;

        }

        case (16712445):                //right

        {   return 200;

            break;

        }

        default: return 320;

    }

    //irrecv.resume(); // 接收下一个值

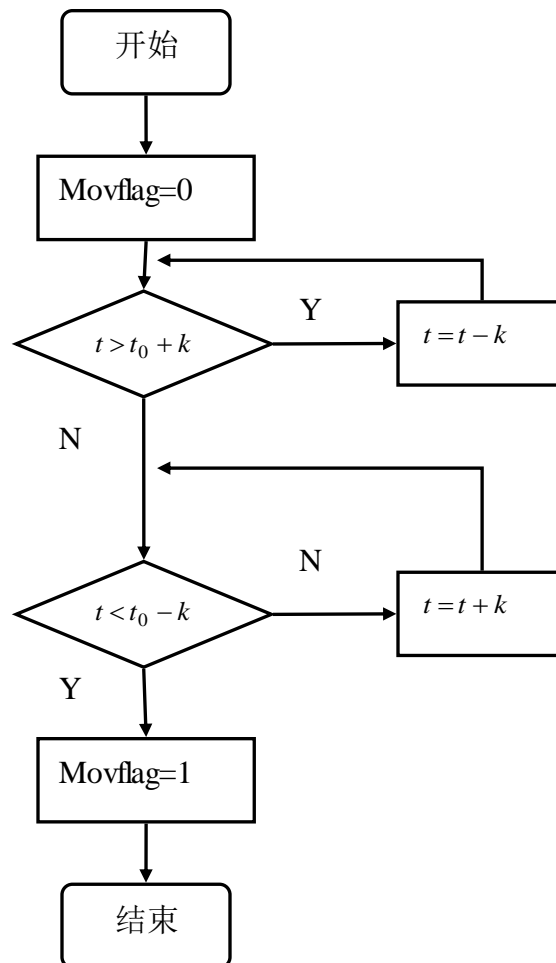
}

}
```

4.2.MG995 模拟舵机:

程序原理:

舵机需要每隔 20ms 发送一次脉冲信号，角度与脉冲宽度成正比，因此只要改变发送脉冲的宽度就能改变舵机角度。为了控制舵机的速度（避免舵机转动过快），程序中分多次以小角度变化累计至所需要角度。并且时刻记录当前的脉冲宽度，以便于程序知道舵机现在所在位置并判断是否进行下一步动作。



主要程序代码：

```

void servopulse(int i,int ilength)//脉冲-引脚输出子程序
{
    digitalWrite(servopin[i],HIGH);//将舵机接口电平至高
    delayMicroseconds(ilength);//延时脉宽值的微秒数
    digitalWrite(servopin[i],LOW);//将舵机接口电平至低
}
  
```

```

void stmovcontrol(float angl1,float angl2,float angl3,float angl4,float
angl5)//依次输出各个舵机的脉冲信号长度
  
```

```

{

float  ipulse[5],iangle[]={angl11,angl12,angl13,angl14,angl15};

g_movflag=1;

for(int i=0;i<5;i++)

{

    ipulse[i]=iangle[i]*10+500;

    if(ipulse[i]<=pulsesmin[i]){ ipulse[i]=pulsesmin[i];}

    if(ipulse[i]>=pulsesmax[i]){ ipulse[i]=pulsesmax[i];}

    if(ipulse[i]>(pulsesnow[i]+movspeed)){pulsesnow[i]+=movspeed;g_movflag=0;}

    else
    if(ipulse[i]<(pulsesnow[i]-movspeed)){pulsesnow[i]-=movspeed;g_movflag=0;}

    else{pulsesnow[i]=ipulse[i];}

    servopulse(i,pulsesnow[i]);

    }//使角度缓慢递进的达到预定值

}

```

4.3.EEPROM 储存模块:

程序原理:

数据写入:

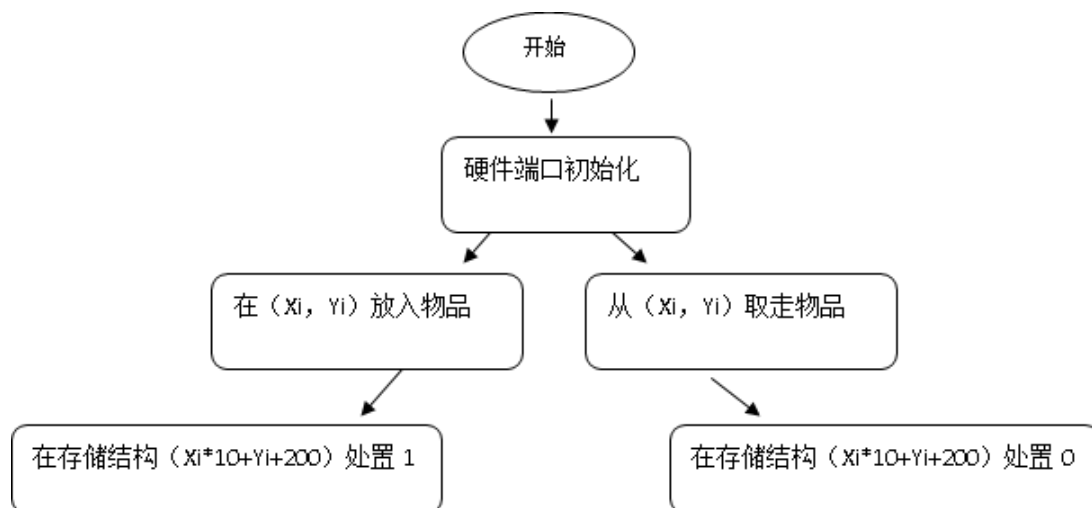
➤货柜物品重量的写入:

将货架的行数定为 x_i ，列数定为 y_i ，写入货架 (x_i, y_i) 处摆放物品的重量，

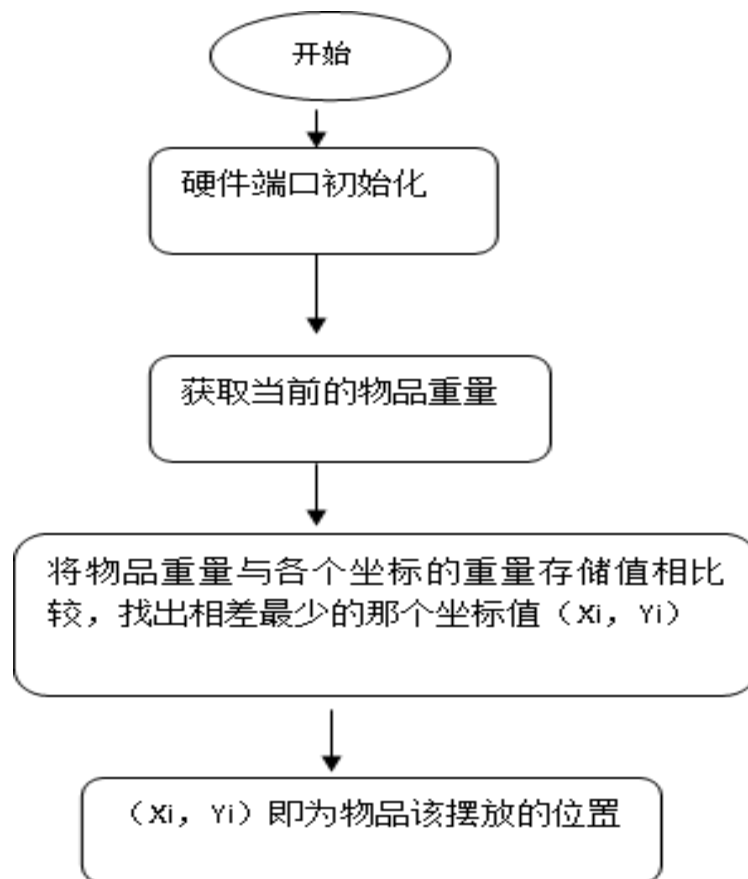
(x_i, y_i) 位置重量的前 8 位置 \longrightarrow 存储结构位置为 $(20 \times x_i + 2 \times y_i)$ 的地方

(x_i, y_i) 位置重量的后 8 位置 \longrightarrow 存储结构位置为 $(20 \times x_i + 2 \times y_i + 1)$ 的地方

➤ 货柜位置是否放有物品的标记:



➤ 识别重物类型:



主要程序代码：

```
void writeweight(byte xi ,byte yi,unsigned int iweight16)//记录 xi,yi 位置物品的重量
{
    if(xi<10&&yi<10)
    {
        EEPROM.write(xi*20+2*yi, iweight16>>8);
        EEPROM.write(xi*20+2*yi+1, iweight16);
    }
}

unsigned int getweight(byte xi ,byte yi)//获取 xi,yi 位置物品的重量
{

```



```
    unsigned int iread;

    if(xi<10&&yi<10)

    {

        iread=(EEPROM.read(xi*20+2*yi)<<8)+EEPROM.read(xi*20+2*yi+1);

        return iread;

    }

}
```

void weightdataclr()//清空储存器

```
{

    for (int iclr = 0; iclr < 300; iclr++)EEPROM.write(iclr, 1);

}
```

byte getposition(unsigned int iiweight16)//获取当前重量的物品的位置 xi,yi

```
{

    unsigned int xi,yi,itemp,difmin=300;

    for(int i=0;i<10;i++)

    {

        for(int j=0;j<10;j++)

        {

            itemp=abs(iiweight16-getweight(i,j));

            if(itemp<difmin){difmin=itemp;xi=i,yi=j;}

        }

    }

}
```

```
    if(difmin<g_res){return xi*10+yi;}

    else{return 0xFF;}

}

byte ifbethere(byte xi,byte yi)//获取当前位置 xi,yi 是否已经有物品存在

{

    return EEPROM.read(xi*10+yi+200);

}


void puttothere(byte xi,byte yi)//将 ifbewhere(xi,yi)置 1

{

    EEPROM.write(xi*10+yi+200,1);

}

void getfromthere(byte xi,byte yi)//将 ifbewhere(xi,yi)置 0

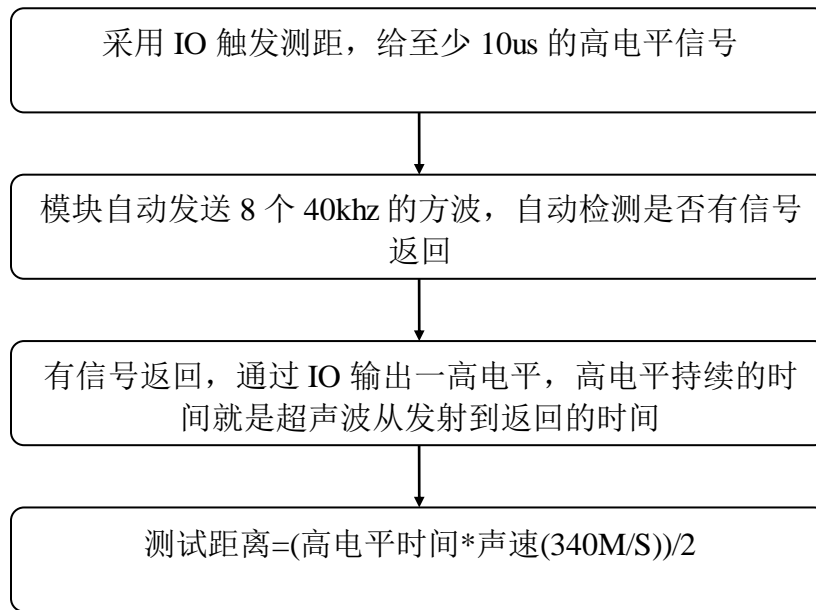
{

    EEPROM.write(xi*10+yi+200,0);

}
```

4.4.超声波测距模块：

程序原理：



主要程序代码：

//CSB-超声波测距子程序-----

const int TrigPin = 4; //定义超声波传感器接口

const int EchoPin = 5;

float distance;

float measure();

float measure()//超声波测距程序

{

float distance;

digitalWrite(TrigPin, LOW);

delayMicroseconds(2);

```

    digitalWrite(TrigPin, HIGH);

    delayMicroseconds(10);

    digitalWrite(TrigPin, LOW);

    // 检测脉冲宽度，并计算出距离

    distance = pulseIn(EchoPin, HIGH,50000) / 58.00;


    return distance;

}

//当前方有障碍物时，会停在原地，直到障碍物被拿走，小车才继续前进

case (110)://look if bump to sometings

    if(measure()<10&&measure()>1)

    {

        int ia[5];

        for(int i=0;i<5;i++)

        {

            ia[i]=pulsetoangle(pulsesnow[i],i);

        }

        setstmov(ia[0],ia[1],ia[2],ia[3],ia[4]);

        g_mode=400;//>>>turn to the already mode 400

    }


    if(g_movflag==1)

    {

```

```
g_movflag=0;

g_mode=400;

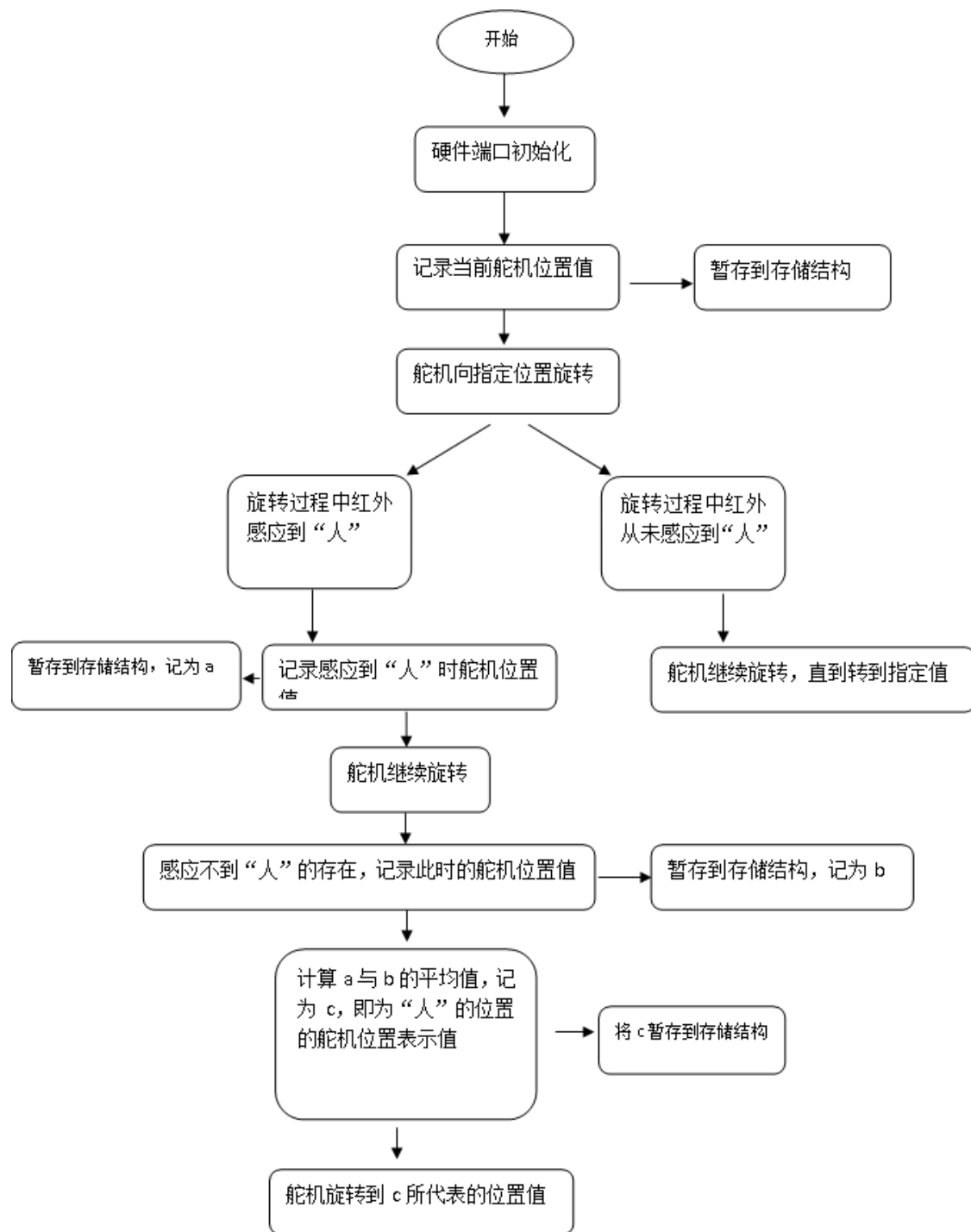
}

break; // >>> turn to the already mode 400
```

4.5. 人体远红外检测模块:

程序原理:

HC-SR501 是基于红外线技术的自动控制模块



主要程序代码：

```
case (106)://turn around the hand
```

```
if(g_movflag==1)
```

```
{
```

```
        g_movflag=0;

        atstmov(180,180,0,0,0);

        g_mode+=1;
    }

    break;

case (107)://look wether a man here

    if(RTGY())

    {

        g_rtgy_start0=pulsesnow[0];

        g_rtgy_start1=pulsesnow[1];

        g_mode+=1;
    }

    if(g_movflag==1)//if there is no man here,turn to the initial position

    {

        g_movflag=0;

        setonemov(0,155);

        setonemov(1,18);

        g_mode+=2;
    }

    break;

case (108): //look where the man is

    if(!RTGY()||g_movflag==1)

    {

        g_movflag=0;
```

```
int ia0,ia1;

g_rtgy_end0=pulsesnow[0];

g_rtgy_end1=pulsesnow[1];

ia0=pulsetoangle((g_rtgy_start0+g_rtgy_end0)/2,1);

ia1=pulsetoangle((g_rtgy_start1+g_rtgy_end1)/2,1);

setonemov(0,ia0);

setonemov(1,ia1);

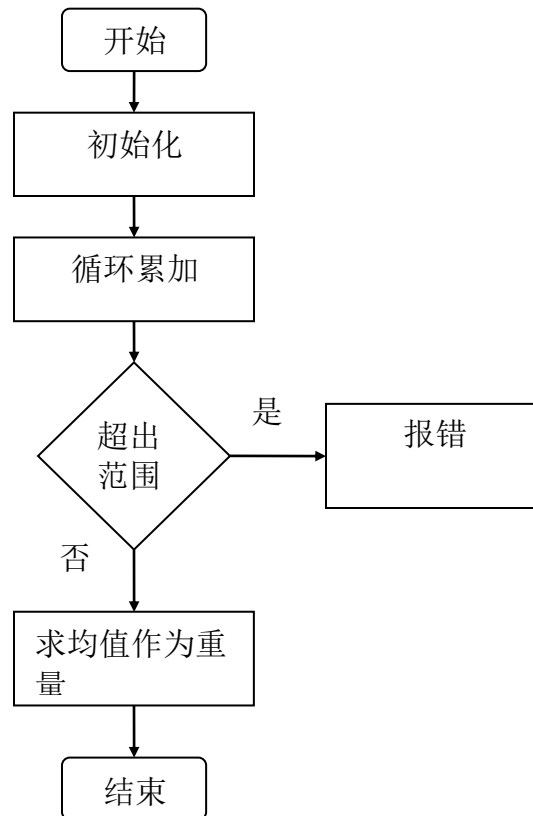
g_mode+=1;

}

break;
```

4.6.测力模块（应力变阻片+ HX711 24 位模数转换模块）：

程序原理：



主要程序代码：

//HX711-HX711 称重子程序-----

HX711 hx(6, 7); //定义 sck、dout 接脚

int hx711_weighting(); //获取当前物品重量

int hx711_weighting() //获取当前物品重量

{

double sum = 0;

int iwe;

for (int i = 0; i < 3; i++) //循环的越多精度越高，当然耗费的时间也越多

```
    sum += hx.read();//累加

    if(sum<10000000&&sum>-10000000)

    {

        iwe=sum/1000; //求平均值进行均差

    }

    else

    {

        iwe=-10000;

    }

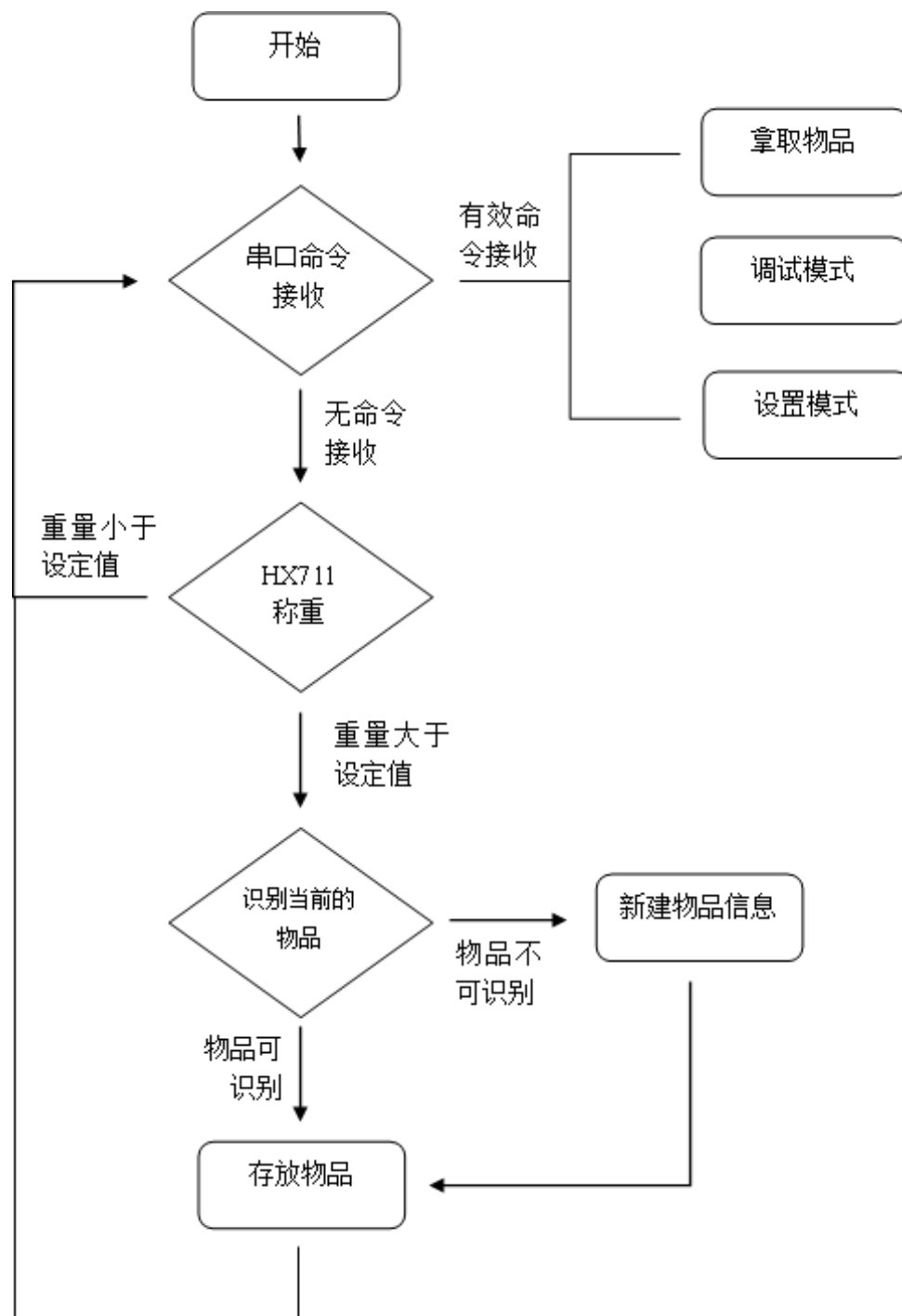
    return iwe;

}
```

5. 系统综合软件设计

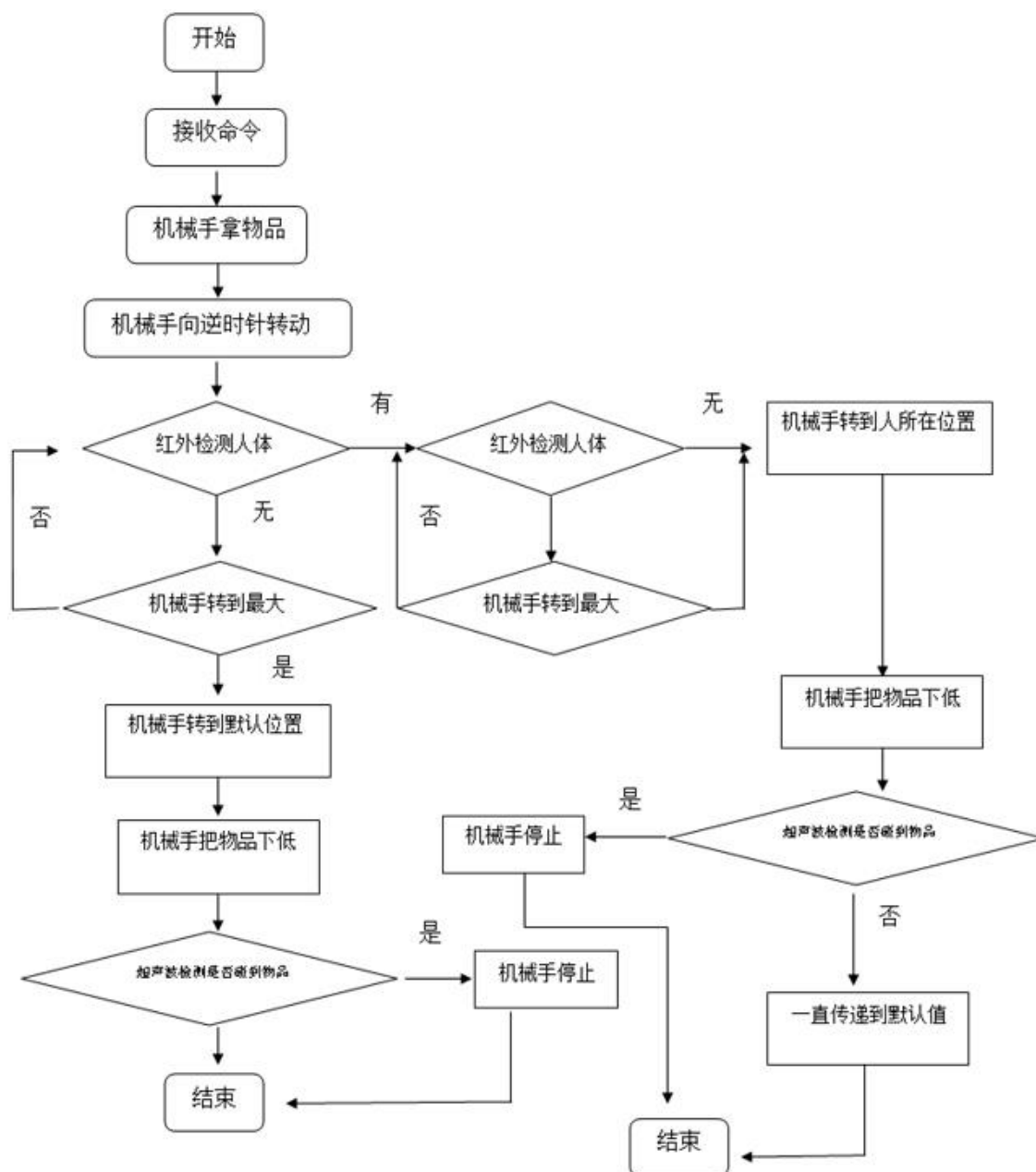
5.1.总体软件流程设计:

系统的主要运行流程如下图所示，系统开始后串口开始接收命令信号，操作者如果通过遥控器发送命令给系统，系统便可以进入到拿取物品模式/调试模式/设置模式，如果没有发送指令，则系统自动称量物品的重量并判断机械手上面是否有物品存在，如有物品则开始存放物品或者新建物品信息，否则就重新开始接收串口命令。



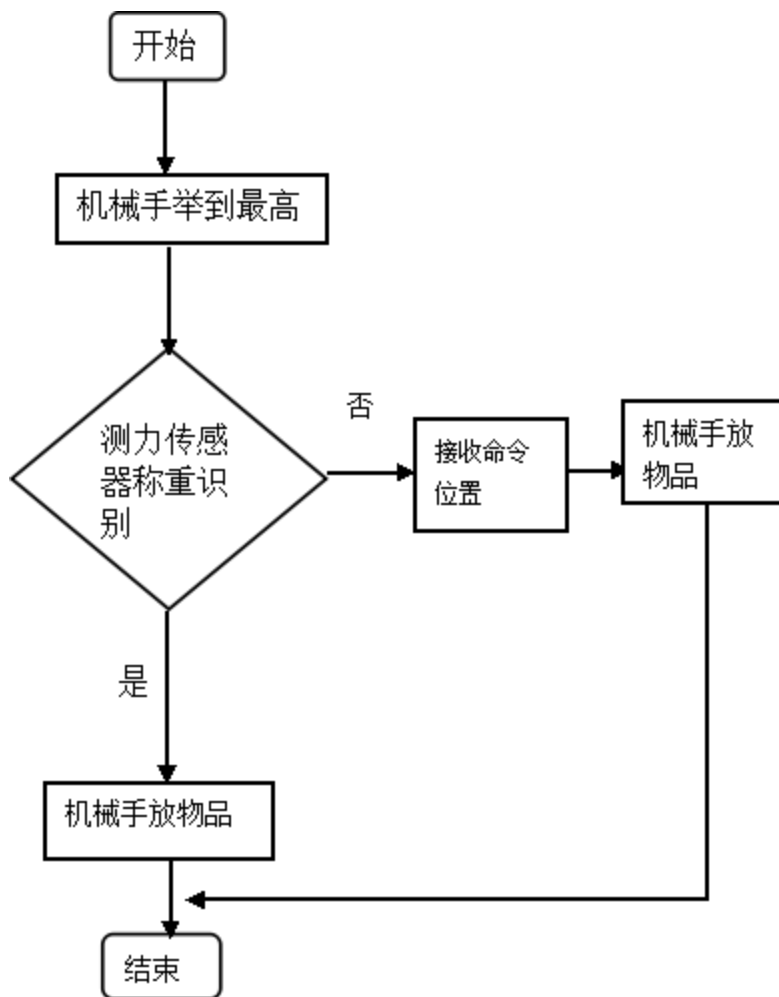
5.2. 拿取物品模式软件流程设计:

软件开始时接收命令（拿取的物品的坐标），遥控器上按下“确定”后，开始取物品，取完物品之后检测人体所在的位置，接着把物品传递给操作者（当机械手检测到快要撞到物体时停止运动）。当操作者从机械手上拿起物品时，系统又回到初始位置。具体软件流程图如下：



5.3.存放物品模式软件流程设计：

人为将物品放上机械手，按开始按钮，操控机械手举到最高，然后测力感应器进行称重识别，如果满足要求，机械手放置物品，结束；如果不满足称重要求，等待接收命令位置，然后再放置物品，结束。



6. 系统的应用及创新

我们的系统主要设计供维修人员或一些小型的小工具架上应用。它的优势是可以自动将物品放到其应该放的位置而不需要花费时间去寻找。并且通过它可以减少人力劳动从而提高人的工作效率。我们的机械臂主要应用在辅助人工操作领域：

- 1、手术台上，主刀医师可以通过口头指令让机械臂为其取手术刀；
- 2、机械制造时作为助手送取工具，将操作者所需的工具从工具台取出并在用完后自动送还摆放整齐；
- 3、可以改进成智能家居整理机器人，实现把主人随便乱放的东西放到该放的位置；
- 4、将作品加上定位和移动系统，可以在智能仓储系统中作为智能搬运机器人来搬运货物和摆放工具；
- 5、它可以在危险的场合下工作，如军用品的装卸、危险品及有害物质的搬运等；
- 6、在电子行业中用来装配印制电路板。进行程序改进使机械臂不停的重复来讲

原件摆放在电路板上进行焊接；

7、机床加工工件的装卸。将机械臂承受力量加大，将所需部件安放在固定位置上进行拼装和焊接。

如果将我们的作品稍加改进，我们也可以应用到更广阔的范围。我们可以将它改进成智能家居整理机器人，可以实现把主人随便乱放物品放到该放的位置。还有，现在的智能仓储系统依然需要很多人力来搬运东西和工具，如果将我们的作品加上定位和移动系统，作为一个智能搬运的机器人，将会得到很好的应用。

附录

1. 主要控制单片机程序：

```
#include <MsTimer2.h>

#include <EEPROM.h>

#include <IRremote.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <HX711.h>


void mydelay(int ta)

{

    int i,j;

    for(i=0;i<ta;i++)for(j=0;j<16000;j++);

}


//GLOBAL-全局子程序和变量-----

int t1_count=1000;//用于延時計时的定時計数器变量
```

```

int yi_mode=0,g_pause=0;//yi_mode:机械手举起和放下的步骤;g_pause:停止标志位

int mode700_movi,mode700_angl;//手动调节模式(g_mode=700+)的舵机控制变量--mode700_movi:舵机号;mode700_angl:舵机脉冲长度;

int g_mode=0,g_xi,g_yi,g_weightcount;//g_mode:当前运行步骤(以 100 为一个
大步骤, 后面为模式中的详细步骤);g_xi/g_yi:当前取放物品的位置号;g_weightnow:当前物品重量;

int g_weightnow;

int g_numi[]={-1,-1},g_lcd;//lcd 当前取放物品的位置号显示辅助变量

int g_setangle[]={155,18,10,0,0},g_movflag;//+++++++当前设置的舵机脉冲长度(进过转换后的)+++++++

int g_rtgy_start0=0,g_rtgy_end0=180,g_rtgy_start1=0,g_rtgy_end1=180;//机器人检测人体方位的辅助变量

const int g_res=350;//称重分辨率

void G_COM();//串口命令接收判断程序

void G_CONTROL();//舵机控制总程序(包括物品称重识别\人体感应\超声波测距)

void G_LCD();//LCD 显示程序

//void G_DCMOV();//电机控制程序(预留)

void G_STMOV();//舵机脉冲周期性发送程序,用于维持舵机运转

void G_PAUSE();//令舵机动作暂停

void LCDinit();//LCD 显示初始化

//GLOBAL_MOV_CONTROL-全局舵机控制子程序
-----

const int Ylift[3][8][3]={

```

```

        { {90,100,150}, {90,100,100},
{90,36,100}, {90,36,111}, {90,63,111}, {90,63,142}, {90,97,142},
{10,100,150} },

        { {10,100,150}, {10,100,150},
{10,100,150}, {10,100,150}, {10,100,150}, {10,100,150}, {10,100,150},
{10,100,150} },

        { {10,100,20}, {10,63,20}, {25,63,20},
{25,63,58}, {10,63,58}, {10,85,85}, {10,100,150}, {10,100,150} }

    };//

const int Ydown[3][8][3]={

    { {60,100,150},{60,90,110},{60,74,100},{70,74,100},{70,53,100},{70,53,65},
{70,96,65},{10,96,150} },

        { {10,100,150}, {10,100,150},
{10,100,150}, {10,100,150}, {10,100,150}, {10,100,150},
{10,100,150} },

    { {10,100,70},{10,35,70},{27,35,70},{27,35,20},{27,105,20},{27,105,150},
{10,100,150},{10,100,150} }

    };

void atstmov(int angl1,int angl2,int angl3,int angl4,int angl5);//改变各个舵机脉冲长度,正数为增加,负数为减少

void setstmov(int angl1,int angl2,int angl3,int angl4,int angl5);//设置各个舵机脉冲长度

void set234mov(int angl3,int angl4,int angl5);//设置 2,3,4 号舵机(最上面的 3 个舵机)的脉冲长度

void atonemov(int _movi,int _angl);//改变单个舵机的脉冲长度(movi:舵机号/angl:改变脉冲长度的值)

void setonemov(int movi,int angl);//设置单个舵机的脉冲长度(movi:舵机号/angl:脉冲长度)

void positturn(int xi);//机器人转到横坐标为 xi 的物品的程序

```



```
void positlift(int yi);//机器人拿起纵坐标为 yi 的物品程序

void Y0_lift();

void Y1_lift();

void Y2_lift();

void positdown(int yi);//机器人放到纵坐标为 yi 的物架程序

void Y0_down();

void Y1_down();

void Y2_down();

//STMOV-底层舵机控制子程序-----

int servopin[]={8,9,10,11,12};//定义舵机接口数字接口

const int movspeed=6;

const int pulsesmin[]={800,600,600,500,700};//定义各个舵机实际最小脉宽

const int pulsesmax[]={2000,2000,1500,1600,2000};//定义各个舵机实际最大脉宽

int pulsesnow[]={2050,800,600,500,700};//定义当前各个舵机实际脉宽,单位为微妙

void movinit();//舵机初始化程序

void servopulse(int i,int ilength);//脉冲-引脚输出子程序

void stmovcontrol(float angl11,float angl12,float angl13,float angl14,float angl15);//依次输出各个舵机的脉冲信号长度

int pulsetoangle(int pulse_temp,int ik);//实际脉冲信号长度和程序中转化后的脉冲长度的转换程序

//CSB-超声波测距子程序-----
```

```
const int TrigPin = 4; //定义超声波传感器接口

const int EchoPin = 5;

float distance;

float measure();

//EEPROM-存储器储存读取、重量位置判断子程序
-----

int xtemp; //暂存变量

void writeweight(byte xi, byte yi, int iweight16); //记录 xi, yi 位置物品的重量

int getweight(byte xi, byte yi); //获取 xi, yi 位置物品的重量

void weightdataclr(); //清空存储器

byte getposition(int iweight16); //获取当前重量的物品的位置 xi, yi

byte ifbewhere(byte xi, byte yi); //获取当前位置 xi, yi 是否已经有物品存在

void puttothere(byte xi, byte yi); //将 ifbewhere(xi, yi)置 1

void getfromthere(byte xi, byte yi); //将 ifbewhere(xi, yi)置 0

//HWX-红外接收转码子程序-----

int RECV_PIN = 2; //定义红外接收器的引脚为 2

unsigned long int rec; //暂存红外接收数据

IRrecv irrecv(RECV_PIN); //定义红外接收引脚

decode_results results;

int HWX(unsigned long int k); //红外转换程序

//LCD-LCD 显示子程序-----
```

```
int itemp[17];

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars
and 2 line display

void lcd_clrdisp(int line,char *p);//第 line 行先清除后显示数据

void lcd_disp(int line,char *p);//在第 line 行显示数据

void lcd_clr(int line);//清除第 line 行的显示数据

void lcd_dispint(int line,unsigned long int int16);//在显示器上显示数据


//HX711-HX711 称重子程序-----

HX711 hx(6, 7);//定义 sck、dout 接脚

int hx711_weighting();//获取当前物品重量


//RTGY-人体探测子程序-----

int RTGY_PIN = 3;//设置人体感应模块引脚

byte RTGY();//获取当前是否有人存在


//Main function//////////////////////////////////////

void flash()//每 20ms 执行一次此函数

{

    if(g_pause)

    {

        G_PAUSE();//舵机暂停处理程序

    }

}
```

```
else

{

    G_STMOV();//舵机脉冲发送程序

}

if(t1_count<5000){t1_count+=20;}//timer1,定时计数器 1（每 20ms 数值+20）

}
```

void setup()//硬件端口初始化

```
{

    pinMode(RTGY_PIN, INPUT);

    pinMode(servopin[0],OUTPUT);//设定舵机接口为输出接口

    pinMode(servopin[1],OUTPUT);

    pinMode(servopin[2],OUTPUT);

    pinMode(servopin[3],OUTPUT);

    pinMode(servopin[4],OUTPUT);

    pinMode(TrigPin, OUTPUT);

    pinMode(EchoPin, INPUT);

    irrecv.enableIRIn(); // 初始化红外接收器

    lcd.init();//显示屏初始化

    lcd.backlight();//显示屏 led 灯打开

    LCDinit();//显示显示初始数据

    Serial.begin(9600);//打开串口

    MsTimer2::set(20, flash); // 20ms period

    MsTimer2::start();//定时中断开启
```

```

}

void loop()//主函数
{
// movinit();//舵机初始化

while(1)

{

G_COM();//串口命令接收判断程序

G_CONTROL();//总控制程序

G_LCD();//LCD 显示程序

}

}

//Subs//////////////////////////////////////

//GLOBAL-----
-----

//comunicaton sub-----

void G_COM()//串口命令接收判断程序
{

if (Serial.available())

{

byte recnum;

```

```
recnum=Serial.read()-'0';

if(recnum==('P'-'0'))g_pause=!g_pause;

switch (g_mode)

{

//start mode,already for man to control-----

case (3):

    if(0<=recnum&&recnum<10&&g_lcdi<2)

    {

        g_numi[g_lcdi]=recnum;

        g_lcdi+=1;

    }

    if(recnum==('B'-'0')&&g_lcdi>0)

    {

        g_lcdi-=1;

        g_numi[g_lcdi]=-1;

    }

    if(recnum==('G'-'0'))

    {

        if(g_numi[0]==9&&g_numi[1]==9)

        {

            g_mode=600;//set mode

        }

        else if(g_numi[0]==8&&g_numi[1]==8)

        {
```

```
        g_mode=700;//manual mode

    }

    else if( ifbethere(g_num1[0],g_num1[1]) )

    {

        g_mode=100;g_xi=g_num1[0];g_yi=g_num1[1];

    }

    else

    {

        g_mode=500;//the thing isn't there

    }

}

if(recnum=='P')

{

    g_pause=!g_pause;

    g_mode=200;

}

break;

//get the things in position (g_xi,g_yi)-----

case (100):

    break;

//put the things to position (g_xi,g_yi)-----

case (200):

    break;

//find the new things,and write the weight to (xi,yi)-----
```

```
case (300):

    if(0<=recnum&&recnum<10&&g_lcdi<2)

    {

        g_numi[g_lcdi]=recnum;

        g_lcdi+=1;

    }

    if(recnum=='B')

    {

        g_lcdi-=1;

        g_numi[g_lcdi]=-1;

    }

    if(recnum=='G')

    {

        // !ifbethe(g_numi[0],g_numi[1])
        ++++++
        ++++++

        if(1)

        {

            g_mode=301;

            g_xi=g_numi[0];

            g_yi=g_numi[1];

        }

        else

        {

            g_mode=503;//the thing is all raedy there
```



```
    }  
  
    }  
  
    break;  
  
//the machine is already for man to pick up the things-----  
  
    case (401):  
  
        if(recnum=='G'-'0')  
  
        {  
  
            g_mode=0;  
  
        }  
  
        break;  
  
//error mode-----  
  
    case (500):  
  
        if(recnum=='G'-'0')  
  
        {  
  
            g_mode=0;  
  
        }  
  
        break;  
  
    case (503):  
  
        if(recnum=='G'-'0')  
  
        {  
  
            g_mode=300;  
  
        }  
  
    case (504):  
  
        if(recnum=='G'-'0')
```

```
    {  
  
        g_mode=0;  
  
    }  
  
//set mode-----  
  
case (600):  
  
    break;  
  
case (601):  
  
    if(recnum==1)//clean all of the data  
  
    {  
  
        weightdataclr();  
  
        LCDinit();  
  
        g_mode=0;  
  
    }  
  
    else if(recnum==2)//inite the machine  
  
    {  
  
        movinit();  
  
        irrecv.enableIRIn(); // 初始化红外接收器  
  
        LCDinit();  
  
        g_mode=0;  
  
    }  
  
    else if(recnum=='B'-'0')  
  
    {  
  
        LCDinit();  
  
        g_mode=0;
```

```
    }

    break;

//manual mode-----

    case (700):

        break;

    case (701):

        switch(recnum)

        {

            case (0):

                mode700_movi=0;

                mode700_angl=-1;

                atonemov(mode700_movi,mode700_angl);

                break;

            case (1):

                mode700_movi=0;

                mode700_angl=1;

                atonemov(mode700_movi,mode700_angl);

                break;

            case (2):

                mode700_movi=1;

                mode700_angl=-1;

                atonemov(mode700_movi,mode700_angl);

                break;

            case (3):
```

```
mode700_movi=1;

mode700_angl=1;

atonemov(mode700_movi,mode700_angl);

break;

case (4):

mode700_movi=2;

mode700_angl=-1;

atonemov(mode700_movi,mode700_angl);

break;

case (5):

mode700_movi=2;

mode700_angl=1;

atonemov(mode700_movi,mode700_angl);

break;

case (6):

mode700_movi=3;

mode700_angl=-1;

atonemov(mode700_movi,mode700_angl);

break;

case (7):

mode700_movi=3;

mode700_angl=1;

atonemov(mode700_movi,mode700_angl);

break;
```

```
case (8):

    mode700_movi=4;

    mode700_angl=-1;

    atonemov(mode700_movi,mode700_angl);

    break;

case (9):

    mode700_movi=4;

    mode700_angl=1;

    atonemov(mode700_movi,mode700_angl);

    break;

case ('E'-'0'):

    atonemov(mode700_movi,mode700_angl);

    break;

case ('B'-'0'):

    LCDinit();

    g_mode=0;

    break;

default:

    break;

}

break;

default:

    break;

}
```

```
    }  
}  
  
//main control sub-----  
  
void G_CONTROL()//舵机控制总程序(包括物品称重识别\人体感应\超声波测  
距)  
{  
  
    switch (g_mode)  
    {  
  
        //start mode,already for man to control-----  
  
        case (0):  
  
            setstmov(155,18,10,30,20);  
  
            g_mode+=1;  
  
            break;  
  
        case (1):  
  
            if(g_movflag==1)  
            {  
  
                g_movflag=0;  
  
                delay(5000);  
  
                g_weightnow=hx711_weighting();  
  
                delay(2000);  
  
                Serial.println(g_weightnow);//-----  
                -----  
  
                g_mode+=1;  

```

```

    }

    break;

case (3):

    if(g_movflag==1)

    {

        g_movflag=0;

        /*

        int i=0;

        i=hx711_weighting();

        if((g_weightnow-i)>120)

        {

            g_weightnow=i;

Serial.println(g_weightnow);//-----
-----

            g_mode=200;//>>>turn to the start mode 200
+++++
+

        }

        /*/

    }

    break;

//get the things in position (g_xi,g_yi)-----

case (100): //preparation,straight the hand and put it to the highest

    setstmov(155,18,10,0,0); //一柱擎天

```

```
    g_mode+=1;

    break;

case (101): //turn to the right angle (g_xi) first

    if(g_movflag==1)

    {

        g_movflag=0;

        positturn(g_xi);

        g_mode+=1;

    }

    break;

case (102): //prepare for hand to the right posit, avoid things falling down

    if(g_movflag==1)

    {

        set234mov(10,100,150); //低头

        g_movflag=0;

        g_mode+=1;

    }

    break;

case (103): //let the automatic hand to the right posit (g_yi)

    positlift(g_yi);

    //Serial.println(yi_mode);

    if(yi_mode==100)

    {

        yi_mode=0;
```



```
        //getfromthere(g_xi,g_yi);/!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

        g_mode+=1;

    }

    break;

case (104): //put up the things

    if(g_movflag==1)

    {

        g_movflag=0;

        set234mov(10,100,150); //低头

        g_mode+=1;

    }

    break;

case (105)://let the automatic hands in the initial position

    if(g_movflag==1)

    {

        g_movflag=0;

        set234mov(10,0,0); //一柱擎天

        g_mode+=1;

    }

    break;

case (106)://turn around the hand

    if(g_movflag==1)

    {

        g_movflag=0;
```

```
        atstmov(180,180,0,0,0);

        g_mode+=1;

    }

    break;

case (107): //look wether a man here

    if(RTGY())

    {

        g_rtgy_start0=pulsesnow[0];

        g_rtgy_start1=pulsesnow[1];

        g_mode+=1;

    }

    if(g_movflag==1) //if there is no man here, turn to the initial position

    {

        g_movflag=0;

        setonemov(0,155);

        setonemov(1,18);

        g_mode+=2;

    }

    break;

case (108): //look where the man is

    if(!RTGY()||g_movflag==1)

    {

        g_movflag=0;

        int ia0,ia1;
```

```
g_rtgy_end0=pulsesnow[0];

g_rtgy_end1=pulsesnow[1];

ia0=pulsetoangle((g_rtgy_start0+g_rtgy_end0)/2,1);

ia1=pulsetoangle((g_rtgy_start1+g_rtgy_end1)/2,1);

setonemov(0,ia0);

setonemov(1,ia1);

g_mode+=1;

}

break;

case (109)://left the things to man

if(g_movflag==1)

{

g_movflag=0;

setonemov(2,10);

setonemov(3,40);

setonemov(4,0);

g_mode+=1;

}

break;

case (110)://look if bump to sometings

if(measure()<10&&measure()>1)

{

int ia[5];

for(int i=0;i<5;i++)
```

```
{

    ia[i]=pulsetoangle(pulsesnow[i],i);

}

setstmov(ia[0],ia[1],ia[2],ia[3],ia[4]);

g_mode=400;//>>>turn to the already mode 400

}


if(g_movflag==1)

{

    g_movflag=0;

    g_mode=400;

}

break;//>>>turn to the already mode 400

//put the things to position (g_xi,g_yi)-----

case (200): //make the hands straight

    setstmov(155,18,10,0,0);//一柱擎天

    g_mode+=1;

    break;

case (201)://weight the things

    if(g_movflag==1)

    {

        g_movflag=0;

        delay(5000);

        int i;
```

```
g_weightnow=hx711_weighting();

//g_weightnow=itemp;

//Serial.println(g_weightnow);//-----
-----

i=getposition(g_weightnow);

if(i!=0xFF)

{

    g_xi=i/10;//get the position (g_xi,g_yi)

    g_yi=i%10;

    writeweight(g_xi ,g_yi,g_weightnow);//correct the weight

    g_mode+=1;

}

else

{

    g_mode=300;//>>>turn to the new things 300

}

}

break;

case (202): //turn to the right angle (g_xi)

    if(g_movflag==1)

    {

        g_movflag=0;

        positturn(g_xi);
```

```
        g_mode+=1;

    }

    break;

case (203): //prepare for the hands put down

    if(g_movflag==1)

    {

        g_movflag=0;

        set234mov(10,100,150); //低头

        yi_mode=0;

        g_mode+=1;

    }

    break;

case (204): //put the automatic hands down

    positdown(g_yi);

    if(yi_mode==100)

    {

        yi_mode=0;

        //puttothere(g_xi,g_yi);//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

        g_mode+=1;

    }

    break;

case (205): //put down the things

    if(g_movflag==1)

    {
```

```
        g_movflag=0;

        set234mov(10,100,150); //低头

        g_mode+=1;

    }

    break;

case (206)://put back the hands

    if(g_movflag==1)

    {

        g_movflag=0;

        set234mov(10,0,0); //一柱擎天

        g_mode+=1;

    }

    break;

case (207)://turn back the hands

    if(g_movflag==1)

    {

        g_movflag=0;

        atstmov(180,180,0,0,0);

        g_mode+=1;

    }

    break;

case (208)://look if there is a man here

    if(RTGY())

    {
```

```
g_rtgy_start0=pulsesnow[0];
```

```
g_rtgy_start1=pulsesnow[1];
```

```
g_mode+=1;//-----++
+++++=
=====
```

```
}
```

```
if(g_movflag==1)//if there is no man here,turn to the initial position
```

```
{
```

```
g_movflag=0;
```

```
setonemov(0,155);
```

```
setonemov(1,18);
```

```
g_mode+=2;
```

```
}
```

```
break;
```

```
case (209): //look where the man is
```

```
if(!RTGY())
```

```
{
```

```
g_movflag=0;
```

```
int ia0,ia1;
```

```
g_rtgy_end0=pulsesnow[0];
```

```
g_rtgy_end1=pulsesnow[1];
```

```
ia0=pulsetoangle((g_rtgy_start0+g_rtgy_end0)/2,1);
```

```
ia1=pulsetoangle((g_rtgy_start1+g_rtgy_end1)/2,1);
```

```
setonemov(0,ia0);
```



```
        setonemov(1,ia1);

        g_mode+=1;

    }

    break;

case (210): //left the hands to man

    if(g_movflag==1)

    {

        g_movflag=0;

        setonemov(2,10);

        setonemov(3,40);

        setonemov(4,0);

        g_mode+=1;

    }

    break;

case (211)://avoid bumping

    if(measure()<10&&measure()>1)

    {

        int ia[5];

        for(int i=0;i<5;i++)

        {

            ia[i]=pulsetoangle(pulsesnow[i],i);

        }

        setstmov(ia[0],ia[1],ia[2],ia[3],ia[4]);

        g_mode=0;//>>>turn to the already mode 400
```

```
    }

    if(g_movflag==1)

    {

        g_movflag=0;

        g_mode=0;

    }

    break;

//find the new things,and write the weight to (xi,yi)-----

    case (300)://

        break;

    case (301):

        writeweight(g_xi ,g_yi,hx711_weighting());//correct the weight

        g_mode=200;//>>>turn to the puting things mode 202

        break;

//the machine is already for man to pick up the things-----

    case (400)://take the things

        if(g_movflag==1)

        {

            g_movflag=0;

            delay(1800);

            g_weightnow=hx711_weighting();

            g_mode+=1;

        }

        break;
```

```
case (401):

    if(abs(g_weightnow-hx711_weighting())>100)

    {

        if(t1_count>980)t1_count=0;

        if(t1_count>300)

        {

            t1_count=1000;

            g_weightnow=hx711_weighting();

            g_mode=0;//>>>turn to the start mode 0

        }

    }

    break;


default:

    break;

}
```

```
//lcd disp sub-----
```

```
void G_LCD()//LCD 显示程序
```

```
{

    switch (g_mode)

    {
```

```
//start mode,already for man to control-----
```

```
case (2):

    lcd_disp(1,"Slect");

    lcd.setCursor(2,2);

    lcd.print("Delete ");

    lcd.setCursor(16,2);

    lcd.print("Go");

    lcd.setCursor(7,3);

    lcd.print("Ready!      ");

    g_mode+=1;//>>>turn to start mode main

    break;

case (3):

    lcd.setCursor(16,1);

    if(g_num1[0]>=0)

    {

        lcd.print(g_num1[0]);

    }

    else

    {

        lcd.print("- ");

    }

    lcd.setCursor(18,1);

    if(g_num1[1]>=0)

    {

        lcd.print(g_num1[1]);
```

```
    }

    else

    {

        lcd.print("- ");

    }

    break;

//get the things in position (g_xi,g_yi)-----

    case (101):

        lcd_disp(1,"Getto");

        lcd.setCursor(2,2);

        lcd.print("----- ");

        lcd.setCursor(16,2);

        lcd.print("--");

        lcd.setCursor(7,3);

        lcd.print("Busy Geting! ");

        break;

//put the things to position (g_xi,g_yi)-----

    case (202):

        lcd.setCursor(16,1);

        lcd.print(g_xi);//-----
        -----

        lcd.setCursor(18,1);

        lcd.print(g_yi);

        lcd_disp(1,"Putto");
```

```
    lcd.setCursor(2,2);

    lcd.print("W:---");

    lcd.setCursor(4,2);

    lcd.print(g_weightnow);

    lcd.setCursor(16,2);

    lcd.print("--");

    lcd.setCursor(7,3);

    lcd.print("Busy Putting! ");

    break;

//find the new things,and write the weight to (xi,yi)-----

case (300):

    lcd.setCursor(16,1);

    if(g_num1[0]>=0)

    {

        lcd.print(g_num1[0]);

    }

    else

    {

        lcd.print("-");

    }

    lcd.setCursor(18,1);

    if(g_num1[1]>=0)

    {

        lcd.print(g_num1[1]);
```

```
    }

    else

    {

        lcd.print("- ");

    }

    lcd.setCursor(2,2);

    lcd.print("Delete ");

    lcd.setCursor(16,2);

    lcd.print("Go");

    lcd.setCursor(7,3);

    lcd.print("Add  New Thing");

    break;

//the machine is already for man to pick up the things-----

    case (400):

        lcd.setCursor(2,2);

        lcd.print("----- ");

        lcd.setCursor(16,2);

        lcd.print("Ok");

        lcd.setCursor(7,3);

        lcd.print("Pick  Please! ");

        break;

//error mode-----

    case (500):

        lcd.setCursor(7,3);
```

```
        lcd.print("Error No.0!  ");

        break;

    case (503):

        lcd.setCursor(7,3);

        lcd.print("Error No.3!  ");

        break;

    case (504):

        lcd.setCursor(7,3);

        lcd.print("Error No.4!  ");

        break;

//set mode-----

    case (600):

        lcd_disp(1,"SET:  1.clr  2.init");

        lcd.setCursor(2,2);

        lcd.print("BACK  ");

        lcd.setCursor(16,2);

        lcd.print("--");

        lcd.setCursor(7,3);

        lcd.print("Set Mode!  ");

        g_mode+=1;

        break;

    case (700):

        lcd_disp(1,"U-1- -3- -5- -7- -9-");

        lcd_disp(2,"N--- ----");
```



```
        lcd_disp(3,"D-0- -2- -4- -6- -8-");

        g_mode=701;

        break;

    case (701):

        lcd.setCursor(1,2);

        lcd.print(g_setangle[0]);

    lcd.print(" ");

        lcd.setCursor(5,2);

        lcd.print(g_setangle[1]);

    lcd.print(" ");

        lcd.setCursor(9,2);

        lcd.print(g_setangle[2]);

    lcd.print(" ");

        lcd.setCursor(13,2);

        lcd.print(g_setangle[3]);

    lcd.print(" ");

        lcd.setCursor(17,2);

        lcd.print(g_setangle[4]);

    lcd.print(" ");

        break;

    default:

        break;

    }

}
```

```
/*  
  
void G_DCMOV()  
  
{  
  
}  
  
*/  
  
void G_STMOV()//舵机脉冲周期性发送程序,用于维持舵机运转  
  
{  
  
stmovcontrol(g_setangle[0],g_setangle[1],g_setangle[2],g_setangle[3],g_setangle[4]);  
  
}  
  
void atstmov(int angl1,int angl2,int angl3,int angl4,int angl5)//改变各个舵机的  
脉冲长度(angl:改变脉冲长度的值)  
  
{  
  
    g_setangle[0]+=angl1;  
  
    g_setangle[1]+=angl2;  
  
    g_setangle[2]+=angl3;  
  
    g_setangle[3]+=angl4;  
  
    g_setangle[4]+=angl5;  
  
}  
  
void G_PAUSE()//舵机暂停服务程序
```

```
{

    int ipauses[5];

    for(int i=0;i<5;i++)ipauses[i]=pulsesnow[i];

    G_STMOV();

    for(int i=0;i<5;i++)pulsesnow[i]=ipauses[i];

}

//-----

void LCDinit()//LCD 初始化显示程序

{

    lcd_disp(0,"-----NX ROBOT-----");

    lcd_disp(1,"SlectPosition: [-,-]");

    lcd_disp(2,"<<Delete          Go>>");

    lcd_disp(3,"State: Ready!          ");

}

//GLOBAL_MOV_CONTROL-----
-----

void setstmov(int angl1,int angl2,int angl3,int angl4,int angl5)//设置各个舵机脉冲长度

{

    g_setangle[0]=angl1;

    g_setangle[1]=angl2;

    g_setangle[2]=angl3;

    g_setangle[3]=angl4;
```

```
    g_setangle[4]=angl5;  
}
```

void set234mov(int angl3,int angl4,int angl5)//设置 2,3,4 号舵机(最上面的 3 个舵机)的脉冲长度

```
{  
    g_setangle[2]=angl3;  
    g_setangle[3]=angl4;  
    g_setangle[4]=angl5;  
}
```

void atonemov(int _movi,int _angl)//改变单个舵机的脉冲长度(movi:舵机号/angl:改变脉冲长度的值)

```
{  
    g_setangle[_movi]=_angl+g_setangle[_movi];  
}
```

void setonemov(int movi,int angl)//设置单个舵机的脉冲长度(movi:舵机号/angl:脉冲长度)

```
{  
    g_setangle[movi]=angl;  
}
```

void positturn(int xi)//机器人转到横坐标为 xi 的物品的程序

```
{
```

```
switch (xi)

{

case (0):

    setonemov(0,30);

    setonemov(1,18);

    break;

case (1):

    setonemov(0,77);

    setonemov(1,18);

    break;

case (2):

    setonemov(0,77);

    setonemov(1,18);

    break;

case (3):

    setonemov(0,77);

    setonemov(1,18);

    break;

default:

    break;

}

}
```

void positlift(int yi)//机器人拿起纵坐标为 yi 的物品程序

```
{

    if(g_movflag)

    {

        g_movflag=0;

        if(yi_mode<8)//Ylift[yi,yi_mode,0]!=0) || (Ylift[yi,yi_mode,1]!=0) ||
(Ylift[yi,yi_mode,2]!=0)

        {

set234mov(Ylift[yi][yi_mode][0],Ylift[yi][yi_mode][1],Ylift[yi][yi_mode][2]);

            yi_mode+=1;

        }

        else

        {

            yi_mode=100;

        }

    }

}

void positdown(int yi)//机器人放到纵坐标为 yi 的物架程序

{

    if(g_movflag)

    {

        g_movflag=0;

        if(yi_mode<8)//Ylift[yi,yi_mode,0]!=0) || (Ylift[yi,yi_mode,1]!=0) ||
(Ylift[yi,yi_mode,2]!=0)

        {
```

```
set234mov(Ydown[yi][yi_mode][0],Ydown[yi][yi_mode][1],Ydown[yi][yi_mode][2]);
```

```
    yi_mode+=1;
```

```
    }
```

```
else
```

```
{
```

```
    yi_mode=100;
```

```
}
```

```
}
```

```
}
```

```
//STMOV-----  
-----
```

```
void movinit()//舵机位置初始化程序
```

```
{
```

```
    for(int i=2;i<5;i++)
```

```
    {
```

```
        for(int j=0;j<50;j++)
```

```
        {switch(i)
```

```
        {case 4:
```

```
            digitalWrite(servopin[4],HIGH);//将舵机接口电平至高
```

```
            delayMicroseconds(pulsesnow[4]);//延时脉宽值的微秒数
```

```
            digitalWrite(servopin[4],LOW);//将舵机接口电平至低
```

```
        case 3:
```

```
digitalWrite(servopin[3],HIGH);//将舵机接口电平至高  
  
delayMicroseconds(pulsesnow[3]);//延时脉宽值的微秒数  
  
digitalWrite(servopin[3],LOW);//将舵机接口电平至低  
  
case 2:  
  
digitalWrite(servopin[2],HIGH);//将舵机接口电平至高  
  
delayMicroseconds(pulsesnow[2]);//延时脉宽值的微秒数  
  
digitalWrite(servopin[2],LOW);//将舵机接口电平至低  
  
default:  
  
delay(50);  
  
}  
  
}  
  
}  
  
for(int i=0;i<2;i++)  
{  
  
for(int j=0;j<50;j++)  
{  
  
switch(i)  
{case 1:  
  
digitalWrite(servopin[1],HIGH);//将舵机接口电平至高  
  
delayMicroseconds(pulsesnow[1]);//延时脉宽值的微秒数  
  
digitalWrite(servopin[1],LOW);//将舵机接口电平至低  
  
case 0:  
  
digitalWrite(servopin[0],HIGH);//将舵机接口电平至高  
  
delayMicroseconds(pulsesnow[0]);//延时脉宽值的微秒数
```



```
digitalWrite(servopin[0],LOW);//将舵机接口电平至低

case 4:

digitalWrite(servopin[2],HIGH);//将舵机接口电平至高

delayMicroseconds(pulsesnow[2]);//延时脉宽值的微秒数

digitalWrite(servopin[2],LOW);//将舵机接口电平至低

case 3:

digitalWrite(servopin[3],HIGH);//将舵机接口电平至高

delayMicroseconds(pulsesnow[3]);//延时脉宽值的微秒数

digitalWrite(servopin[3],LOW);//将舵机接口电平至低

case 2:

digitalWrite(servopin[4],HIGH);//将舵机接口电平至高

delayMicroseconds(pulsesnow[4]);//延时脉宽值的微秒数

digitalWrite(servopin[4],LOW);//将舵机接口电平至低

default:

delay(50);

}

}

}

}

void servopulse(int i,int ilength)//脉冲-引脚输出子程序

{

digitalWrite(servopin[i],HIGH);//将舵机接口电平至高

delayMicroseconds(ilength);//延时脉宽值的微秒数
```

```
digitalWrite(servopin[i],LOW);//将舵机接口电平至低

}

void stmovcontrol(float angl11,float angl12,float angl13,float angl14,float
angl15)//依次输出各个舵机的脉冲信号长度

{

    float ipulse[5],iangle[]={angl11,angl12,angl13,angl14,angl15};

    g_movflag=1;

    for(int i=0;i<5;i++)

    {

        ipulse[i]=iangle[i]*10+500;

        if(ipulse[i]<=pulsesmin[i]){ipulse[i]=pulsesmin[i];}

        if(ipulse[i]>=pulsesmax[i]){ipulse[i]=pulsesmax[i];}

        if(ipulse[i]>(pulsesnow[i]+movspeed)){pulsesnow[i]+=movspeed;g_movflag=0;}

        else
        if(ipulse[i]<(pulsesnow[i]-movspeed)){pulsesnow[i]-=movspeed;g_movflag=0;}

        else{pulsesnow[i]=ipulse[i];}

        servopulse(i,pulsesnow[i]);

    }

}

int pulsetoangle(int pulse_temp,int ik)//实际脉冲信号长度和程序中转化的脉
冲长度的转换程序

{
```

```
    return (pulse_temp-500)/10;

}

//CSB-----
-----

float measure()//超声波测距程序
{
    float distance;

    digitalWrite(TrigPin, LOW);

    delayMicroseconds(2);

    digitalWrite(TrigPin, HIGH);

    delayMicroseconds(10);

    digitalWrite(TrigPin, LOW);

    // 检测脉冲宽度，并计算出距离

    distance = pulseIn(EchoPin, HIGH,50000) / 58.00;

    return distance;
}

//EEPROM-----
-----

void writeweight(byte xi ,byte yi,int iweight16)//记录 xi,yi 位置物品的重量
{
    if(xi<10&&yi<10)
    {
```

```
EEPROM.write(xi*20+2*yi, iweight16>>8);

EEPROM.write(xi*20+2*yi+1, iweight16);

}

}

int getweight(byte xi ,byte yi)//获取 xi,yi 位置物品的重量

{

    int iread;

    if(xi<10&&yi<10)

    {

        iread=(EEPROM.read(xi*20+2*yi)<<8)+EEPROM.read(xi*20+2*yi+1);

        return iread;

    }

}

void weightdataclr()//清空存储器

{

    for (int iclr = 0; iclr < 300; iclr++)EEPROM.write(iclr, 1);

}

byte getposition(int iweight16)//获取当前重量的物品的位置 xi,yi

{

    int xi,yi,itemp=3000,difmin=3000;

    for(int i=0;i<10;i++)
```

```
{  
    for(int j=0;j<10;j++)  
    {  
        itemp=abs(iiweight16-getweight(i,j));  
        if(itemp<difmin){difmin=itemp;xi=i;yi=j;}  
    }  
}  
if(difmin<240){return xi*10+yi;}  
else{return 0xFF;}  
}  
  
byte ifbethere(byte xi,byte yi)//获取当前位置 xi,yi 是否已经有物品存在  
{  
    return EEPROM.read(xi*10+yi+200);  
}  
  
void puttothere(byte xi,byte yi)//将 ifbewhere(xi,yi)置 1  
{  
    EEPROM.write(xi*10+yi+200,1);  
}  
  
void getfromthere(byte xi,byte yi)//将 ifbewhere(xi,yi)置 0  
{  
    EEPROM.write(xi*10+yi+200,0);  
}
```

```
}
```

```
//HWX-----  
-----
```

```
int HWX(unsigned long int k)//红外信号转换程序
```

```
{
```

```
    switch (k) {
```

```
        case (16738455):                //0
```

```
        { return 0;
```

```
          break;
```

```
        }
```

```
        case (16724175):                //1
```

```
        { return 1;
```

```
          break;
```

```
        }
```

```
        case (16718055):                //2
```

```
        { return 2;
```

```
          break;
```

```
        }
```

```
        case (16743045):                //3
```

```
        { return 3;
```

```
          break;
```

```
        }
```

```
        case (16716015):                //4
```

```
    {   return 4;

        break;

    }

case (16726215):           //5

{   return 5;

    break;

}

case (16734885):           //6

{   return 6;

    break;

}

case (16728765):           //7

{   return 7;

    break;

}

case (16730805):           //8

{   return 8;

    break;

}

case (16732845):           //9

{   return 9;

    break;

}

case (16720605):           //left
```

```
        {   return 100;

            break;

        }

        case (16712445):           //right

        {   return 200;

            break;

        }

        default: return 320;

    }

    //irrecv.resume(); // 接收下一个值

}

//LCD-----
-----

void lcd_clrdisp(int line,char *p)//第 line 行先清除后显示数据

{

    lcd.setCursor(0,line);

    lcd.print("                ");

    lcd.setCursor(0,line);

    lcd.print(p);

}

void lcd_disp(int line,char *p)//在第 line 行显示数据

{

    lcd.setCursor(0,line);
```



```
    lcd.print(p);

}

void lcd_clr(int line)//清除第 line 行的显示数据

{

    lcd.setCursor(0,line);

    lcd.print("                ");

}


void lcd_disprint(int line,unsigned long int int16)//在显示屏上显示整型 int 数据

{

    lcd.setCursor(0,line);

    unsigned long int ienable,i;

    unsigned long int iint;

    ienable=0;

    iint=int16;

    for(i=0;i<16;i++)

    {

        itemp[i]=0;

        itemp[i]=iint%10;

        iint=iint/10;

        itemp[i]+=48;

    }

    for(i=16;i>0;i--)

    {
```

```
if(ite mp[i-1]!=48){ ienable=1;}

if(ienable==1){ lcd.write(ite mp[i-1]);}

    }

}

//HX711-----
-----

int hx711_weighting()//获取当前物品重量

{

    double sum = 0;

    int iwe;

    for (int i = 0; i < 3;i++)

    {

        sum += hx.read();

    }

    if(sum<10000000&&sum>-10000000)

    {

        iwe=sum/1000;

    }

    else

    {

        iwe=-10000;

    }

    iwe=abs(iwe);
```

```
    return iwe;

}
```

```
//RTGY-----
-----
```

```
byte RTGY()//获取当前是否有人存在

{

return digitalRead(RTGY_PIN);

}
```

2. 红外信号接收单片机程序:

```
#include <IRremote.h>

int RECV_PIN = 9;//设置红外接收引脚

unsigned long int rec;

IRrecv irrecv(RECV_PIN);

decode_results results;

char HWX(unsigned long int k);

void setup()

{

    Serial.begin(9600);
```

```
    irrecv.enableIRIn(); // 初始化红外接收  
  
}
```

```
void loop()  
{  
  
    //delay(50);  
  
    if (irrecv.decode(&results))  
    {  
  
        rec=results.value;  
  
        irrecv.resume(); // 接收下一组数据  
  
        Serial.write(HWX(rec));  
  
        //Serial.print(rec);  
  
    }  
  
}
```

```
char HWX(unsigned long int k)  
{  
  
    switch (k) {  
  
        case (16738455):           //0  
  
            { return '0';  
  
              break;  
  
            }  
  
        case (16724175):           //1  
  
            { return '1';
```

```
        break;

    }

    case (16718055):                //2

    {   return '2';

        break;

    }

    case (16743045):                //3

    {   return '3';

        break;

    }

    case (16716015):                //4

    {   return '4';

        break;

    }

    case (16726215):                //5

    {   return '5';

        break;

    }

    case (16734885):                //6

    {   return '6';

        break;

    }

    case (16728765):                //7

    {   return '7';
```

```
        break;

    }

    case (16730805):                //8

    {   return '8';

        break;

    }

    case (16732845):                //9

    {   return '9';

        break;

    }

    case (16720605):                //left

    {   return 'B';

        break;

    }

    case (16712445):                //right

    {   return 'G';

        break;

    }

    case (16761405):

    {

        return 'P';

        break;

    }

    case (4294967295):
```

```
    {  
  
        delay(100);  
  
        return 'E';  
  
        break;  
  
    }  
  
    default:  
  
        break;  
  
    }  
  
}
```