

1186 普通 LCD 人体秤模板

使用说明与注意事项

REV1.0

深圳市西城微科电子有限公司

地址：深圳市福田区新闻路深茂商业中心 26B

电话：0755-83063040 传真：0755-83065035

网站：<http://www.sictech.com.cn>

版本历史

历史版本	修改内容	版本日期
REV 1.0	初始版本	2019 年 1 月 4 日

目录

第一章	初识模板	5
1.1	模板程序介绍	5
1.2	程序规范相关	5
1.2.1	子程序命名：	5
1.2.2	变量命名：	5
1.3	工程文件组织	6
1.3	流程图	6
第二章	配置文件	6
2.1	配置文件介绍	6
2.2	如何修改配置文件	6
2.3	注意事项	6
第三章	库文件	7
3.1	引入库情况	7
3.2	库文件说明及使用方法	7
	● Fun_Math_Add2_2	7
	● Fun_Math_Sub2_2	7
	● Fun_Math_Sub2_2_Neg	7
	● Fun_Math_Sub3_3	8
	● Fun_Math_Sub3_3_Neg	8
	● Fun_Math_Div6_3	8
	● Fun_Math_Div6_3_Rounded	9
	● Fun_Math_Mul3_3	9
	● Fun_Math_Hex3_Bcd	10
	● Fun_ProcAdc	10
	● Fun_ProcAdcStart	10
	● Fun_SetZeroPoint	10
	● Fun_SetCountZero	11
	● Fun_CurAD_Sub_ZeroAD	11
	● Fun_ScanWeihgt	11
	● Fun_GetAutoOnADC	11
	● Fun_GetCount	11
	● Fun_key_Scan	11
	● Fun_3W_CAL_Init	11
	● Fun_3W_CAL_Disable	11
	● Fun_3W_CAL_ResetLock	11
	● Fun_3W_CAL	12
	● Fun_Delay_100MS	12
	● Fun_Delay_40MS	12
	● Fun_Delay_20MS	12
	● Fun_Delay_10MS	12
	● Fun_Delay_1MS	12
	● Fun_power_Init	12

● Fun_power_Close	12
● Fun_GPIO_Init.....	13
● Fun_ADC_Init	13
● Fun_ADC_Close.....	13
● Fun_TIMER_init.....	13
● Fun_TIMER_close.....	13
● Fun_RAM_Zero.....	13
● Table_Lcd_Num	13
● Fun_LCD_Init.....	13
● Fun_LCD_Close	13
● Fun_LCD_Load	14
● Fun_OTP_READ_CAL.....	14
● Fun_OTP_WRITE_CAL	14
第四章 应用层	14
4.1 GPIO.....	14
4.2 系统 TIMER.....	14
4.3 系统时钟.....	14
4.4 看门狗	15
4.5 LCD 显示.....	15
4.5.1 LCD 显示驱动原理	15
4.5.2 如何根据 LCD 逻辑图来定义段码位图	15
4.6 按键.....	15
4.7 重量开机.....	15
4.8 标定.....	16
4.9 ADC	16
4.10 重量点数	16
4.11 默认单位	16
4.12 外部中断	17
4.13 低电压检测.....	17
第五章 扩展与兼容性	17

第一章 初识模板

1.1 模板程序介绍

该模板工程以芯海科技 8 BIT RISC SOC (CSU8RP1186) 为载体实现了一个普通 LCD 人体秤功能，同时模板工程将部分代码封装成库的形式供应用逻辑层调用，工程中相近功能的代码文件以模块化的形式组织，方便后期的开发与维护工作。

1.2 程序规范相关

考虑到后期开发的不可可能由专人来维护指定架构风格的代码，现对代码做一定程序的规范。

1.2.1 子程序命名：

以 Fun_xx_yy 形式

Fun 表示要实现某一功能的方法，

xx 表示这个方法隶属于哪个类别，例如 LCD，ADC，OTP 等

yy 表示这类别要执行的具体功能概述，例如 init,close,write,read 等

例：LCD 初始化子程序名字的写法，Fun_LCD_Init。

1.2.2 变量命名：

变量的定义没有子程序那样的严格，只要遵循一个原则：看到变量名就可明白这个变量目的，例如零点，根据其中长度一般为 3 字节，我们用汇编表示为 ZeroH、ZeroM、ZeroL。

另外要特别说明一下位定义的要求，由于随着程序功能的增加，标志位的使用也会成倍增加，标志位使用错乱的可能就提高了，A 变量中标志位 xx，可能在使用时却把 B 变量弄成操作数 1，例如 BSF B,xx，这样程序编译不会报错一运行就不正常了。

于是我们寻找了一种方法，当然还有更好的方法。我们拿主流程变量 SysFlow 来说，在它下面有两个标志位，一个用来表示上电处理流程命名为 B_SysFlow_PWR，一个用来表示秤体流程命名为 B_SysFlow_SCALE。如图 1

```

SysFlow          DS 1
    B_SysFlow_PWR EQU 0
    B_SysFlow_SCALE EQU 1
  
```

图 1 SysFlow 及标志位定义

拿 B_SysFlow_PWR 来说，其中

B：表示这是一个位定义

SysFlow: 表示它隶属于哪个变量

PWR：表示上电

写成这样的形式主要是为了后期查找 BUG 时的方便，比如我们由于其它原因把本意要修改的标志位写在了其它的变量名下，如 BSF ScaleFlow,B_SysFlow_PWR,当我们在运行调试时发现程序没有按我们预期设定的那样运行，于是当我们看到这行语句时，我们会发现标志位与隶属变量不是同一个，这样就解决了一个 BUG。

1.3 工程文件组织

图 2 列出了 IDE 工程文件的分布组织，

主要分 4 个部分：头文件，主流程，用户区，秤模块

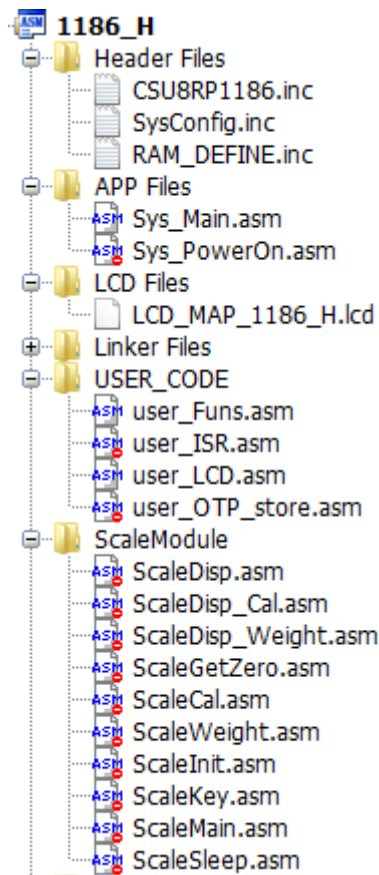


图 2 工程文件分布组织

1.3 流程图



1186_LCD普通人体秤模板_流程图.pdf



PowerOn子流程图.pdf



ScaleModule子流程图.pdf

上面三个文档为整个工程模板的流程图

第二章 配置文件

2.1 配置文件介绍

文件” SysConfig.inc “里面是一些宏定义，用于初始化对应的片内外设与其它功能模块的参数设定

2.2 如何修改配置文件

后面的章节会对各部的宏定义作详细介绍

2.3 注意事项

后面的章节会对各部的宏定义作详细介绍

第三章 库文件

3.1 引入库情况

目前引入的库主要封装了不常改动但是必须的代码段。

3.2 库文件说明及使用方法

- Fun_Math_Add2_2

功能描述：两字节长度的变量加法，TempRam3、4 += TempRam5、6

输入：操作数 1：TempRam3(高 8 位)、TempRam4 (低 8 位)

操作数 2：TempRam5(高 8 位)、TempRam6 (低 8 位)

输出：操作数 1：TempRam3(高 8 位)、TempRam4 (低 8 位)

例：1000 = 112 + 888

```
MVL    LOW 112
MOVWF  TempRam4
MVL    HIGH 112
MOVWF  TempRam3
MVL    LOW 888
MOVWF  TempRam6
MVL    HIGH 888
MOVWF  TempRam5
CALL   Fun_Math_Add2_2
```

上面的代码段执行完后，TempRam3 中值为 03H，TempRam4 中值为 E8H，十六进制 03E8H 的十进制为 1000。

- Fun_Math_Sub2_2

功能描述：两字节长度的变量减法，TempRam3、4 -= TempRam5、6

输入：操作数 1：TempRam3(高 8 位)、TempRam4 (低 8 位)

操作数 2：TempRam5(高 8 位)、TempRam6 (低 8 位)

输出：操作数 1：TempRam3(高 8 位)、TempRam4 (低 8 位)

例：776 = 888 - 112

```
MVL    LOW 888
MOVWF  TempRam4
MVL    HIGH 888
MOVWF  TempRam3
MVL    LOW 122
MOVWF  TempRam6
MVL    HIGH 122
MOVWF  TempRam5
CALL   Fun_Math_Sub2_2
```

上面的代码段执行完后，TempRam3 中值为 03H，TempRam4 中值为 08H，十六进制 0308H 的十进制为 776。

- Fun_Math_Sub2_2_Neg

功能描述：两字节长度的变量减法后，如果需要取这两个差的绝对值，调用这个子程序，必须在调用 Fun_Math_Sub2_2 后再调用 Fun_Math_Sub2_2_Neg

例：776 = |112 - 888|

```
MVL    LOW 112
```

```
MOVWF  TempRam4
```

```
MVL    HIGH 112
```

```
MOVWF  TempRam3
```

```
MVL    LOW 888
```

```
MOVWF  TempRam6
```

```
MVL    HIGH 888
```

```
MOVWF  TempRam5
```

```
CALL    Fun_Math_Sub2_2
```

```
CALL    Fun_Math_Sub2_2_Neg
```

上面的代码段执行完后，TempRam3 中值为 03H，TempRam4 中值为 08H，十六进制 0308H 的十进制为 776

- Fun_Math_Sub3_3
同 Fun_Math_Sub2_2，只不过是操作数为 3 字节长度

- Fun_Math_Sub3_3_Neg
同 Fun_Math_Sub2_2_Neg，只不过是操作数为 3 字节长度

- Fun_Math_Div6_3
功能描述：六字节长度 / 三字节长度变量除法,不带四舍五入

输入：

操作数 1：TempRam1~ TempRam6

操作数 2：TempRam11 (高 8 位)、TempRam12 (中 8 位)、TempRam13 (低 8 位)

输出：

商：TempRam4(高)~TempRam6 (低)

余数：TempRam1(高)~TempRam3 (低)

例：1000/256 = 3 ... 232

```
MVL    LOW 1000
```

```
MOVWF  TempRam6
```

```
MVL    HIGH 1000
```

```
MOVWF  TempRam5
```

```
CLRF   TempRam4
```

```
CLRF   TempRam3
```

```
CLRF   TempRam2
```

```
CLRF   TempRam1
```

```
MVL    LOW 256
```

```
MOVWF  TempRam13
```

```
MVL    HIGH 256
```

```
MOVWF  TempRam12
```

```
CLRF   TempRam11
```

```
CALL    Fun_Math_Mul3_3
```

上面的代码段执行完后，

商：TempRam4 中值为 00H，TempRam5 中值为 00H，TempRam6 中值为 03H

十六进制 000003H 的十进制为 3

余数：TempRam1 中值为 00H, TempRam2 中值为 00H, TempRam3 中值为 E8H

十六进制 0000E8H 的十进制为 232

- Fun_Math_Div6_3_Rounded

功能描述：调用六字节长度 / 三字节长度变量除法后，如果需要四舍五入时调用这个子程序

例：1000/256 = 3 ... 232

```
MVL      LOW 1000
MOVWF    TempRam6
MVL      HIGH 1000
MOVWF    TempRam5
CLRF     TempRam4
CLRF     TempRam3
CLRF     TempRam2
CLRF     TempRam1
MVL      LOW 256
MOVWF    TempRam13
MVL      HIGH 256
MOVWF    TempRam12
CLRF     TempRam11
CALL     Fun_Math_Mul3_3
CALL     Fun_Math_Div6_3_Rounded
```

上面的代码段执行完后，

商：TempRam4 中值为 00H, TempRam5 中值为 00H, TempRam6 中值为 04H

十六进制 000004H 的十进制为 4

余数：TempRam1 中值为 00H, TempRam2 中值为 00H, TempRam3 中值为 E8H

十六进制 0000E8H 的十进制为 232

- Fun_Math_Mul3_3

功能描述：三字节长度的变量乘法，

输入：

操作数 1：TempRam4(高 8 位)、TempRam5 (中 8 位)、TempRam6 (低 8 位)

操作数 2：TempRam11(高 8 位)、TempRam12 (中 8 位)、TempRam13 (低 8 位)

输出：

TempRam1(高)~TempRam6 (低)

例：99456 = 888 * 112

```
MVL      LOW 888
MOVWF    TempRam6
MVL      HIGH 888
MOVWF    TempRam5
MVL      00H
MOVWF    TempRam4
```

```

MVL      LOW 122
MOVWF    TempRam13
MVL      HIGH 122
MOVWF    TempRam12
MVL      00H
MOVWF    TempRam11
CALL     Fun_Math_Mul3_3

```

上面的代码段执行完后，

TempRam1 中值为 00H, TempRam2 中值为 00H, TempRam3 中值为 00H, TempRam4 中值为 01H,

TempRam5 中值为 84H, TempRam6 中值为 80H

十六进制 018480H 的十进制为 99456

- Fun_Math_Hex3_Bcd

功能描述：三字节的变量转换成 6 位的 BCD 码，主要用于显示查表工作

输入：

操作数 1：TempRam11 (高 8 位)、TempRam12 (中 8 位)、TempRam13 (低 8 位)

输出：

TempRam1(高)~TempRam6 (低)

例：01E240H 转换在 6 位的 BCD 码为 123456

```

MVL      040H
MOVWF    TempRam13
MVL      0E2H
MOVWF    TempRam12
MVL      001H
MOVWF    TempRam11

```

```
CALL     Fun_Math_Hex3_Bcd
```

上面的代码段执行完后，

TempRam1 中值为 01H, TempRam2 中值为 02H, TempRam3 中值为 03H

TempRam4 中值为 04H, TempRam5 中值为 05H, TempRam6 中值为 06H

- Fun_ProcAdc

功能描述：ADC 滤波器

输入：无

输出：无

这个子程序主要是将 ADC 输出的信号量进行处理，用户不用关心内部操作

- Fun_ProcAdcStart

功能描述：复位 ADC 滤波器并开始运行

- Fun_SetZeroPoint

功能描述：刷新零点，同时调用 Fun_SetCountZero

输入：无

输出：无

- Fun_SetCountZero
功能描述：秤台重量低于最小锁定重量时调用
输入：无
输出：无
- Fun_CurAD_Sub_ZeroAD
功能描述：当前滤波器输出 ADC 减去零点 ADC
输入：无
输出：无
- Fun_ScanWeihgt
功能描述：重量开机检测，秤处于低功耗模式时 1 秒调用一次。
输入：无
输出：标志位【SysFlag1,B_SysFlag1_WakeUp】，如果为 1 表明重量开机。
- Fun_GetAutoOnADC
功能描述： 计算开机重量
输入：无
输出：无
- Fun_GetCount
功能描述： 将 ADC 内码转换成重量
输入：无
输出：CountH, CountL
- Fun_key_Scan
功能描述： 按键长短按处理,输出按键长短按状态
输入：Key_IO_PRESS
输出：Key_TRG,Key_HOLD
- Fun_3W_CAL_Init
功能描述： 使能并初始化重量标定
输入：无
输出：无
- Fun_3W_CAL_Disable
功能描述： 关闭重量标定
输入：无
输出：无
- Fun_3W_CAL_ResetLock
功能描述： 解除锁定调用处理对应重量标定的标志位
输入：无

输出：无

- Fun_3W_CAL

功能描述：解除锁定调用处理对应重量标定的标志位

输入：无

输出：标志位【ScaleFlag3,B_ScaleFlag3_3wCalOk】，1 为三点标定成功

- Fun_Delay_100MS

功能描述：延时 100MS

输入：无

输出：无

- Fun_Delay_40MS

功能描述：延时 40MS

输入：无

输出：无

- Fun_Delay_20MS

功能描述：延时 20MS

输入：无

输出：无

- Fun_Delay_10MS

功能描述：延时 10MS

输入：无

输出：无

- Fun_Delay_1MS

功能描述：延时 1MS

输入：无

输出：无

- Fun_power_Init

功能描述：设定芯片模拟电源部分

输入：无

输出：无

- Fun_power_Close

功能描述：关闭芯片模拟电源部分

输入：无

输出：无

- Fun_GPIO_Init
功能描述：初始化芯片 GPIO
输入：无
输出：无
- Fun_ADC_Init
功能描述：初始化芯片 ADC
输入：无
输出：无
- Fun_ADC_Close
功能描述：关闭芯片 ADC
输入：无
输出：无
- Fun_TIMER_init
功能描述：初始化芯片 TIMER
输入：无
输出：无
- Fun_TIMER_close
功能描述：关闭芯片 TIMER
输入：无
输出：无
- Fun_RAM_Zero
功能描述：清除连续多个字节的 RAM
输入：[BSR,IRP0],[WORK]
输出：无
- Table_Lcd_Num
功能描述：LCD 数字查表
输入：[WORK]
输出：LCD 数字段码
- Fun_LCD_Init
功能描述：初始化芯片 LCD
输入：无
输出：无
- Fun_LCD_Close
功能描述：关闭芯片 LCD
输入：无
输出：无

- Fun_LCD_Load
功能描述：加载显示缓冲区数据到 LCD 显示寄存器
输入：无
输出：无
- Fun_OTP_READ_CAL
功能描述：读取标定数据到 RAM 中
输入：无
输出：无
- Fun_OTP_WRITE_CAL
功能描述：写标定数据到 OTP ROM 中
输入：无
输出：无

第四章 应用层

4.1 GPIO

```

;--- io
AIENB_CFG_VALUE      EQU  00001000B
PT1EN_CFG_VALUE      EQU  00000000B
PT1PU_CFG_VALUE      EQU  11111111B
PT1_CFG_VALUE        EQU  11111111B

PT2EN_CFG_VALUE      EQU  00000000B
PT2PU_CFG_VALUE      EQU  11111111B
PT2_CFG_VALUE        EQU  11111111B

PT2MR_CFG_VALUE      EQU  00000000B
PT2CON_CFG_VALUE     EQU  00000000B

PTINT_CFG_VALUE      EQU  00000010B
AIENB2_CFG_VALUE     EQU  00000000B
  
```

SysConfig.inc 中宏定义

4.2 系统 TIMER

```

;--- TIMER
TMCON_USE            EQU  1
TMCON_CFG_VALUE      EQU  00001100B ; TMCLK/4 = 3906HZ/4 = 976.5HZ ,1/32S
  
```

SysConfig.inc 中宏定义

4.3 系统时钟

```

8 ;--- CLK
9 MCK_CFG_VALUE EQU 00000100B ; M3_CK=1,M2_CK=1,M1_CLK=0 [CPU 1000khz]
10 MCK2_CFG_VALUE EQU 00000010B
11 PCK_CFG_VALUE EQU 0

```

SysConfig.inc 中宏定义

4.4 看门狗

```

5 ;--- WDT
6 WDTCON_CFG_VALUE EQU 10001011B ; TIMEOUT 1S
7

```

SysConfig.inc 中宏定义

4.5 LCD 显示

```

;--- LCD
NETD_CFG_VALUE EQU 00000111B ; 2.8V , 10K REF
LCDENR_CFG_VALUE EQU 10100111B ; ,1/3 BIAS, 1/4 DUTY

```

SysConfig.inc 中宏定义

4.5.1 LCD 显示驱动原理

4.5.2 如何根据 LCD 逻辑图来定义段码位图

4.6 按键

```

;--- KEY,有效按键位定义,最多支持8个独立按键
KEY_USED_BITS EQU 00000001B
B_KEY_UNIT EQU 0

KEY_UNIT_PORT EQU PT2
KEY_UNIT_PIN EQU 0

KEY_LONG_TIME EQU 34 ; 33*30MS

```

SysConfig.inc 中宏定义

4.7 重量开机

```

;--- 重量开机参数设定(按点计算)
WEIGHT_ON_MODE           EQU 0
WEIGHT_ON_CONT_FAST      EQU 43 ;4.3kg/0.1kg=43
WEIGHT_ON_CONT_SLOW      EQU 47 ;4.7kg/0.1kg=47
WEIGHT_DOWN_CONT         EQU 21 ;2.1kg/0.1kg=21
WEIGHT_ON_FAST_UP_CNT    EQU 5
WEIGHT_ON_SLOW_UP_CNT    EQU 5
WEIGHT_DOWN_CNT          EQU 3
  
```

SysConfig.inc 中宏定义

4.8 标定

```

;--- 标定时，第一标定点通过的ADC变化量阈值
CALDOT1_ADC_THRESHOLD    EQU 500
  
```

SysConfig.inc 中宏定义

4.9 ADC

```

;--- ADc
ADCON_CFG_VALUE          EQU 00000111B ; 250K,/8192
NETA_CFG_VALUE           EQU 00000000B ; AI0,AI1->ADC
NETC_CFG_VALUE           EQU 00000110B ; 64X,AD_EN=1
TEMPC_CFG_VALUE          EQU 11100000B
  
```

```

;--- ADC 滤波系数
PRO_ADC_WINDOWDOWN      EQU 20
PRO_ADC_STABLE_RANG     EQU 5
PRO_ADC_STABLE_TIME     EQU 5
  
```

SysConfig.inc 中宏定义

4.10 重量点数

```

8 ;--- 50KG 对应的点数,50.0KG/0.1KG = 500
9 COUNT_DOT              EQU 500
0 START_COUNT            EQU 50
1 MEM_COUNT              EQU 3
2 MAX_COUNT              EQU 1531
3 UNLOCK_COUNT           EQU 100
4 TRACK_ZERO_CNT         EQU 3
  
```

SysConfig.inc 中宏定义

模板程序以 0.1KG 分度计算处理重量

4.11 默认单位


```
;--- 上电默认单位
        DEFAULT_UNIT      EQU    B_ScaleUnit_KG
        UNIT_MAX          EQU    08H
```

SysConfig.inc 中宏定义

上电默认 KG 单位

4.12 外部中断

模板中采用 PT20 的外部中断 0 程序，在开机时 PT20 做单位键使用，低功耗时 PT20 下降沿空唤醒开机，详情请见模板程序。

4.13 低电压检测

这部分代码由于灵活性较强，暂不放入库中

第五章 扩展与兼容性

用户可在此基础上进行其它功能的扩展，比如脂肪秤，蓝牙秤等