

毕业设计助手应用实例-模拟电子秤

一：引言

一篇好的毕业论文是人生成长的里程碑,是对学业肯定,是就业的敲门砖。在经验不足,基础知识不扎实、动手能力差的前提下,如何写一篇毕业设计论文?如何快速写一篇毕业论文?如何写好一篇毕业论文?如何用最短的时间做出论文的实物?做一篇毕业论文给人一种“蜀道之难,难于上青天”的感觉;

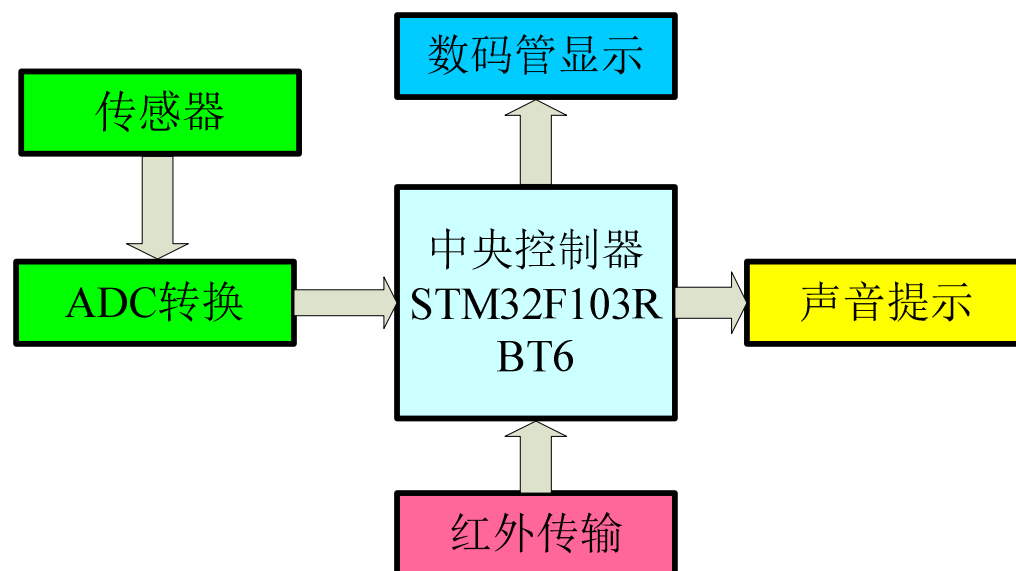
今天大家再也不用为写毕业论文发愁了,针对高校毕业生毕业设计困难的问题,北京光轮电子依据独立研发的 TreeOS 操作系统和自动代码软件 AlphaMCU 设计出毕业设计助手;这款产品将毕业论文中用到频率高的模块驱动代码加以整理;利用公司专利 AlphaMCU 自动将选用模块驱动呈现给用户;我们的产品有以下亮点:

※ 我们的提供常用模块驱动,驱动都是通过长期验证并加以详细注释;你不用再从网上抓一包全是“坑”的代码,你不用再煞费苦心的去读别人“光秃秃”的代码,让你站在巨人的肩膀上远眺,繁重和劳累的工作我们工程师都给你做好了;

※ TreeOS 是我们自主研发的经过 200 多个项目验证的无核构件化的操作系统;无核代表啥?代表这款操作系统不会占用系统紧张宝贵的资源;

※ 毕业设计助手提供设备驱动接口,你不用花老半天时间调个 iic,让串口配置折磨的面容憔悴;不再会为移植一个别人的通信协议,设备驱动感觉无从下手。。。。。

你只要掌握其中的几个点,就能很快的上手。关于产品的使用说明大家可以上网站查找,“纸上得来终觉浅,绝知此事要躬行”。现在想通过一个小小的项目案例-模拟电子秤,给大家展示一下毕业设计助手使用方法。模拟电子秤框图如下所示:



二：项目简介

主控芯片是 STM32F103RBT6; 称重传感器将物体的压力信号转换成模拟信号。模拟信号通过 ADC0804 (一款 8 位、单通道、低价格 A/D 转换器 AD) 转换成数字信号, 数字信号通过数码管显示重量数值, 通过红外更改设备的编号, JQ8900-16P 是串口控制的语音模块提示用户的一些操作。

三：设计步骤

1、登录光轮电子的官方网站 (<http://www.treeos.com/>)，如果你是新用户需要注册账号，在我们公司产品栏中打开毕业设计助手；

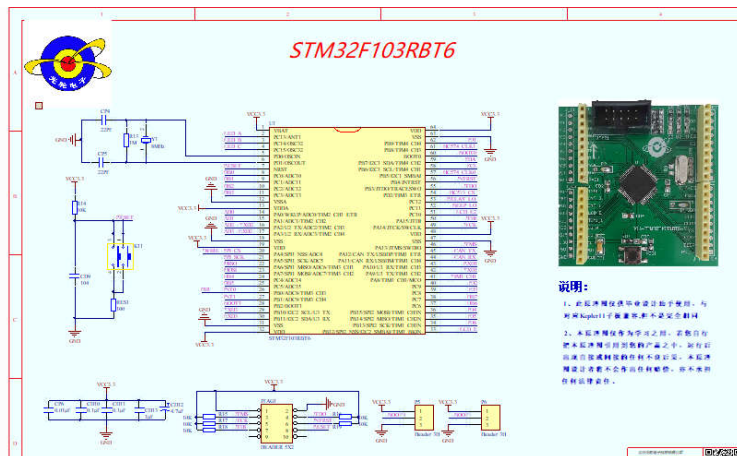


2、到毕业设计助手界面中下载毕业设计助手 AD 工程。至于产品的使用方法大家参见公司网站上的其他文章；

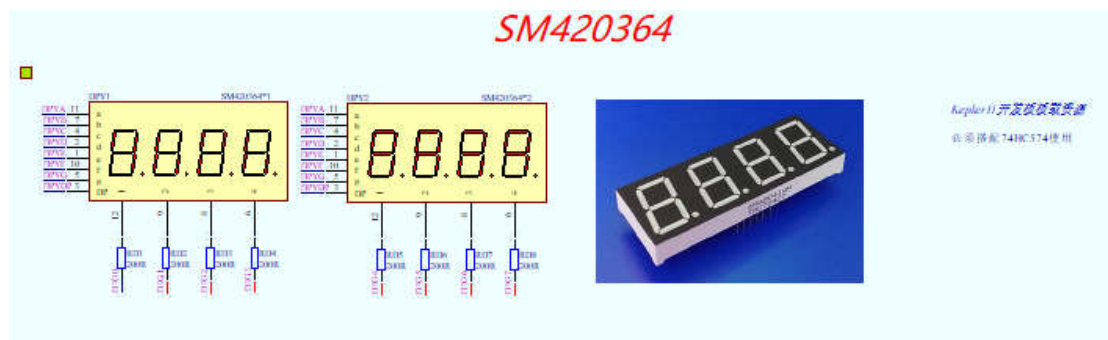


3 通过正版的 AD 软件打开当前的工程。我们使用了 AD 软件的屏蔽罩功能，处于屏蔽状态下的功能模块不被编译，打开屏蔽功能的时候这些模块被默认添加进来。

3.1 首先进行器件的选型，找到我们设计中使用的模块，并且将器件的屏蔽罩打开：



3.2 依次通过这种方法打开我们使用的全部模块，将我们要使用的模块全部打开屏蔽罩，使其处于可控状态。



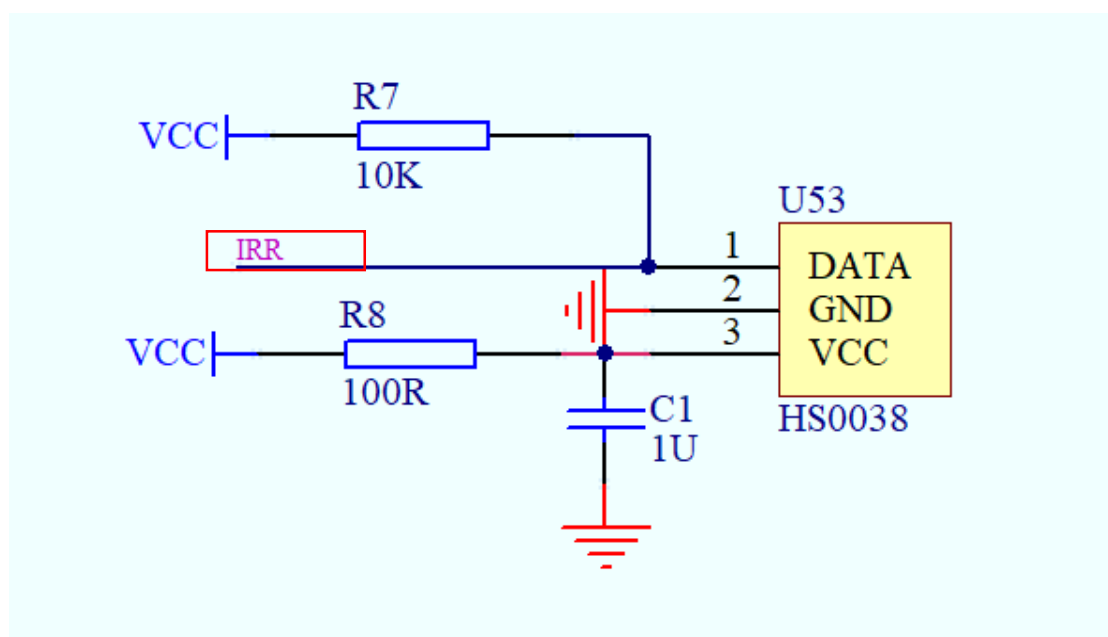
通过以上步骤我们的原理图模块设计已经完毕。快吧？意想不到吧。我们只要轻松的点一下鼠标，将我们想要的模块就添加进来了，省去了你花费相当长的时间去绘制模块的封装，然后进行原理图添加。如果中间添加有问题还会出现难以查找的错误，我们的功能模块都是有现成的原理图库，用户完全可以放心使用，这就减少了我么很多不必要的时间浪费。

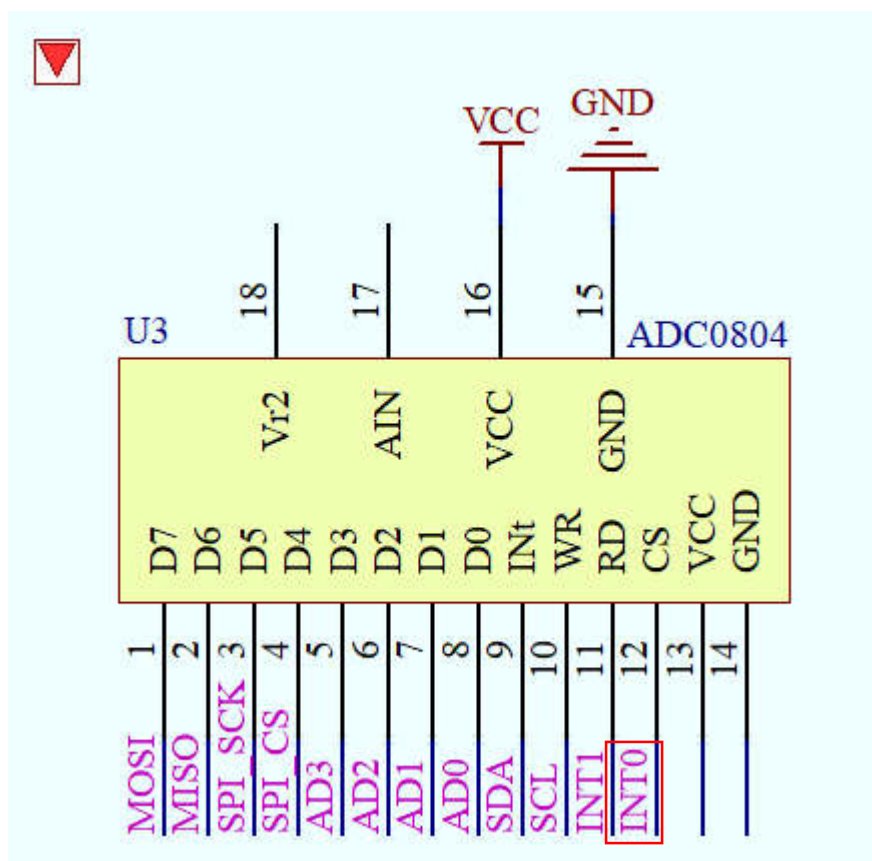
4、接下来就是更改原理图的电气连接关系。请大家看明白了，我们只是更改。我们的单片机引脚默认和模块的引脚存在电气连接关系。如果没有管脚冲突希望大家还是使用我们默认的引脚连接关系。这些电气连接关系都是我们工程师兼顾方方面面的考虑来设计的，本项目出现了管脚冲突的情况，我们就要给出相应更改。

在这个项目中红外接收引脚和 ADC0804 驱动引脚同时用到了 PB0、PB0 要同时设置成上拉输入和推挽输出功能，这肯定不行。鉴于我们红外捕获功能使用了单片机内部的定时器来完成，定时器的输入捕获通道必须挂载到 PB0 口上。我们只能调节 ADC0804 的控制功能引脚，控制功能引脚随便配置。我们将 ADC0804 的片选引脚进行调整到 PC0。

实验进行到这里，大家是否觉得有种“衣带渐宽终不悔，为伊消得人憔悴”的感觉？没事，成功已经尽在眼前了。

	DB5	25	PC5/ADC15
IRR	INT0	26	PB0/ADC8/TIM3_CH3
	INT1	27	PB1/ADC9/TIM3_CH4





5、引脚调整完毕后案检查一遍，编译工程选择生成.NET 文件，将我们生成的 NET 文件加载到我们的毕业设计助手的 AlphaMCU 中，点击选择文件加载我们的 NET 文件，然后点击输出代码，我们工程代码和资料包就生成了！我们将这个工程文件下载下来，解压打开这个工程：

※STM32F103RBT6 是这个项目的工程文件；

※资料里面包含了我们的模块技术资料的百度文库链接；

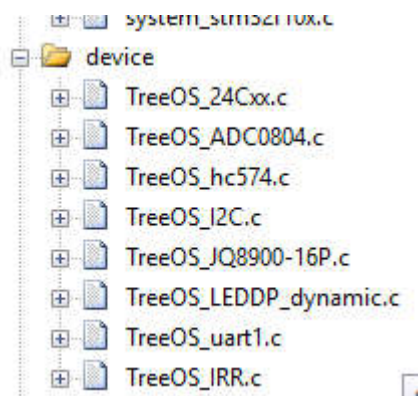
※论文信息给出了这个设计的参考论文；

名称	修改日期	类型	大小
STM32F103RBT6	2018/1/20 14:54	文件夹	
资料	2018/1/20 10:06	文件夹	
论文信息1516413944087	2018/1/20 10:05	Microsoft Excel ...	55 B

AlphaMCU输入电路图 自动生成代码



6、打开 KEIL 工程后我们要加载我们常用的外设的一些模块资料。添加完成之后我们编译一下，编译一遍成功，整个项目的代码工程就搞定了。这样一来我们的驱动代码就写好了，我觉得让一个初学来写这些驱动要花费几天时间：



7、用户代码是这些模块之间的逻辑组合以及接口函数调用，最终实现用户的功能。

我们调整初始化外设的顺序并且加入我们的初始化场景，上电首先初始化外部存储设备的 IIC 端口，读取存入存储器件中的设备的 ID 。然后将读取到的 ID 放置在数码管的段码显示位置，初始化语音播报功能，播放一段欢迎使用的语音。之后进行数码管驱动的初始化，ADC 模块的初始化，红外捕获功能初始化，然后播放一段文字说明设备已经就绪。播放文字中间加一个较长时间的延时。


```

//=====devices周边设备初始化=====

at24cxx_init();           //AT24C系列芯片初始化
at24cxx_rcv_data(0x0000, 1, &Device_ID_EE);
LEDDP_init();           //数码管初始化
LEDDP_buff[1] = seg7_num[Device_ID_EE];
JQ8900_init();          //语音模块初始化
JQ8900_Play((ui8*)&Song_wlcome, sizeof(Song_wlcome));
hc574_init();           //hc574使用的IO口初始化
hc574_output_init();    //开机后, 再使能所有74HC574
ADC0804_Init();        //ADC0804初始化
IRR_cap_init();
delayms(5000);
JQ8900_Play((ui8*)&Song_ready, sizeof(Song_ready));
delayms(5000);
//=====

```

编写设备的组合逻辑, 在循环中要处理的问题, 扫描按键数值的变化更新当前 ID:

```

} {
//红外遥控器扫描
IRR_scan_key();

//-----用IR代码来识别-----
if (IRRkeyval)
{
switch(IRRkeyval)
{
case IRR_KEY0 : { LEDDP_buff[1] = seg7_num[0]; Device_ID=0; break;};
case IRR_KEY1 : { LEDDP_buff[1] = seg7_num[1]; Device_ID=1; break;};
case IRR_KEY2 : { LEDDP_buff[1] = seg7_num[2]; Device_ID=2; break;};
case IRR_KEY3 : { LEDDP_buff[1] = seg7_num[3]; Device_ID=3; break;};
case IRR_KEY4 : { LEDDP_buff[1] = seg7_num[4]; Device_ID=4; break;};
case IRR_KEY5 : { LEDDP_buff[1] = seg7_num[5]; Device_ID=5; break;};
case IRR_KEY6 : { LEDDP_buff[1] = seg7_num[6]; Device_ID=6; break;};
case IRR_KEY7 : { LEDDP_buff[1] = seg7_num[7]; Device_ID=7; break;};
case IRR_KEY8 : { LEDDP_buff[1] = seg7_num[8]; Device_ID=8; break;};
case IRR_KEY9 : { LEDDP_buff[1] = seg7_num[9]; Device_ID=9; break;};

default:break;
}
}

```

这是红外按键解析处理函数, 解析得到设备的 ID:

```

if(Device_ID!=Device_ID_EE)
{
at24cxx_send_data(0x0000, 1, &Device_ID);
}

```

以上是当设备的 ID 变化的时候将这个 ID 存储到 EEPROM 中。

循环中采样 AD, 然后进行滤波取平均值。

```

Start_Adc();

tempdata[adc_cnt++]=ADC0804_Read();
if(adc_cnt>=5)
{
    adc_cnt=0;
    adc_data=Data_Average((ui32 *)&tempdata, 5);
    data = adc_data*100;
}
    
```

1 秒钟更新一下当前的显示:

```

)   if(second_flag) //判断秒标志是否置位
)   {
)       second_flag=0; //清除秒标志
)       sec_count++;
)       LEDDP_buff[4] = seg7_num[adc_data/SAMPLE_UNIT] | seg7_dot;
)       LEDDP_buff[5] = seg7_num[data/SAMPLE_UNIT/10%10];
)       LEDDP_buff[6] = seg7_num[data/SAMPLE_UNIT%10];
    
```

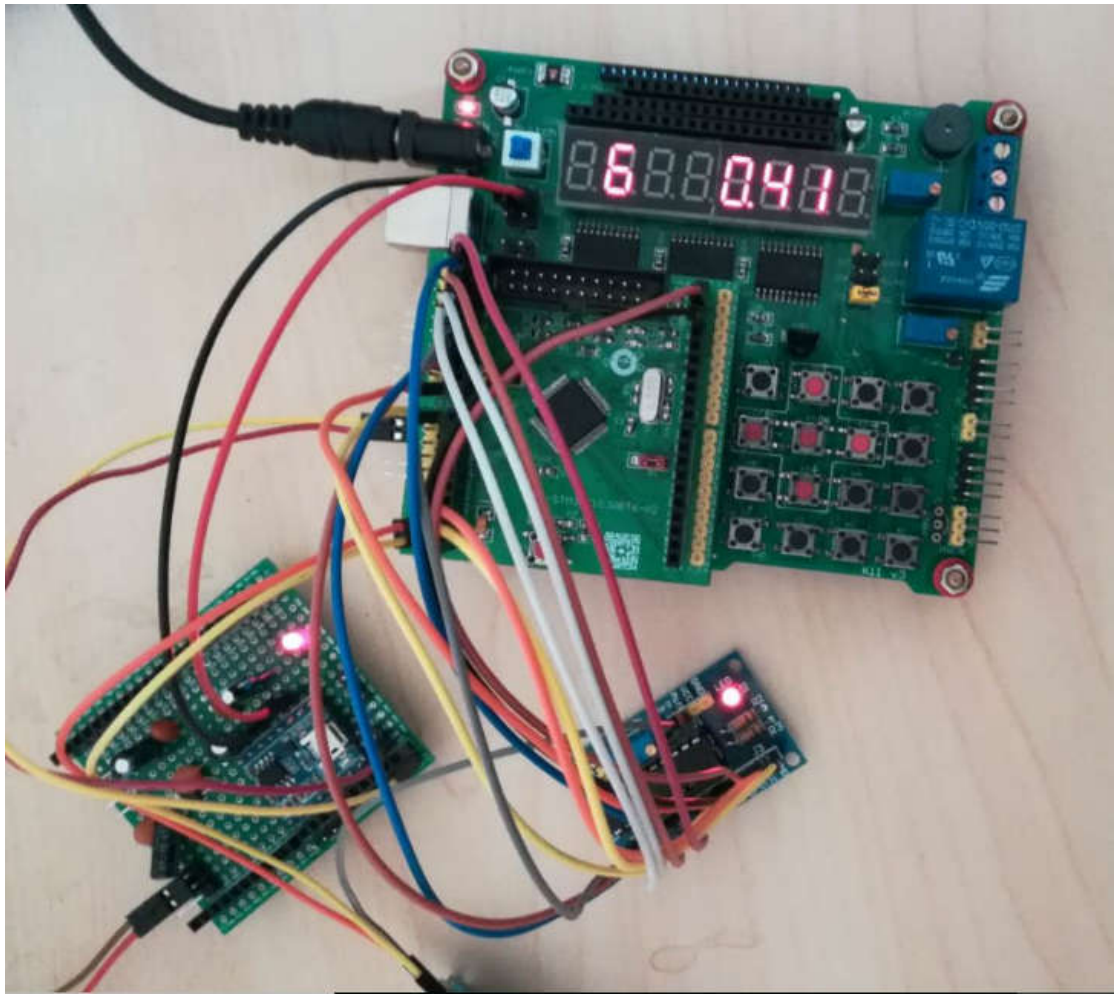
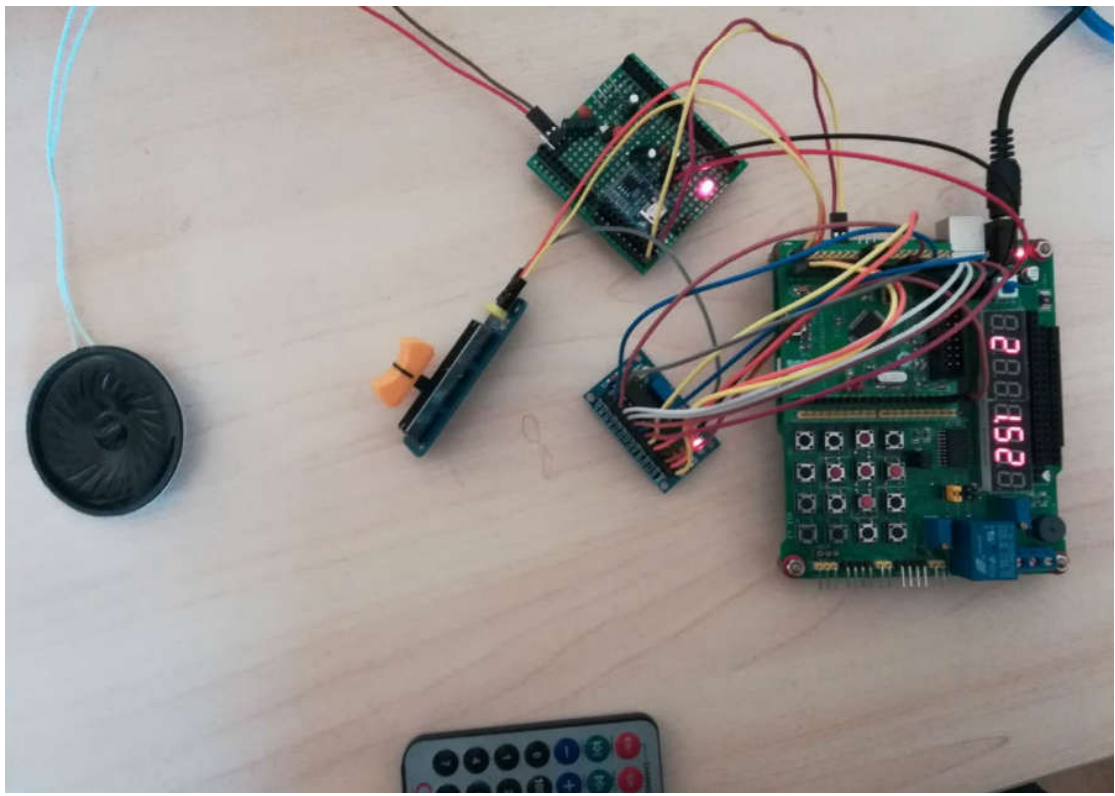
到达语音播报刷新的时间后进行语音的刷新:

```

        if(sec_count>=FILTER_CNT)//设置语音播报刷新的间隔
        {
            sec_count=0;
            if(data/SAMPLE_UNIT<ADC_MIN_VALUE)//重量太轻
            {
                JQ8900_Play((ui8*)&Song_min, sizeof(Song_min));
            }
            else if(data/SAMPLE_UNIT>ADC_MAX_VALUE)//重量太重
            {
                JQ8900_Play((ui8*)&Song_max, sizeof(Song_max));
            }
            else//重量测量已经上传
            {
                JQ8900_Play((ui8*)&Song_result, sizeof(Song_result));
            }
        }
    }
}
    
```

四：硬件实物

下面两张图是模拟电子秤在 K11 开发板上的实现照片:



五：代码量对比

TreeOS_24Cxx.C 和.H 文件共:	229 行;
TreeOS_ADC0804.c 和 TreeOS_ADC0804.h	201 行
TreeOS_hc574.c 和 TreeOS_hc574.h	250 行
TreeOS_I2C.c 和 TreeOS_I2C.h	244 行
TreeOS_IRR.c 和 TreeOS_IRR.h	352 行
TreeOS_JQ8900-16P.c 和 TreeOS_JQ8900-16P.h	309 行
TreeOS_LEDDP_dynamic.c 和 TreeOS_LEDDP_dynamic.h	234 行

统自动生成代码：1961 行

用户代码：142 行

用户代码占得比例：7.2%;

到此我们这个项目就完全搞定了。通过用户代码占的比例分析，毕业设计助手的优势一目了然。今天毕业设计助手作为一个新生事物如同星星之火，我相信明天他会成为天空之皓月放射出万丈光芒。。。