宏晶杯
单片机应用大赛
设
计
报
告

学　　校：　山东大学

坠物检测及自动接收装置

目录

摘要

本系统是基于 IAP15W4K58S4 单片机系统的设计。论文介绍了基本要求及总体方案设计，硬件设计，软件设计以及系统的调试与分析。本系统实现了检测坠物，通过接收装置实现接收坠物，并在接收后成功收起。坠落高度、坠物质量和坠物数量，通过 lcd1602 显示屏显示和语音模块播报。

关键词：IAP15W4K58S4 单片机，坠物检测和接收

ABSTRACT

This system is based on the design of IAP15W4K58S4 single chip microcomputer system.

This paper introduces the basic requirements and overall scheme design, hardware design, software design and system debugging and analysis.

This system realizes the detection of falling objects, receiving falling objects through the receiving device, and after receiving the object, the object is put away successfully.

The height, mass and quantity of falling objects are displayed on LCD1602 display screen and broadcast by voice module.

KEY WORDS: IAP15W4K58S4 single chip microcomputer, falling object detection and reception.

第一章 基本要求及分析

1.1 基本要求

（1） 基本功能

1）坠物直径 3 厘米左右，从非固定高度做自由落体运动，可用橡皮做测试；

2）检测装置与接收装置之间的高度作为一个评价标准，报告和答辩中应说明；

3）接收装置正常为收起状态，检测到坠物之后方能动作，需要检测是否能够接收成功，成功之后必须收起。

（2） 拓展部分

1） 可连续检测接收，并计算坠物个数；

2） 能显示初始高度；

3） 语音播报；

4） 其它。

1.2 总体方案设计

本系统通过两个红外线传感器检测坠物，可以得到坠物通过两传感器的时间差，通过数学计算得到坠物的起始高度。并且当坠物通过第一个红外传感器时，传感器发送一个信号使得舵机转动 45 度，带动接收装置接收到坠物。第二个红外传感器在纸杯上方且紧靠纸杯，所以坠物通过第二个红外传感器后发送一个信号代表接收成功，超出一定时间之后接收失败，接收成功后，短暂延时之后，舵机转回，带动接收装置收起。通过称重模块测出坠物的质量。各数据可以通过 lcd1602 显示屏显示，同时通过语音模块进行播报。通过软件设计，可以实现连续检测和接收。

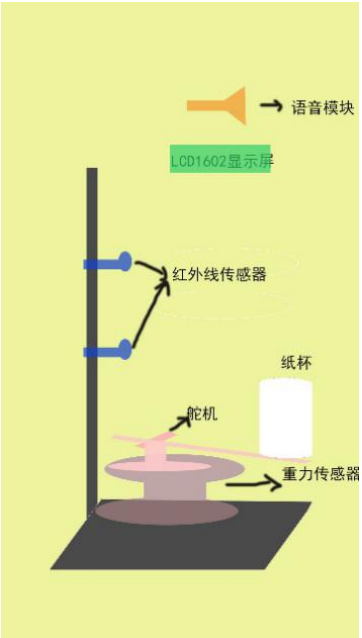

图 1-1 系统效果图

图 1-2 系统实物图

第二章 硬件设计

2.1 硬件电路主要器件

（1）电池管理模块：

提供 5V 电压的移动电源。

（2）控制模块：

IAP15W4K58S4 单片机最小系统。

（3）称重模块：

由电阻应变式压力传感器和 HX711AD 模块组成。

（4）接收装置模块

接收装置由 E18-D80NK 红外避障传感器，MG90D 数字舵机和纸杯组成。

（5）显示模块

LCD1602 液晶显示屏。

（6）语音模块

选用 JQ8900-16P 语音模块。

2.2 控制模块

本系统由大赛组委会提供的 IAP15W4K58S4 芯片，IAP15W4K58S4 系列单片机是 STC 生产的单时钟/机器周期（1T）的单片机，是高速/高可靠/低功耗/超强抗干扰的新一代 8051 单片机，采用 STC 第八代加密技术，无法解密，指令代码完全兼容传统 8051，但速度上快 8-12 倍。内部集成高精度 R/C 时钟（±0.3%），±1%温漂（-40℃~+85℃），常温下温漂±0.6%（-20℃~+65℃），ISP 编程时 5Mhz~35Mhz 宽范围可设置，可彻底省掉外部昂贵的晶振和外部复位电路（内部可集成高可靠复位电路，ISP 编程时 8 级复位门槛电压可选）。3 路 CCP/PWM/PCA，8 路高速 10 位 A/D 转换（30 万次/秒），内置 2K 字节大容量 SRAM，2 组超高速异步串行通信端口（UART1/UART2，可在 5 组管脚之间进行切换，分时复用可做 5 组串口使用），1 组高速同步串行通信端口 SPI，针对多串行口通信/电机控制/强干扰场合。

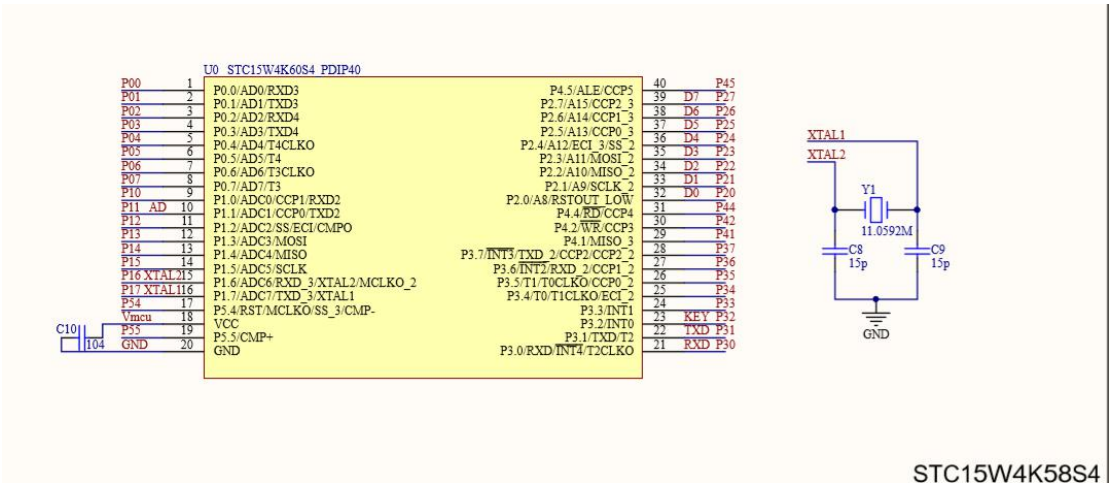

图 2-1 STC15W4K58S4 电路图

2.3 称重模块

由电阻应变式压力传感器和 HX711AD 模块组成。

HX711AD 模块采用了海芯科技集成电路专利技术，是一款专为高精度电子秤而设计的 24 位 A/D 转换器芯片。与同类型其它芯片相比，该芯片集 成了包括稳压电源、片内时钟振荡器等其它同类型芯片所需要的外围电路，具有集成度高、 响应速度快、抗干扰性强等优点。降低了电子秤的整机成本，提高了整机的性能和

可靠性。

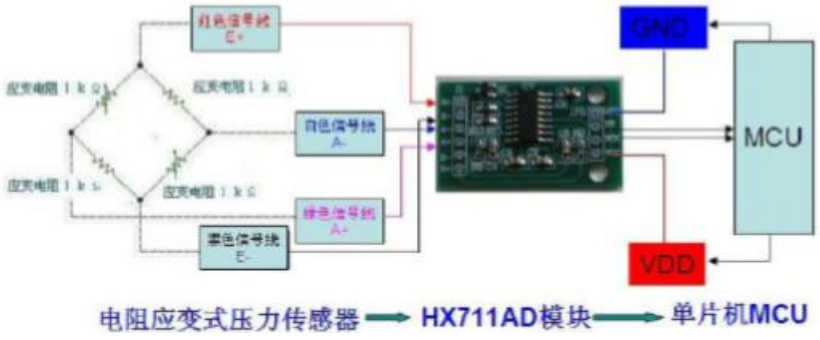

图 2-2 称重模块连接框图

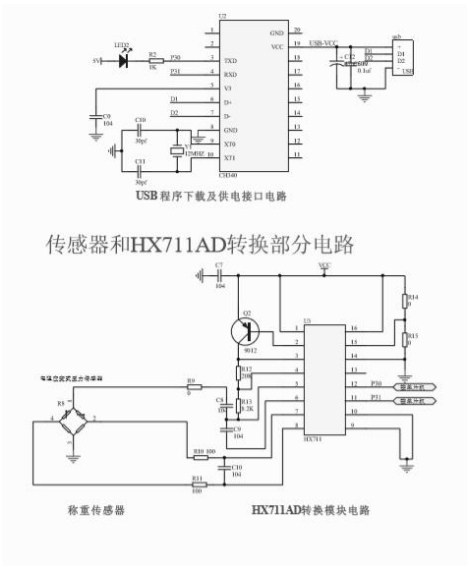

图 2-3 称重传感器和 HX711AD 部分转换电路

技术参数表：

量程(kg)		1, 2 5	
综合误差(%F.S)	0.05	额定输出温度飘移(%F.S/10℃)	≤0.15
灵敏度(mv/v)	1.0±0.1	零点输出(mV/V)	±0.1
非线性(%F.S)	0.05	输入电阻(Ω)	1000±50
重复性(%F.S)	0.05	输出电阻(Ω)	1000±50
滞后(%F.S)	0.05	绝缘电阻(MΩ)	≥2000(100VDC)
蠕变(%F.S/3min)	0.05	推荐激励电压(V)	3～12
零点漂移(%F.S/1min)	0.05	工作温度范围 (℃)	-10～+50
零点温度漂移(%F.S/10℃)	0.2	过载能力(%F.S)	150

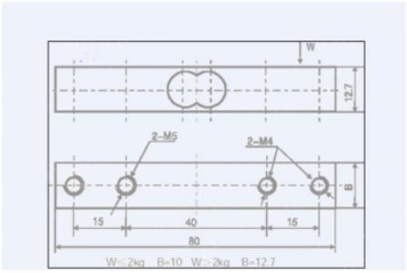

图 2-4 技术参考表

2.4 接收装置模块

接收装置由 E18-D80NK 红外避障传感器，MG90D 数字舵机和纸杯组成。坠物通过两个红外传感器发送信号时间差 t，经数学和物理公式计算得到坠物初始高度 H。图中，h1 为坠落物到第一个红外线传感器的距离，h2 为两个红外线传感器的距离，h3 为为近地传感器到参考地面的距离。

计算过程如下：

$$v_2^2 - v_1^2 = 2gh_2$$

$$v_2 = v_1 + gt$$

$$(v_1 + gt)^2 - v_1^2 = 2gh^2;$$

$$v_1 = \frac{2gh_2 - g^2t^2}{2gt};$$

$$h_1 = \frac{4h_2^2 - 4h_2gt^2 + g^2t^4}{8gt^2}$$

$$\mathrm{H} = \frac{4h_2^2 - 4h_2\,gt^2 + g^2t^4}{8gt^2} + h_2 + h_3;$$

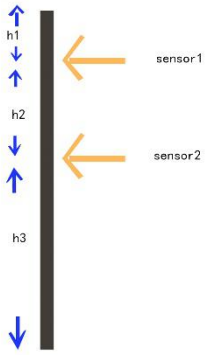

图 2-5 红外传感器位置图

当 sensor1 传感器检测到坠物时，舵机带动纸杯转动 90° 准备接收坠物，然后经 150ms，若在此时间内 sensor2 检测到坠物代表接收成功，反之接收失败。

E18-D80NK 红外避障传感器是一种集发射与接收于一体的光电传感器，发射光经过调制后发 出，接收头对反射光进行解调输出。有效的避免了可见光的干扰。透镜的使用，也使 得这款传感器最远可以检测 80 厘米距离的问题，检测障碍物的距离可以根据要求通过尾部的电位器旋钮进行调节。优势：该传感器具有探测距离远、受可见光干扰小、价格便宜、易于装配、使用方便等特点，可以广泛应用于机器人避障、流水线计件等众多场合。

图 2-6 E18-D80NK 红外避障传感器实物图

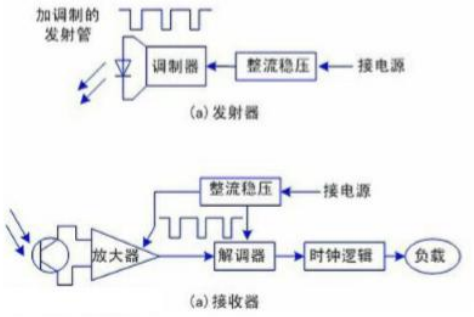

图 2-7 E18-D80NK 红外避障传感器内部原理图

MG90D 数字舵机其组成部分主要有齿轮组、电机、电位器、电机控制板、壳体这几大部分。优势是：1.因为微处理器的关系，数字舵机可以在将动力脉冲发送到舵机马达之前，对输入的信号根据设定的参数进行处理。这意味着动力脉冲的宽度，就是说激励马达的动力，可以根据微处理器的程序运算而调整，以适应不同的功能要求，并优化舵机的性能。2.数字舵机以高得多的频率向马达发送动

力脉冲。3.价格便宜，金属齿轮，耐用度好，相比于 MG90s，升级版，扭力大。

图 2-8 MG90D 数字舵机实物图

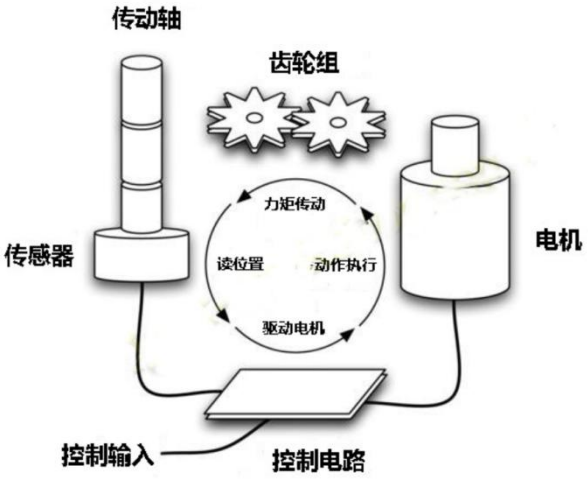

图 2-9 MG90D 数字舵机内部组成图

2.5 显示模块

LCD1602 液晶显示屏是一种工业字符型液晶，能够同时显示 16x02 即 32 个字符。LCD1602 液晶显示的原理是利用液晶的物理特性，通过电压对其显示区域进行控制，即可以显示出图形。LCD1602 液晶显示模块可以只用 D4-D7 作为四位数据分两次传送。这样的话，可以节省 MCU 的 I/O 口资源。lcd1602 可以显示 2 行 16 个字符，有 8 为数据总线 D0-D7，和 RS、R/W、EN 三个控制端口，工作电压为 5V，并且带有字符对比度调节和背光。LCD1602 液晶显示模块可以和单片机直接接口，电路连接简单。

优点：

1、是字符型液晶，显示字母和数字比较方便

2、控制简单

3、成本较低

缺点：

1、显示的字体有大小限制

2、不能显示图形等等

3、它不能显示曲线

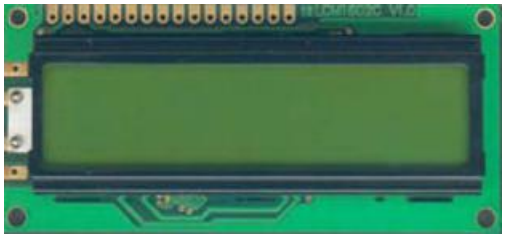

图 2-10 LCD1602 液晶显示屏实物图

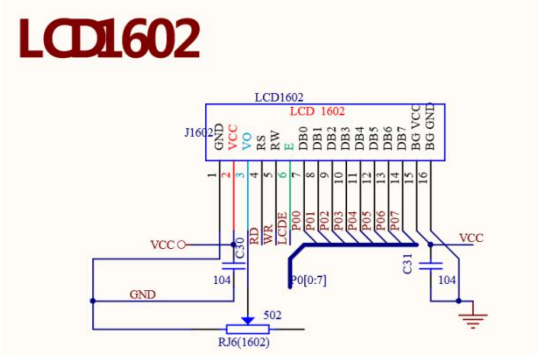

图 2-11 LCD1602 液晶显示屏电路图

2.6 语音模块

选用 JQ8900-16P 语音模块。JQ8900-16P 选用的是 SOC 方案，集成了一个 16 位的 MCU，以及一个专门针对音频解码的 ADSP，采用硬解码的方式，更加保证了系统的稳定性和音质。小巧尺寸更加满足嵌入其它产品的需求。SPI-flash 更换语音内容，此芯片最大的优势在于能够灵活的更换 SPI-flash 内的语音内容，省去了传统语音芯片需要安装上位机更换语音的麻烦，SPI FLASH 直接模拟成 U 盘，跟拷贝 U 盘一样，非常方便。使得产品研发和生产变得便捷简单。一线串口控制模式、RX232 串口控制可选，为研发提供更多的选择性。

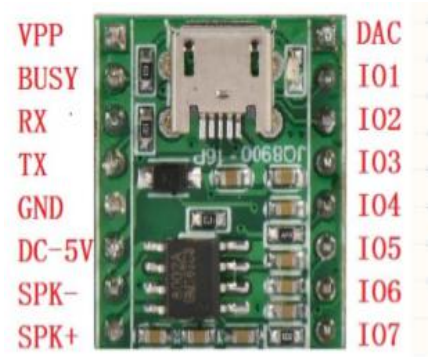

图 2-12 JQ8900-16P 语音模块实物图

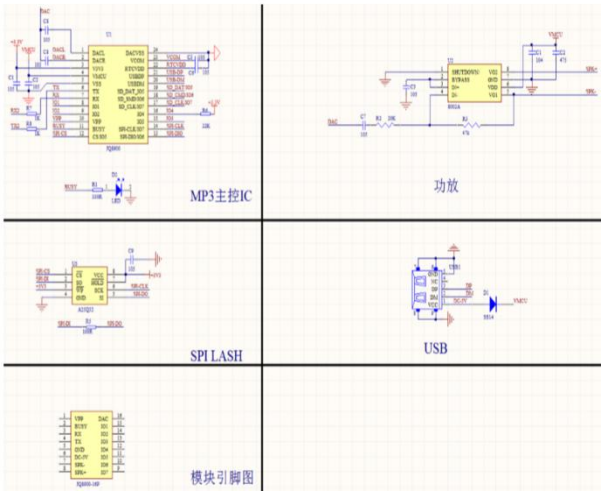

图 2-13 JQ8900-16P 语音模块电路图

第三章 软件算法设计

3.1 软件开发环境：

Keil Software 公司：Keil uVision4 集成开发环境（C 语言）

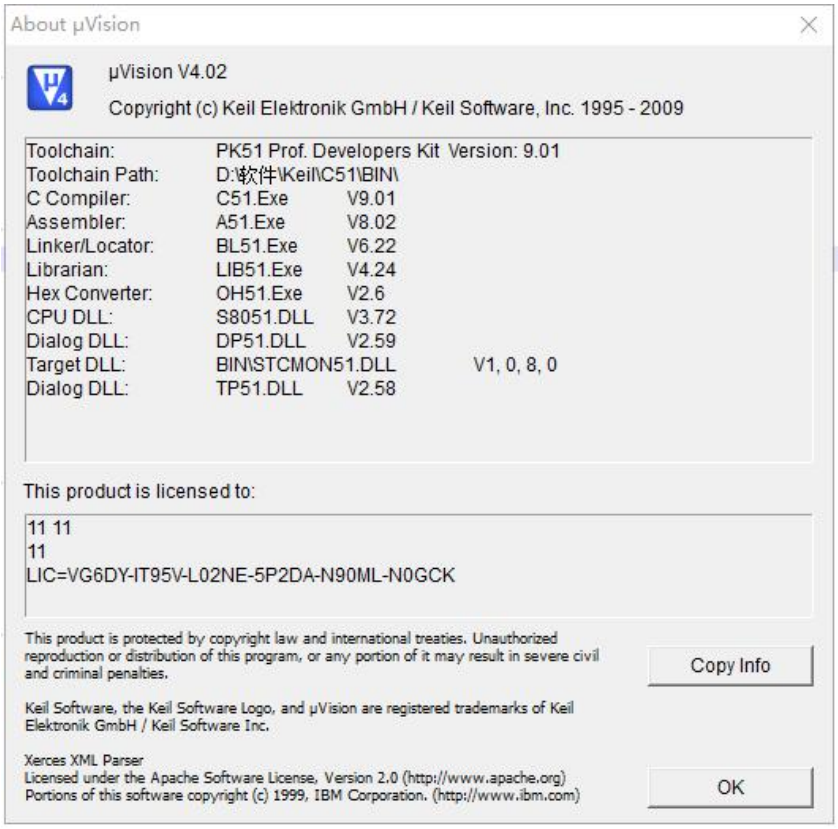

图 3-1 keil 开发环境图

3.2 设计流程图：

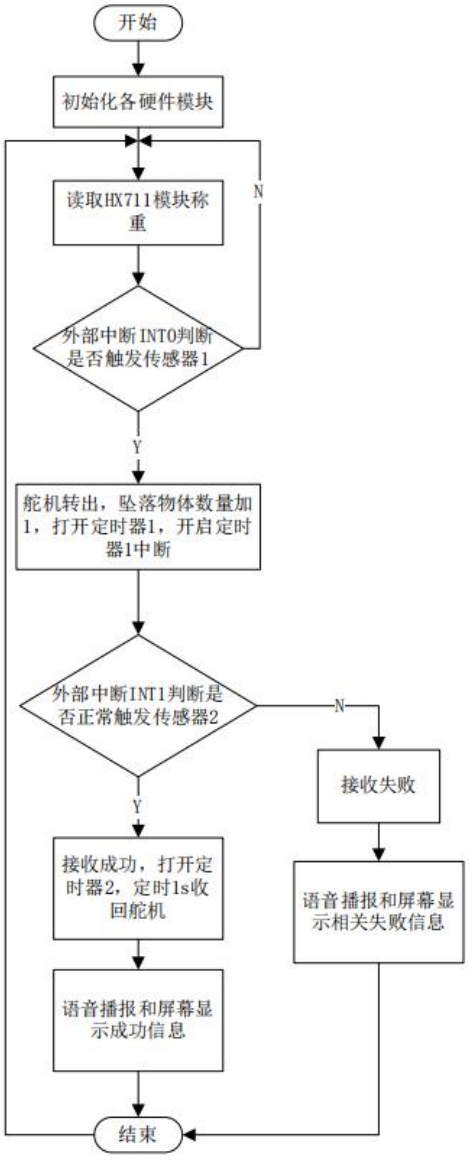

图 3-2 系统流程图

3.3 程序模块：

3.3.1 各 C 文件模块说明：（具体程序详见附录）

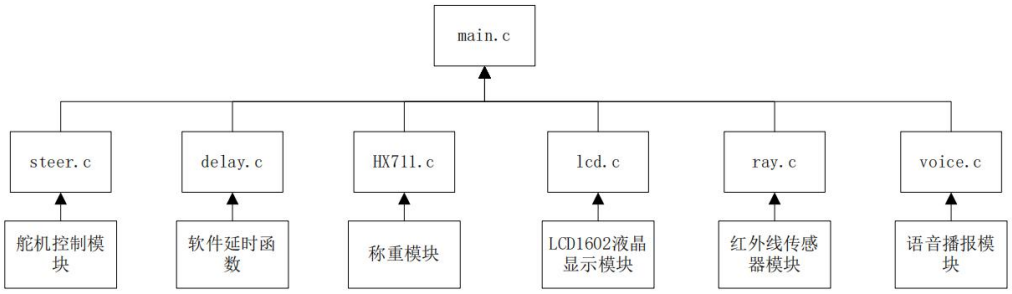

图 3-3 .c 程序文件图

3.3.2 相关头文件：

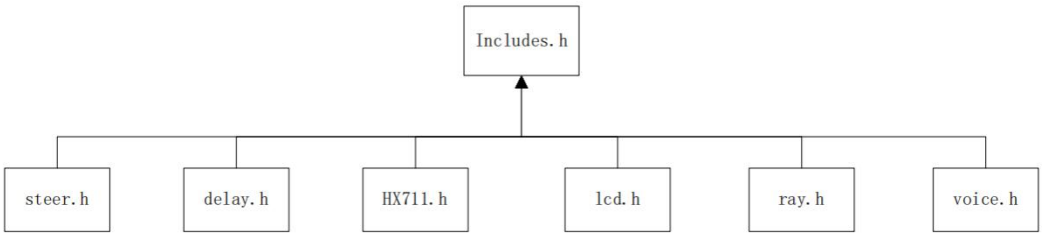

图 3-4 .h 头文件程序图

3.4 下载程序：

使用了 STC 公司提供的 STC-ISP 在线系统编程软件烧写程序，并在软件中设置 IAP15W4K58S4 的内部时钟为 24Mhz，保证单片机在较为稳定的工作频率范围提供较快的运行速度。

图 3-5 STC-ISP 软件

第四章 测试及调节过程

在系统调试过程中，主要采取了分析计算与实验相结合的方法，改进系统准确性和稳定性以适应可接收到更高的物体坠落。

整个系统的调试过程中遇到的问题和改进方法如下：

（1）装置组装过程中，每个模块之间的拼接不牢固。因此我们除必要连接点用螺丝紧固外，令用热熔胶，502 速干胶固定，在红外传感器与铁架台的铁杆之间除用夹子固定，令在其之间用卫生纸以增加摩擦力，并在夹子和铁杆之间用胶带多次缠绕。在最小系统和传感器之间连接线处用胶带固定，防止可能因为搬运过程中造成导线脱落等问题。

（2）接收装置在实验中，发现随着坠物高度当达到 1.5 米附近时，坠物因为冲力太大会将在舵机之上的接收杯子砸落，因此我们做出改进，在舵机和杯子之间用直尺相连，在直尺的另一端家重物（两个金属锁头），并在杯子内放置海绵，并将杯子口粘上塑料减震气泡，利用类似篮球框篮网的减速原理缓冲，通过这两个措施来解决坠物因太高，冲力太大，造成接收装置被砸坏的现象。

（3）舵机反应速度太慢导致不能接收坠物的问题。改进可以把第一个红外传感器相应升高来给舵机更多的反应时间，但是这样会造成可测坠物高度范围减小，所以此方法不可行。第二种方法，把舵机的旋转角度有 90° 改为现在的 45°，可以节省一半的反应时间，经试验此方法可行。

（4）误差问题。经实验分析误差原因有以下几点：

①两个红外传感器之间的距离，按照程序设定为 15cm，但是因为工具精

度原因会造成不可避免的误差，因此得到的最终的高度会有一定的误差，并且由于重力加速度的原因，速度会逐渐增大，误差会随高度增大而增大。

②计算坠物经过两红外传感的时间，为提高反应速度和精度，我们用中断来开启和关闭定时器，但坠物通过两红外传感的时间随高度加速缩短，定时器和单片机指令处理相互配合，可靠精度只能达到 1ms 左右，因此导致的不可避免误差。

③当地的重力加速度会造成误差，为尽可能减少误差我们选用了青岛当地的重力加速度 $g=9.7985m/s^2$。

④坠物坠落时，人的手可能会不自觉的加入一个向下的初速度。

经过改良之后，经过经验测得误差在一米之内误差小于 5cm，在两米之内误差小于 30cm。

（5）：原本想使用称重模块的称重功能进行坠物数量识别，后来发现差分算法出现了问题，无法实现预设功能，于是改成了使用红外传感器进行感知。

可以改进的地方：

① ：在舵机的选择上，我们可以选用速度更加快的气动开关和电磁阀来解决舵机转速过慢导致的接收问题

② ：除了实现这些基本功能之外，还可以连接 Bluetooth 或 Zigbee 模块来和计算机或手机进行串口通信

③ ：语音播报时不够自然流畅

创新点：

① ：采用了充电宝对系统进行供电，相比于使用计算机直接供电，充电宝更加便携，可移动性更好

② ：对杯子进行了特殊的处理，使得其可以承受更大的冲力，接收到高度更高的物体，系统更加可靠

③ ：采用了称重模块，可以对接收到的物体进行称重并显示

④ ：对整个系统进行了封装，使其更加美观，易于移动。

⑤ ：利用有限的单扬声器产生了意料之外的立体声效果

第五章 总结

系统最终实现了：

（1）检测坠物从非固定高度落下，已测试检测高度最高为 2.1m；

（2）第一个检测装置与接收装置间的距离为 18cm，第二个检测装置与接收装置间的距离为 3cm；

（3）接收装置正常为收起状态，检测到坠物后转出，接收成功后收回，舵机反应时间经粗略测量为 0.3s。

（4）可连续检测接收，并计算坠物个数；

（5）能显示初始高度、坠物个数和坠物重量；

（6）语音播报；

（7）称量接收坠物的重量。

经过两周多的时间的程序编写和实物系统调试，我们遇到了各种各样的问题，经过不断的调试，不畏艰难，精诚合作。完成了坠物检测与接收系统，并可以实现相应的功能。纵使有所不足，但是总体还算是差强人意。

通过这次比赛，锻炼了我们三人的合作能力和动手能力。并且对以前的单片机和 c 语言的知识进行了复习。为了这次比赛，付出了很多，经过一次又一次的讨论、计算、测试、安装。花费了很多时间。在整个过程中，我们遇到了各种各样的问题，这些在书本上无法找到答案就需要我们自己一个一个摸索，不论是软件还是硬件上的问题。但同样也收获了很多，不论这次比赛结果如何，我们尽力了，从中学到了很多知识并且得到了许多经验。这是最重要的。

第六章 参考文献

[1] 律者无疆.pwm 波控制舵机原理[EB/OL].

https://blog.csdn.net/weixin_38907560/article/details/81546006 ,2018-08-09

[2] 山重水复又一村.舵机工作原理及程序[EB/OL].

https://blog.csdn.net/sinat_40603235/article/details/78226505 ,2017-10-13

[3] 郭天祥.51 单片机 c 语言教程[M].北京：电子工业出版社，2018.01

[4]栗华，单片机原理与应用实验教程，济南：山东大学出版社，2015.04

[5]阎石，数字电路技术基础，北京：高等教育出版社，2016.04

[6]孙肖子，模拟电子电路及技术基础，西安：西安电子科技大学出版社，20113，06

[7]严雨，51 单片机 c 语言实战教程，北京：电子工业出版社，2017，04

[8]周旭欣，单片机原理及应用，北京：北京航空航天大学出版社，2016.09

[9]丁志杰，数字电路与系统设计，北京：电子工业出版社，2014.08

[10]JQ8900-16P 语音模块使用说明书 V1.3，深圳佳强电子科技有限公司

[11]光电传感器模块 E18-D80NK 漫反射式红外光电开关 避障传感器模块

程序源代码：

//main.c:

```
#include "Includes.h"

void main()
{
    GPIOInit();
    LcdInit();
    INT0Init();
    INT1Init();
    GetGross();

    while(1)
    {
        GetWeight();                    //称重

        LcdWriteCom(0x80);                                  //LCD1602 第一行起始地址
        LcdWriteData(NetWeight/1000 + 0X30);        //0X30 为 ASCII 码表中'0'的值
      LcdWriteData(NetWeight%1000/100 + 0X30);
      LcdWriteData(NetWeight%100/10 + 0X30);
      LcdWriteData(NetWeight%10 + 0X30);
        LcdWriteWord(" g");

        LcdWriteCom(0x80+0x0a);
        LcdWriteWord("num:");
        LcdWriteData(Cnt/10 + 0X30);
        LcdWriteData(Cnt%10 + 0X30);

        SteerBack();            //判断是否收回舵机

        GetHeight(); //计算坠物初始高度

        LcdWriteCom(0x80+0x40);                             //LCD1602 第二行起始地址
        LcdWriteData((long)Height/100000 + 0X30);
        LcdWriteData((long)Height%100000/10000 + 0X30);
        LcdWriteData((long)Height%10000/1000 + 0X30);       //此处暂时为两传感器间距
      LcdWriteData((long)Height%1000/100 + 0X30);
        LcdWriteData('.');
      LcdWriteData((long)Height%100/10 + 0X30);
      LcdWriteData((long)Height%10 + 0X30);
        LcdWriteWord(" cm");
```

```
            if (ReceiveScs == 1)
            {
                VoiceScs();         //接收成功语音播报
                ReceiveScs = 0;     //重置接收成功标志位
            }
            if (ReceiveFal == 1)
            {
                VoiceFal();         //接收失败语音播报
                ReceiveFal = 0;     //重置接收失败标志位
            }
        }
}

/*******************************************************************************
* 函 数 名         : GPIOInit
* 函数功能         : 初始化通用 I/O 口
* 输    入         : 无
* 输    出         : 无
* 说    明         : IO 口设置为准双向口
*******************************************************************************/
void GPIOInit (void)
{
    P0M0 = 0x00;
    P0M1 = 0x00;
    P1M0 = 0x00;
    P1M1 = 0x00;
    P2M0 = 0x00;
    P2M1 = 0x00;
    P3M0 = 0x00;
    P3M1 = 0x00;
    P4M0 = 0x00;
    P4M1 = 0x00;
    P5M0 = 0x00;
    P5M1 = 0x00;
    P6M0 = 0x00;
    P6M1 = 0x00;
    P7M0 = 0x00;
    P7M1 = 0x00;
}

//Includes.h:
#ifndef __INCLUDES_H_
#define __INCLUDES_H_
```

```
/*****包含头文件*****/
#include <stc15.h>
#include <intrins.h>
#include "steer.h"
#include "delay.h"
#include "HX711.h"
#include "lcd.h"
#include "ray.h"
#include "voice.h"

/*****重定义关键词*****/
#ifndef uchar
#define uchar unsigned char
#endif

#ifndef uint
#define uint unsigned int
#endif

#ifndef ulong
#define ulong unsigned long
#endif

/**********************************
外部变量声明
**********************************/
extern double Height;          //初始高度
extern ulong GrossWeight;      //毛重
extern long NetWeight;         //净重
extern bit ReceiveScs;         //接受成功标志位
extern bit ReceiveFal;         //接收失败标志位
extern uchar Cnt;              //坠物计数位
extern ulong TimeBuf;          //通过两传感器的间隔时间

/**********************************
函数声明
**********************************/
/*初始化 GPIO 口*/
void GPIOInit (void);

#endif

//steer.c:
```

```
#include "steer.h"

/*  舵机高电平对应角度：
      0.5ms----------0 度；

    1.0ms----------45 度；

    1.5ms----------90 度；

    2.0ms----------135 度；

    2.5ms----------180 度；*/

#define CYCLE    0x1D4C0L      //定义 PWM 周期为 20ms（6Mhz 时钟频率下）(最大值为 32767)

/*******************************************************************************
* 函 数 名         : SteerInit
* 函数功能         ：初始化驱动舵机
* 输    入         : angle（转动角度：0|45|90|135|180）
* 输    出         ：无
* 说    明         ：针对 24Mhz 主时钟下进行 4 分频输出时钟
*******************************************************************************/
void SteerInit (uchar angle)
{
    float duty;          //占空比

    switch (angle)      //选择不同角度修改占空比
    {
        case (0):    duty = 2.5;
            break;
        case (45):   duty = 5;
            break;
        case (90):   duty = 7.5;
            break;
        case (135):  duty = 10;
            break;
        case (180):  duty = 12.5;
            break;
        default:     break;
    }

    P_SW2 |= 0x80;                  //使能访问 XSFR
    PWMCFG = 0x00;                  //配置 PWM 的输出初始电平为低电平
    PWMCKS = 0x03;                  //选择 PWM 的时钟为 Fosc/(3+1)
```

```
    PWMC = CYCLE;                       //设置 PWM 周期
    PWM2T1 = 0x0000;                    //设置 PWM2 第 1 次反转的 PWM 计数
    PWM2T2 = CYCLE * duty / 100;        //设置 PWM2 第 2 次反转的 PWM 计数
                                        //占空比为(PWM2T2-PWM2T1)/PWMC
    PWM2CR = 0x00;                      //选择 PWM2 输出到 P3.7,不使能 PWM2 中断
    PWMCR = 0x01;                       //使能 PWM 信号输出
    PWMCR |= 0x80;                      //使能 PWM 模块
    P_SW2 &= ~0x80;
}

//steer.h:
#ifndef __STEER_H
#define __STEER_H

#include <stc15.h>

/*****重定义关键词*****/
#ifndef uchar
#define uchar unsigned char
#endif

#ifndef uint
#define uint unsigned int
#endif

/**********************************
函数声明
**********************************/
/*舵机初始化驱动程序*/
void SteerInit (uchar angle);

#endif

//delay.c:
#include "delay.h"

/*****************************************************************************
* 函 数 名         : Delay_us
* 函数功能         : 延时函数，延时 n*1us
* 输    入         : n
* 输    出         : 无
* 说    明         : 该函数是在 24MHZ 晶振下的延时。
*****************************************************************************/
void Delay_us (uint n)          //@24.000MHz
```

```
{
    uint k, a, b;
    for (k=0; k<n; k++)
    {
    for(b=1; b>0; b--)
        {
        for(a=3;a>0;a--);
        }
    }
}

/*****************************************************************************
* 函 数 名          : Delay_ms
* 函数功能          ： 延时函数，延时 n*1ms
* 输    入          : n
* 输    出          ： 无
* 说    明          ： 该函数是在 24MHZ 晶振下的延时。
*****************************************************************************/
void Delay_ms (uint n)          //@24.000MHz
{
    uint a, b, k;
    for (k=0; k<n; k++)
    {
        for(b=129;b>0;b--)
        {
        for(a=45;a>0;a--);
        }
    }
}

/*****************************************************************************
* 函 数 名          : Delay_s
* 函数功能          ： 延时函数，延时 n*1s
* 输    入          : n
* 输    出          ： 无
* 说    明          ： 该函数是在 24MHZ 晶振下的延时。
*****************************************************************************/
 void Delay_s(uint n)           //@24.000MHz
{
    uint a, b, c, m, i;
      for(i=0; i<n; i++)
      for(c=142;c>0;c--)
            for(b=168;b>0;b--)
                  for(a=250;a>0;a--);
```

```
    for(m=2;m>0;m--);
    _nop_();
}

//delay.h:
#ifndef __DELAY_H_
#define __DELAY_H_

#include <stc15.h>
#include <intrins.h>

/*****重定义关键词*****/
#ifndef uchar
#define uchar unsigned char
#endif

#ifndef uint
#define uint unsigned int
#endif

/***********************************
函数声明
***********************************/
/*软件延时 n*1 微秒*/
void Delay_us (uint n);
/*软件延时 n*1 毫秒*/
void Delay_ms (uint n);
/*软件延时 n*1 秒*/
void Delay_s(uint n);

#endif

//HX711.c:
#include "HX711.h"

ulong GrossWeight = 0;  //毛重
long NetWeight = 0;                //净重

//校准参数
//因为不同的传感器特性曲线不是很一致，因此，每一个传感器需要矫正这里这个参数才能使测量值很准确。
//当发现测试出来的重量偏大时，增加该数值。
//如果测试出来的重量偏小时，减小改数值。
//该值可以为小数
#define GapValue 300
```

```
/*******************************************************************************
* 函 数 名         : HX711_Read
* 函数功能         : 读取 HX711 数据
* 输    入         : 无
* 输    出         : count
* 说    明         : 输出单位为克(g)
*******************************************************************************/
ulong HX711_Read (void)        //增益 128
{
        ulong count;
        uchar i;
        HX711_DOUT = 1;
        Delay_us (1);
        HX711_SCK = 0;
        count = 0;

        for (i=0; i<24; i++)
        {
                HX711_SCK = 1;
                count = count << 1;
                HX711_SCK = 0;
                if (HX711_DOUT)
                        count++;
        }
        HX711_SCK = 1;
      count = count^0x800000;//第 25 个脉冲下降沿来时，转换数据
        Delay_us (1);
        HX711_SCK = 0;
        return (count);
}

/*******************************************************************************
* 函 数 名         : GetWeight
* 函数功能         : 读取净重量
* 输    入         : 无
* 输    出         : 无
* 说    明         : 输出单位为克(g)
*******************************************************************************/
void GetWeight(void)
{
        NetWeight = HX711_Read();
        NetWeight = NetWeight - GrossWeight;                //获取净重
        if(NetWeight > 0)
```

```
	{
		NetWeight = (uint)((float)NetWeight/GapValue);	//计算实物的实际重量
	}
	else
	{
		NetWeight = 0;
	}
}

/**********************************************************************************
* 函 数 名         : GetGross
* 函数功能		   : 读取毛皮重量
* 输    入         : 无
* 输    出         : 无
* 说    明		   : 输出单位为克(g)
**********************************************************************************/
void GetGross(void)
{
	GrossWeight = HX711_Read();
}

//HX711.h:
#ifndef __HX711_H__
#define __HX711_H__

#include <stc15.h>
#include <intrins.h>
#include "delay.h"

/*****重定义关键词*****/
#ifndef uchar
#define uchar unsigned char
#endif

#ifndef uint
#define uint unsigned int
#endif

#ifndef ulong
#define ulong unsigned long
#endif

/***********************************
全局变量声明
```

```
***********************************/
extern ulong GrossWeight;
extern long NetWeight;

/***********************************
PIN 口定义
***********************************/
#define HX711_DOUT    P21
#define HX711_SCK    P20

/***********************************
函数声明
***********************************/
/*读取 HX711 数据*/
ulong HX711_Read (void);
/*获取净重量*/
void GetWeight (void);
/*获取毛皮重量*/
void GetGross (void);

#endif

//lcd.c:
#include "lcd.h"

/*******************************************************************************
* 函 数 名              : LcdWriteCom
* 函数功能              : 向 LCD 写入一个字节的命令
* 输    入              : com
* 输    出              : 无
*******************************************************************************/
void LcdWriteCom(uchar com)     //写入命令
{
        LCD1602_E = 0;        //使能
        LCD1602_RS = 0;      //选择发送命令
        LCD1602_RW = 0;      //选择写入

        LCD1602_DATAPINS = com;        //放入命令
        Delay_ms (1);                  //等待数据稳定

        LCD1602_E = 1;                    //写入时序
        Delay_ms (5);           //保持时间
        LCD1602_E = 0;
}
```

```
/*******************************************************************************
* 函 数 名              : LcdWriteData
* 函数功能              ： 向 LCD 写入一个字节的数据
* 输    入              : dat
* 输    出              ： 无
*******************************************************************************/
void LcdWriteData(uchar dat)                //写入数据
{
    LCD1602_E = 0;   //使能清零
    LCD1602_RS = 1; //选择输入数据
    LCD1602_RW = 0;//选择写入

    LCD1602_DATAPINS = dat; //写入数据
    Delay_ms (1);

    LCD1602_E = 1;    //写入时序
    Delay_ms (5);    //保持时间
    LCD1602_E = 0;
}

/*******************************************************************************
* 函 数 名          : LcdInit()
* 函数功能            ： 初始化 LCD 屏
* 输    入          ： 无
* 输    出          ： 无
*******************************************************************************/
void LcdWriteWord (uchar *s)
{
    while (*s>0)
    {
        LcdWriteData (*s);
        s++;
    }
}

/*******************************************************************************
* 函 数 名          : LcdInit()
* 函数功能            ： 初始化 LCD 屏
* 输    入          ： 无
* 输    出          ： 无
*******************************************************************************/
void LcdInit(void)                                  //LCD 初始化子程序
{
```

```
    LcdWriteCom (0x38);    //开显示
    LcdWriteCom (0x0c);    //开显示不显示光标
    LcdWriteCom (0x06);    //写一个指针加 1
    LcdWriteCom (0x01);    //清屏
    LcdWriteCom (0x80);    //设置数据指针起点
}

//lcd.h:
#ifndef __LCD_H_
#define __LCD_H_

/***********************************
包含头文件
***********************************/
#include <stc15.h>
#include "delay.h"

//---重定义关键词---//
#ifndef uchar
#define uchar unsigned char
#endif

#ifndef uint
#define uint unsigned int
#endif

/***********************************
PIN 口定义
***********************************/
#define LCD1602_DATAPINS P0
#define LCD1602_E P27
#define LCD1602_RW P25
#define LCD1602_RS P26

/***********************************
函数声明
***********************************/
/*LCD1602 写入 8 位命令子函数*/
void LcdWriteCom (uchar com);
/*LCD1602 写入 8 位数据子函数*/
void LcdWriteData (uchar dat);
/*LCD1602 写入字符串子函数*/
void LcdWriteWord (uchar *s);
/*LCD1602 初始化子程序*/
```

```
void LcdInit (void);

#endif

//ray.c:
#include "ray.h"

bit FlagFall_1 = 0;          //传感器 1 标志位
bit FlagFall_2 = 0;          //传感器 2 标志位
bit ReceiveScs = 0;              //接收成功标志位
bit ReceiveFal = 0;          //接收失败标志位
ulong TimeBuf = 0;               //存储间隔时间
ulong Buf = 0;                   //间隔时间的缓存
uchar Tmp = 0;                   //定时器 2 定时时间  temp*10 ms
uchar Cnt = 0;                   //坠物数量
double Height = 0;           //初始高度
float Distance = 0.15;       //两传感器间距
float RecvrHeight = 0.3;         //参考地面到近地传感器的距离


/*************************************************************************
* 函 数 名             : Timer1Init
* 函数功能              : 初始化定时器 1
* 输    入             : 无
* 输    出             : 无
* 说    明              : 定时 1ms
*************************************************************************/
void Timer1Init(void)            //1 毫秒@23.993MHz
{
    Buf = 0;             //清零时间缓存区
    AUXR |= 0x40;            //定时器时钟 1T 模式
    TMOD &= 0x0F;            //设置定时器模式
    TL1 = 0x47;          //设置定时初值
    TH1 = 0xA2;          //设置定时初值
    TF1 = 0;             //清除 TF1 标志
    TR1 = 1;             //定时器 1 开始计时
    ET1 = 1;             //使能定时器 1 中断
    EA = 1;
}

/*************************************************************************
* 函 数 名             : Timer1Server
* 函数功能              : 定时器 1 中断服务程序
* 输    入             : 无
```

```
* 输    出            : 无
* 说    明            : 计算通过两次传感器的时间差
*******************************************************************************/
void Timer1Server() interrupt 3
{
    TL1 = 0x47;           //设置定时初值
    TH1 = 0xA2;           //设置定时初值
    Buf++;
    if (Buf > 150)        //通过两传感器之间最大时间为 142.868ms
    {
        ReceiveFal = 1;           //接收失败
        FlagFall_1 = 0;           //重置传感器 1 标志位
        ET1 = 0;                  //关闭定时器 1 中断
    }
}

/*******************************************************************************
* 函 数 名          : GetHeight
* 函数功能          : 计算坠物初始高度
* 输    入          : 无
* 输    出          : 无
* 说    明          : 大量浮点运算，调用时尽量放在主程序最后
*******************************************************************************/
void GetHeight (void)
{
    double num = 0;
    double num1 = 0;
    double num2 = 0;
    double num3 = 0;
    double den = 0;
    double g = 9.7985;        //青岛重力加速度
    double t = 0;
    double Hm = 0;

    t = (double)TimeBuf / 1000;     //把时间统一为国际单位 s

    num1 = 4 * Distance * Distance;
    num2 = 4 * Distance * g * t * t;
    num3 = g * g * t * t * t * t;
    den = 8 * g * t * t;
    num = num1 - num2 + num3;

    if(den == 0)                    //防止开机时 Hm 溢出显示错误
    {
```

```
            Hm = 0;
        }
        else
        {
            Hm = num / den;
        }

        Height = Hm + Distance + RecvrHeight;       //计算初始高度 单位为 m
        Height = Height * 10000; //调整单位到 cm，精确到小数点后 2 位
}

/*****************************************************************************
* 函 数 名          : INT0Init
* 函数功能          : 初始化外部中断 0
* 输    入          : 无
* 输    出          : 无
* 说    明          : 传感器 1 使用，IO 口为 P32
*****************************************************************************/
 void INT0Init (void)
 {
      IT0 = 1;                          //设置 INT0 的中断类型 (1:仅下降沿 0:上升沿和下降沿)
     EX0 = 1;                          //使能 INT0 中断
     EA = 1;
 }

/*****************************************************************************
* 函 数 名          : INT0Init
* 函数功能          : 外部中断 0 服务函数：传感器 1 触发则把传感器 1 中断标志位置 1
* 输    入          : 无
* 输    出          : 无
* 说    明          : 传感器 1 使用，IO 口为 P32
*****************************************************************************/
void INT0Server() interrupt 0                   //中断入口
{
        SteerInit(90);                  //舵机转出
        Timer1Init();         //初始化定时器 1
        FlagFall_1 = 1;
        Cnt++;                                  //坠物个数加 1

}

/*****************************************************************************
* 函 数 名          : INT1Init
* 函数功能          : 初始化外部中断 1
```

```
* 输    入          : 无
* 输    出          : 无
* 说    明          : 传感器 2 使用，IO 口为 P33
**********************************************************************************/
 void INT1Init (void)
 {
     IT1 = 1;                        //设置 INT1 的中断类型 (1:仅下降沿 0:上升沿和下降沿)
    EX1 = 1;                         //使能 INT1 中断
    EA = 1;
 }

/**********************************************************************************
* 函 数 名          : INT1Init
* 函数功能          : 外部中断 1 服务函数：传感器 2 触发则把传感器 2 中断标志位置 1
* 输    入          : 无
* 输    出          : 无
* 说    明          : 传感器 2 使用，IO 口为 P33
**********************************************************************************/
void INT1Server() interrupt 2                //中断入口
{
    ET1 = 0;                    //关闭定时器 1 中断
    TimeBuf = Buf;
    FlagFall_2 = 1;
    FlagFall_1 = 0;
}

/**********************************************************************************
* 函 数 名          : SteerBack
* 函数功能          : 舵机收回
* 输    入          : 无
* 输    出          : 无
* 说    明          : 坠物接收标志位触发后收回舵机
**********************************************************************************/
void SteerBack (void)
{
    if (FlagFall_2 == 1)
    {
        Timer2Init();
        FlagFall_2 = 0;
    }
}

/**********************************************************************************
* 函 数 名          : Timer2Init
```

```
* 函数功能              : 初始化定时器 2
* 输    入              : 无
* 输    出              : 无
* 说    明              : 定时 10ms
*******************************************************************************/
void Timer2Init(void)              //10 毫秒@23.993MHz
{
    Tmp = 0;                        //计数标志
    AUXR &= 0xFB;                   //定时器时钟 12T 模式
    T2L = 0xE6;         //设置定时初值
    T2H = 0xB1;         //设置定时初值
    AUXR |= 0x10;               //定时器 2 开始计时
    IE2 = 0x04;         //开启定时器 2 中断位 ET2
}

/*******************************************************************************
* 函 数 名              : Timer1Server
* 函数功能              : 定时器 1 中断服务程序
* 输    入              : 无
* 输    出              : 无
* 说    明              : 计算通过两次传感器的时间差
*******************************************************************************/
void Timer2Server() interrupt 12
{
    T2L = 0xE6;         //设置定时初值
    T2H = 0xB1;         //设置定时初值
    Tmp++;
    if (Tmp > 100)      //定时 1s
    {
        SteerInit(0);
        IE2 = 0;      //关闭定时器 2 中断
        ReceiveScs = 1;
    }
}

//ray.h:
#ifndef __RAY_H_
#define __RAY_H_

/**********************************
包含头文件
**********************************/
#include <stc15.h>
#include "delay.h"
```

```
#include "steer.h"

//---重定义关键词---//
#ifndef uchar
#define uchar unsigned char
#endif

#ifndef uint
#define uint unsigned int
#endif

#ifndef ulong
#define ulong unsigned long
#endif

/***********************************
全局变量声明
***********************************/
extern double Height;
extern bit ReceiveScs;
extern bit ReceiveFal;
extern uchar Cnt;
extern ulong TimeBuf;

/***********************************
PIN 口定义
***********************************/
//P32 为传感器 1 使用的 INT0 口
//P33 为传感器 2 使用的 INT1 口

/***********************************
函数声明
***********************************/
/*初始化定时器 0 子函数*/
void Timer1Init (void);
/*定时器 0 中断服务子函数*/
void Timer1Server();
/*计算坠物初始高度子函数*/
void GetHeight (void);
/*初始化外部中断 0 子函数*/
void INT0Init (void);
/*外部中断 0 中断服务子程序*/
void INT0Server();
/*初始化外部中断 1 子函数*/
```

```
void INT1Init (void);
/*外部中断 1 中断服务子函数*/
void INT1Server();
/*判断接收成功并收回舵机子函数*/
void SteerBack (void);
/*定时器 2 初始化子程序*/
void Timer2Init(void);
/*定时器 2 中断服务子程序*/
void Timer2Server();

#endif

//voice.c:
#include "voice.h"

/*********************************************************************************
* 函 数 名          : ONE_LINE_VOL
* 函数功能          : 设置音量
* 输    入          : vol
* 输    出          : 无
* 说    明          : 无
*********************************************************************************/
void ONE_LINE_VOL (uchar vol)
{
    if (vol > 30)
      {
          vol = 30;
      }
    SendData (vol / 10);
    SendData (vol % 10);
    SendData (0x0C);
}

/*********************************************************************************
* 函 数 名          : ONE_LINE_PLAY_SONG
* 函数功能          : 播放指定曲目
* 输    入          : song
* 输    出          : 无
* 说    明          : 无
*********************************************************************************/
void ONE_LINE_PLAY_SONG (uchar song)
{
    uchar i;
    uchar tmp[3];
```

```
    tmp[0] = song / 100;
      song %= 100;
    tmp[1] = song / 10;
    tmp[2] = song % 10;

    SendData (0x0a);
    for(i=0; i<3; i++)
    {
        if (tmp[i] != 0)
        {
            SendData (tmp[i]);
        }
    }
    SendData (0x0b);
}

/*******************************************************************************
* 函 数 名          : SendData
* 函数功能          : 发送数据函数
* 输    入          : addr
* 输    出          : 无
* 说    明          : 无
*******************************************************************************/
void SendData (uchar addr)
{
    uchar i;

    EA = 0;/*发送时关掉中断，防止中断影响时序 */
    SDA = 1;/*开始拉高 */
    Delay_us (1000);

    SDA = 0;/*开始引导码*/
    Delay_us (2000);/*此处延时最少要大于 2ms，此参数延时为 310ms */
    for (i=0; i<8; i++ ) /*总共 8 位数据 */
    {
        SDA = 1;
        if (addr & 0x01) /*3:1 表示数据位 1,每个位用两个脉冲表示 */
        {
            Delay_us (1200);
            SDA = 0;
            Delay_us (400);
            }
            else /*1：3 表示数据位 0 ,每个位用两个脉冲表示 */
```

```
            {
             Delay_us (400);
             SDA = 0;
             Delay_us (1200);
          }
          addr >>= 1;
      }
      SDA = 1;
      EA = 1;//恢复中断
}

/*******************************************************************************
* 函 数 名          : VoiceScs
* 函数功能          : 播报接收成功子函数
* 输    入          : 无
* 输    出          : 无
* 说    明          : 无
*******************************************************************************/
void VoiceScs (void)
{
        ulong height_voice[6] = 0;                      //存放初始高度
        uint cnt_voice[2] = 0;                          //存放坠物计数
        uint i = 0;

        ONE_LINE_VOL (30);                              //设置音量
      Delay_ms (20);                            //一条指令发送完毕之后，需要间隔 20ms 在发送下一条指令

        ONE_LINE_PLAY_SONG (16);                        //播放“接收成功”
        Delay_ms (600);                                                         //2500

        height_voice[5] = (long)(Height / 1) % 10;    //个位
        height_voice[4] = (long)(Height / 10) % 10;     //十位
        height_voice[3] = (long)(Height / 100) % 10;    //百位
        height_voice[2] = (long)(Height / 1000) % 10;         //千位
        height_voice[1] = (long)(Height / 10000) % 10;         //万位
        height_voice[0] = (long)(Height / 100000) % 10;             //十万位

        for(i=0;i<6; )                                  //找到第一个数字位置
        {
              if (height_voice[i] == 0)           //指向有效数字
              {
                    i++;
              }
              else
```

```
        {
            break;
        }
    }

    for ( ; i<6; i++)                                   //播报数字 0-9
    {
        switch (height_voice[i])
        {
            case (0):     ONE_LINE_PLAY_SONG (11);
            break;
            case (1):     ONE_LINE_PLAY_SONG (1);
            break;
            case (2):     ONE_LINE_PLAY_SONG (2);
            break;
            case (3):     ONE_LINE_PLAY_SONG (3);
            break;
            case (4):     ONE_LINE_PLAY_SONG (4);
            break;
            case (5):     ONE_LINE_PLAY_SONG (5);
            break;
            case (6):     ONE_LINE_PLAY_SONG (6);
            break;
            case (7):     ONE_LINE_PLAY_SONG (7);
            break;
            case (8):     ONE_LINE_PLAY_SONG (8);
            break;
            case (9):     ONE_LINE_PLAY_SONG (9);
            break;
            default:      break;
        }

        if (i == 1)
        {
            Delay_ms (100);                                 //
            ONE_LINE_PLAY_SONG (14);        //播报“百”
            Delay_ms (100);                                 //
            if((height_voice[2] == 0) && (height_voice[3] == 0))        //防止十位个位为零  读出零的情况
            {
                i = i + 2;
            }

        }
        else if((i == 2) && (height_voice[2] != 0))      //十位数字不为零时
```

```
                {
                    Delay_ms (200);
                    ONE_LINE_PLAY_SONG (19);          //播报“十”
                    Delay_ms (100);
                    if (height_voice[3] == 0)
                    {
                        i++;
                    }
                }
            Delay_ms (100);

            if (i == 3)          //小数点
            {
                Delay_ms (100);
                ONE_LINE_PLAY_SONG (12);
                Delay_ms (200);
            }
            else if (i == 5)       //厘米
                {
                    Delay_ms (200);
                    ONE_LINE_PLAY_SONG (13);
                    Delay_ms (200);
                }
    }
}

/*******************************************************************************
* 函 数 名             : VoiceFal
* 函数功能              ：播报接收失败子函数
* 输    入             ：无
* 输    出             ：无
* 说    明              ：无
*******************************************************************************/
void VoiceFal (void)
{
    ONE_LINE_VOL(30);                           //设置音量
   Delay_ms(20);                          //一条指令发送完毕之后，需要间隔 20ms 在发送下一条指令
    ONE_LINE_PLAY_SONG(15);                     //播放“接收失败”
    Delay_ms(1000);
}

//voice.h:
#ifndef __VOICE_H_
#define __VOICE_H_
```

```
/**********************************
包含头文件
**********************************/
#include <stc15.h>
#include "delay.h"

//---重定义关键词---//
#ifndef uchar
#define uchar unsigned char
#endif

#ifndef uint
#define uint unsigned int
#endif

#ifndef ulong
#define ulong unsigned long
#endif


/**********************************
PIN 口定义
**********************************/
#define SDA P10

/**********************************
外部变量声明
**********************************/
extern double Height;
extern uchar Cnt;

/**********************************
函数声明
**********************************/
/*发送数据函数*/
void SendData (uchar addr);
/*设置音量子函数*/
void ONE_LINE_VOL (uchar vol);
/*播放指定曲目子函数*/
void ONE_LINE_PLAY_SONG (uchar song);
/*播报接收成功子函数*/
void VoiceScs (void);
/*播报接收失败子函数*/
```

```
void VoiceFal (void);

#endif
```