

一、任务要求

设计一个用于评价 0-7 岁儿童的营养发育评价仪，可以根据输入的儿童性别、年龄和体重信息判断出该儿童的生长发育是否正常。

【基本要求】：

1. 根据 WHO 标准建立相关的数据库
2. 被测儿童的身体特征信息可通过键盘输入
3. 输入参数和测试结果在 LCD 上显示
4. 附加功能：可根据父母身高预测该儿童的未来身高
5. 测试过程有声光提示
6. 有输入错误提示功能

二、功能特点与使用说明

功能介绍：

- a) 发育评价功能，可输入性别、年龄、身高、体重并从身高一年龄，体重一年龄，身高一体重，BMI-年龄四个标准对 7 岁以下的男儿童的生长发育情况进行评价。对于各评价标准的最严重的情况有黄灯警告。
- b) 对于儿童具有编号功能，可以记忆和复查指定编号儿童的数据和评价情况。
- c) 儿童身高预测功能，可根据儿童性别和父母身高预测其以后的最大最小身高值。
- d) 待机界面，在主功能选择界面停留 10s 时间可以进入待机界面，第一行显示“Welcome to use”，第二行显示当前时间（包含时分秒）。
- e) 可以修改当前时间（包含时分秒），并可以设置闹钟的开关和时间。
- f) 优良的输入体验，每次按键按下后很快绿灯和蜂鸣器有短促的响声提示按键成功，对于数据输入可以自由的退格，对于不可处理或非法的数据具有自动重新输入的功能，对于无效按键自动做无效处理。

使用说明：

- a) 复位后进入主功能选择界面，在主功能选择界面停留 10s 之后可以进入待机界面显示当前时间，按任意键回到主功能选择界面。
- b) 在可输入界面中，若为单键输入可直接按键进入；若为长数据输入（例如身高体重），F1 为确定，F2 为退格。
- c) 在主功能界面按 F1 可进入儿童发育评价系统，然后出现数据操作选择界面，按 F1 可以新建儿童数据，然后选择儿童编号进行覆盖，依次输入性别、年份、月份（小于 13 周还会提示输入周数）、身高、体重，确认后可依次查看身高一年龄，体重一年龄，身高一体重，BMI-年龄四个标准的评价，注意若有黄灯报警说明当前儿童的生长发育情况及其严重需要引起重视，按 F1 切换。在数据操作选择界面按 F2 后可以选择查看指定编号儿童的输入数据和评价数据，按 F1 切换。
- d) 在主功能界面按 F2 可以进入身高预测系统，依次输入性别，父母身高后按 F1 确定，第一行显示身高预测最大值，第二行显示身高预测最小值。
- e) 在主功能界面按 F3 可以修改当前系统时间，依次输入时分秒后 F1 确认。
- f) 在主功能界面按 F4 可以修改闹钟，0 和 1 分别是开启和关闭闹钟，如果选择开启闹钟可以设定闹钟的时分秒，F1 确认。

三、方案设计

（含设计思路、分析与计算、资源分配）

设计思路:

- a) 硬件模块: 芯片初始化和延时函数部分由 funcf020.h 完成, 为开发打下基础。键盘部分由 keyout.h 完成, 实现对单个键值的返回和对浮点数的返回。LCD 部分由 lcdcontrol.h 完成, 完成单个字符的显示, 指定位置的显示, 紧跟光标的显示功能。
- b) 生长发育评价数据处理和身高预测模块: 主要由 who.h 完成, 实现数据库 (L、M、S 值) 的存储和调用, 以及各评价指标 z 分数的计算, 还有身高的预测。
- c) 用户使用模块在主文件 (main.c) 实现, 包括输入提示, 非法数据检查, 工作状态切换, 逻辑检查高级功能实现。其中儿童生长发育评价功能子函数中, 在修改和查看数据时需要先选择儿童编号。
- d) 时钟/闹钟功能模块在主文件 (main.c) 实现, 其独立与其他功能, 时钟单独增加, 具有最高的优先级, 若处于待机态还具有显示当前时间的功能。
- e) 蜂鸣器和 LED1 闪烁由 funcf020.h 和 main.c 共同实现, 其开启由 lighting() 函数开启, 同时打开定时器, 在定时器中断子程序中关闭, 从而控制其闪烁时间。
- f) 定时器 0 用来定时扫描键盘列, 由 Keywait() 函数开启定时, 中断子程序扫描中若出现低电平, 则防抖检查后将键盘状态位清零并关闭定时器。
- g) 定时器 1 由 lighting() 函数开启 (同时打开 LED1 和蜂鸣器), 用于键盘输入提示和闹铃报警, 定时一段时间后将 LED1 和蜂鸣器关闭, 同时关闭定时器。
- h) 定时器 2 由 work_status_change() 函数开启, 用来定时待机等待时间, 达到时间后标志状态进入待机态, 同时在第一行显示欢迎词。任意按键可关闭定时。

分析与计算:

- a) z-score 计算公式:

$$z_{ind} = \frac{\left[\frac{y}{M(t)} \right]^{L(t)} - 1}{S(t)L(t)}$$
$$z_{ind}^* = \begin{cases} z_{ind} & \text{if } |z_{ind}| \leq 3 \\ 3 + \left(\frac{y - SD3pos}{SD23pos} \right) & \text{if } z_{ind} > 3 \\ -3 + \left(\frac{y - SD3neg}{SD23neg} \right) & \text{if } z_{ind} < -3 \end{cases}$$
$$SD3pos = M(t)[1 + L(t) * S(t) * (3)]^{\frac{1}{L(t)}}$$
$$SD3neg = M(t)[1 + L(t) * S(t) * (-3)]^{\frac{1}{L(t)}}$$
$$SD23pos = M(t)[1 + L(t) * S(t) * (3)]^{\frac{1}{L(t)}} - M(t)[1 + L(t) * S(t) * (2)]^{\frac{1}{L(t)}}$$
$$SD23neg = M(t)[1 + L(t) * S(t) * (-2)]^{\frac{1}{L(t)}} - M(t)[1 + L(t) * S(t) * (-3)]^{\frac{1}{L(t)}}$$

- b) 定时器计算 (晶振 16MHz):

- i. 定时器 0: 用于键盘扫描, 选择 12 分频, 每 ms 计数 1333.333333 次, 若需 150ms 左右扫描一次, 则需计数 200000 次, 那么每次计数 50000 次, 共四次计数算一次有效定时。65536 (0x10000) - 50000 (0xC350) = 15536 (0x3cb0);
- ii. 定时器 1: 用于 LED1 和蜂鸣器自动关闭程序, 选择 12 分频, 每 ms 计数 1333.333333

- 次，若短鸣需 200ms 左右关闭，则需计数 266667 次，那么每次计数 50000 次，计数 5 次，共计 183ms，已达到精度要求。 $65536(0x10000) - 50000(0xC350) = 15536(0x3cb0)$ ；
- iii. 定时器 2: 用于待机时间检查程序，选择 12 分频，每 ms 计数 1333.333333 次若短鸣需 10s 左右关闭，则需计数 266667 次，由于时间较长且精度要求相当低，直接使用最长的定时以减少不必要的计算和浪费，那么一次可以定时 $65536 * 12 / 16 \text{ us} = 49152 \text{ us}$ ， $10s / 49152us$ 约为 202 次；
- iv. 定时器 3: 用于时钟程序，需要较为精准的控制，选择 12 分频，每 ms 计数 1333.333333 次，需要延时 1s，共计计数 1333333 次，若要消除中断服务（包括判断和赋值）子程序程序和中断本身限制带来的影响，可设为 1333300 次，若每次计数 53332 次，共计 25 次可实现精确计数。考虑到晶振可能存在误差，在后期测试中，每分钟误差已小于 1s，满足最低要求。 $65536(0x10000) - 53332(0xD054) = 12204(0x2FAC)$

资源分配:

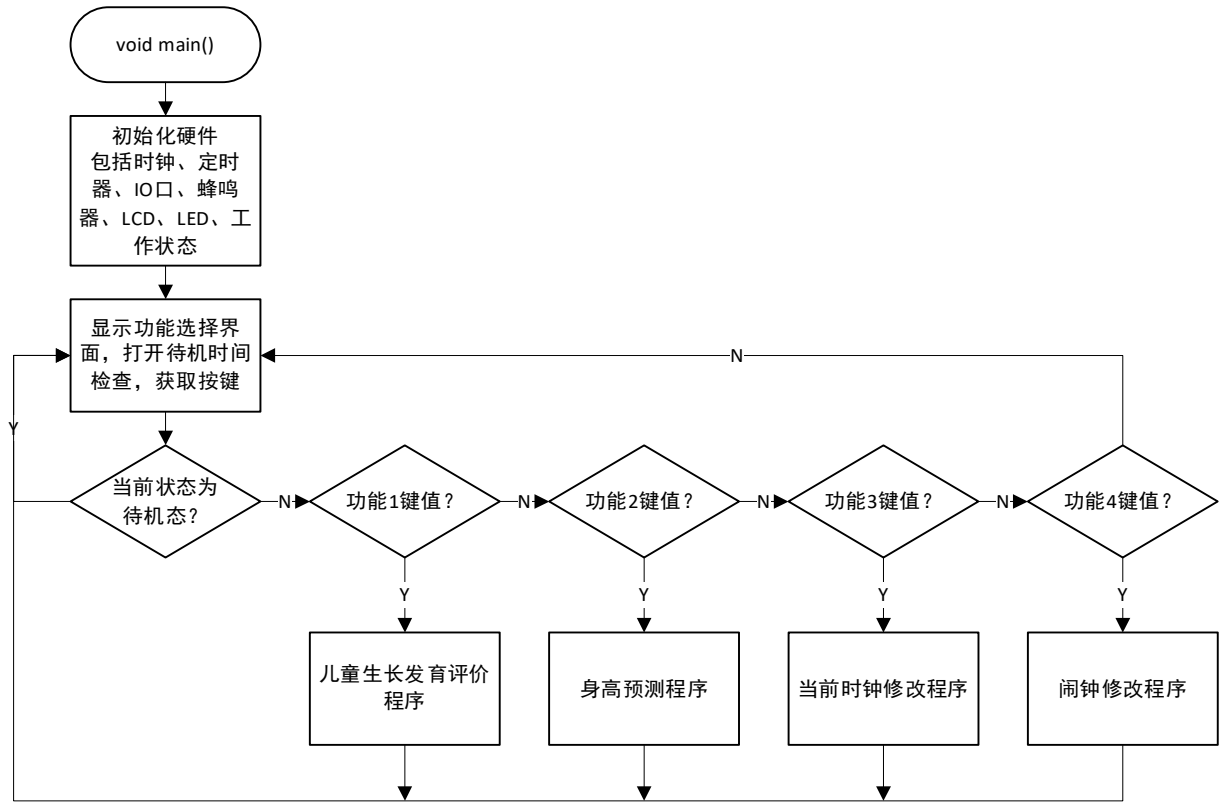
仅讨论主要数据。

- a) WHO 数据库，以 4 字节 float 浮点数存储于程序存储器中
- b) bit key_wait; //键盘中断标志位
bit work_status; //待机状态标志位
bit alarm; //闹钟开关标志位
uchar xdata lighting_time = 4; //存储蜂鸣器响应长度变量
四个状态标志位，以三个 1bit 数和一个 1 字节无符号整型数存储于内部数据存储器。
- c) uchar xdata store_sex[maxdata]; //用于存储数据的变量
uchar xdata store_month[maxdata];
uchar xdata store_year[maxdata];
uchar xdata store_week[maxdata];
float xdata store_height[maxdata];
float xdata store_weight[maxdata];
这些用于存储数据的变量由于仅需要简单写入和读取，存储与外部数据存储器，在以后的开发板中可存储于外部 EEPROM 或 FLASH 中用于断电保存
- d) uchar xdata month = 0, week = 0, sex = 0, year = 0; //用于计算的变量
float xdata height = 0, weight = 0;
这些用于将数据传入计算公式，只用于临时存取数据，故存在外部数据存储器
- e) uchar xdata hour = 0, minute = 0, second = 0; //存取当前时间
uchar xdata hour1 = 0, minute1 = 0, second1 = 0; //存取当前闹钟时间
存在外部数据存储器
- f) 函数
float Calcu_zcords(float L, float M, float S, float num) //根据 LMS 计算 Z 分数
中声明的三个变量:
float data Z_ind, SD3_pos, SD23_pos, SD3_neg, SD23_neg;
由于计算过程中设计大量 4 字节浮点数加减乘除乘方开方过程，故需要存在内部数据存储器以提高计算速度。
- g) void Lcd_Delay1ms(uchar c); //延时 1ms 子函数
void Delay10us(void); //延时 10us 子函数
void Delay10ms(void); //防抖动专用
三个函数中声明的变量，由于本身是用来延时功能，可存在外部数据存储器，提高运算时间，减少代码量。

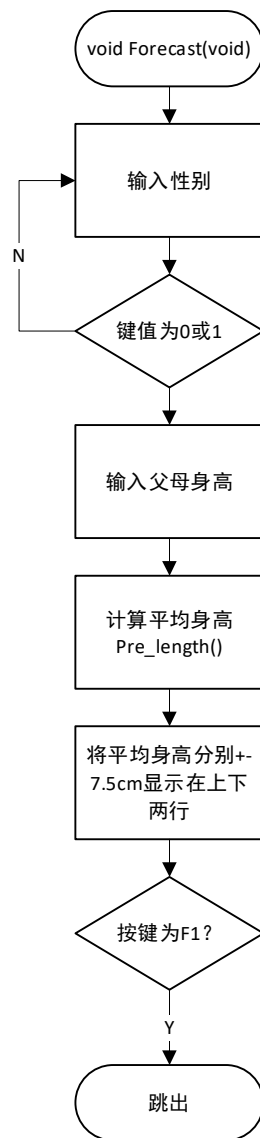
四、流程图（主程序、子程序的详细流程）

仅做较复杂程序流程图，图中忽略非法数据重新输入判断环节，仅注重逻辑关系

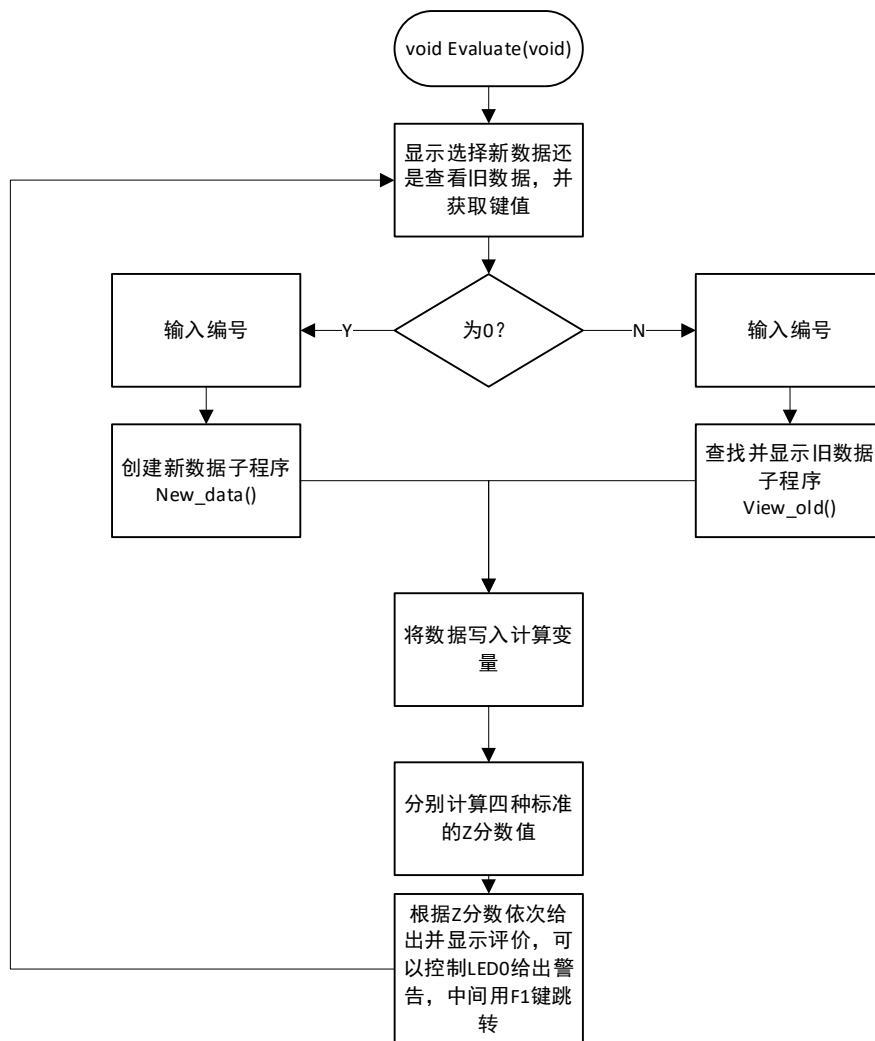
1. void main()



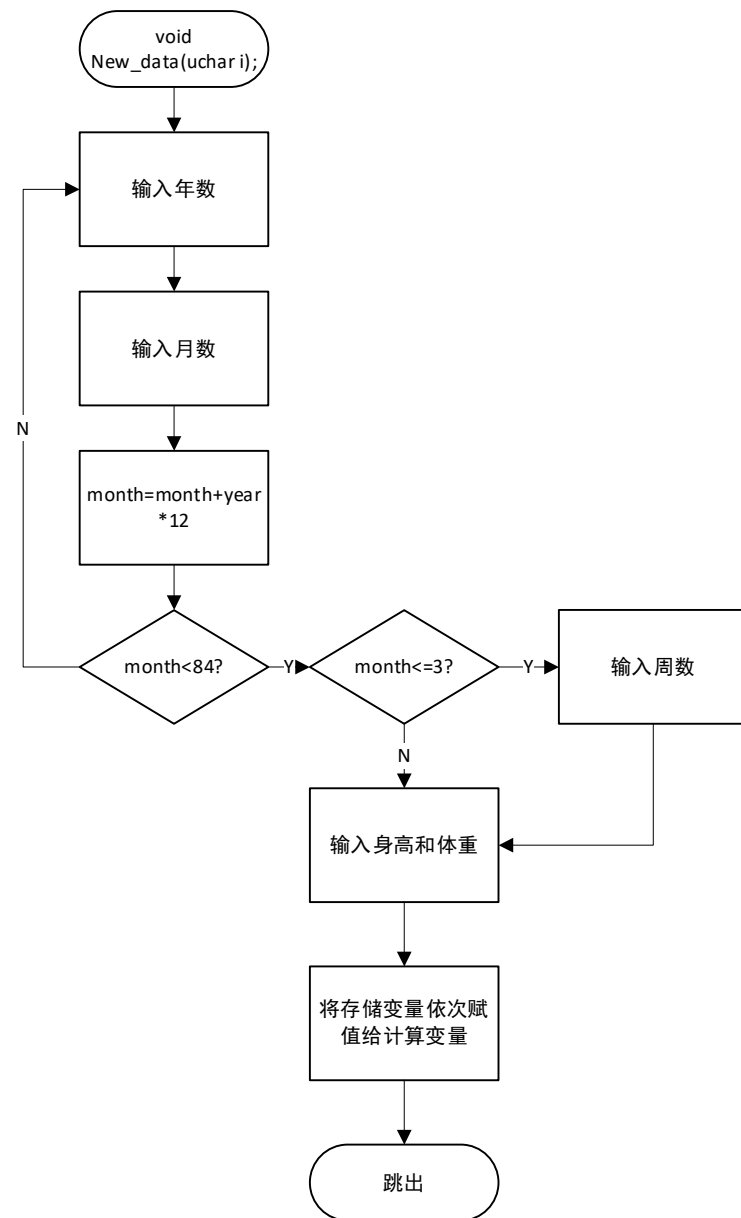
2. void Forecast(void) //身高预测子程序



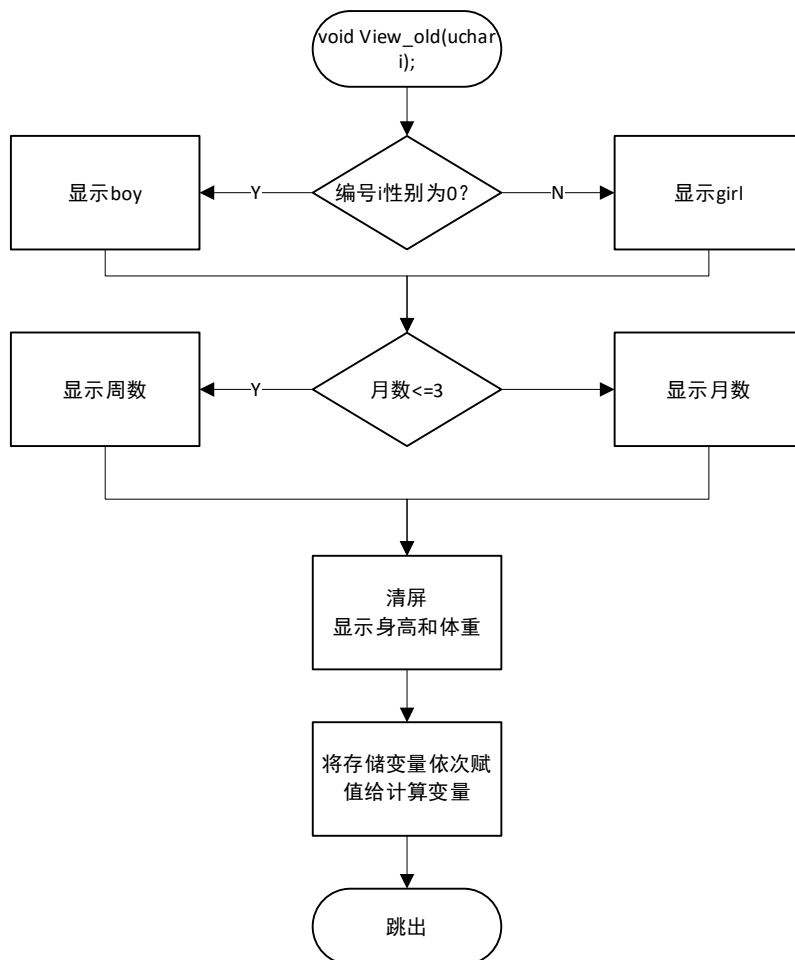
3. `void Evaluate(void)` //营养发育评价功能



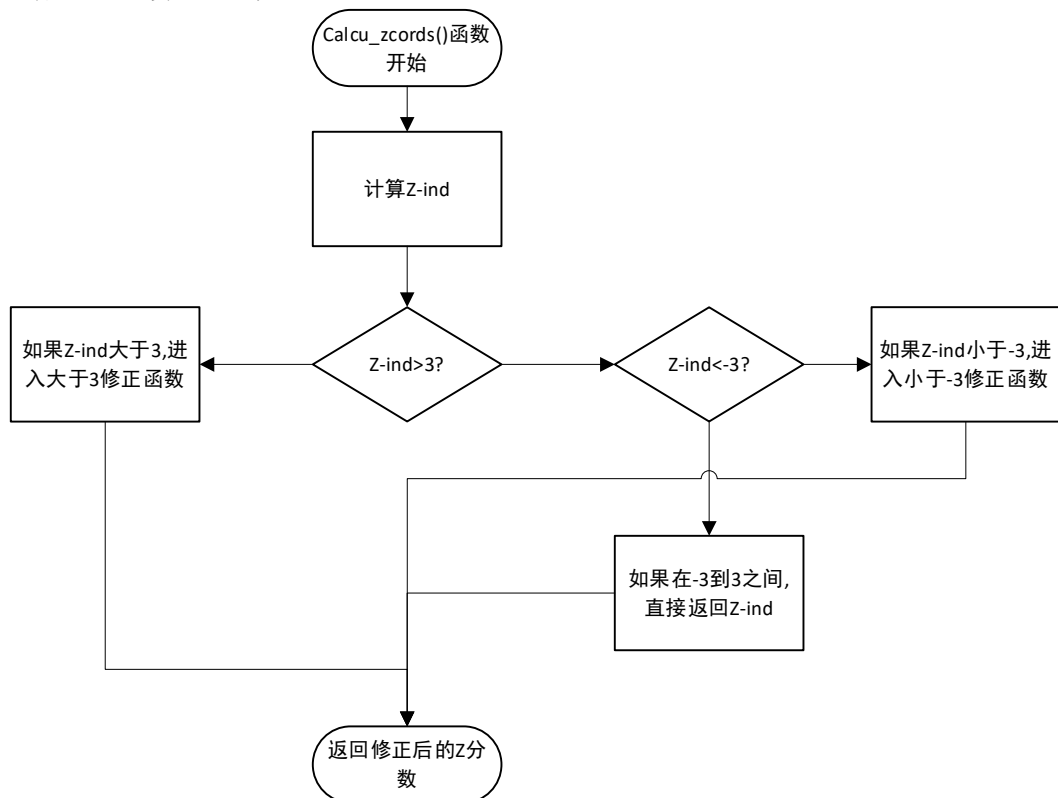
4. void New_data(uchar i); //新建数据子函数



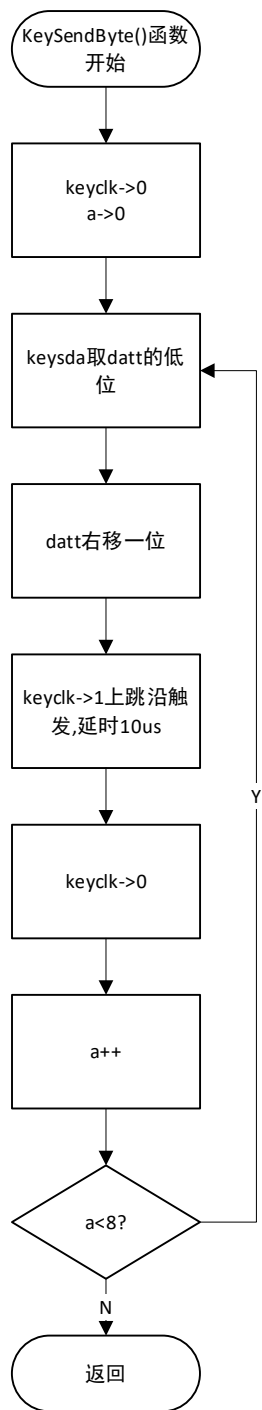
5. void View_old(uchar i); //查看旧数据子函数



6. float Calcu_zcords(float L, float M, float S, float num);
//根据 LMS 计算 Z 分数

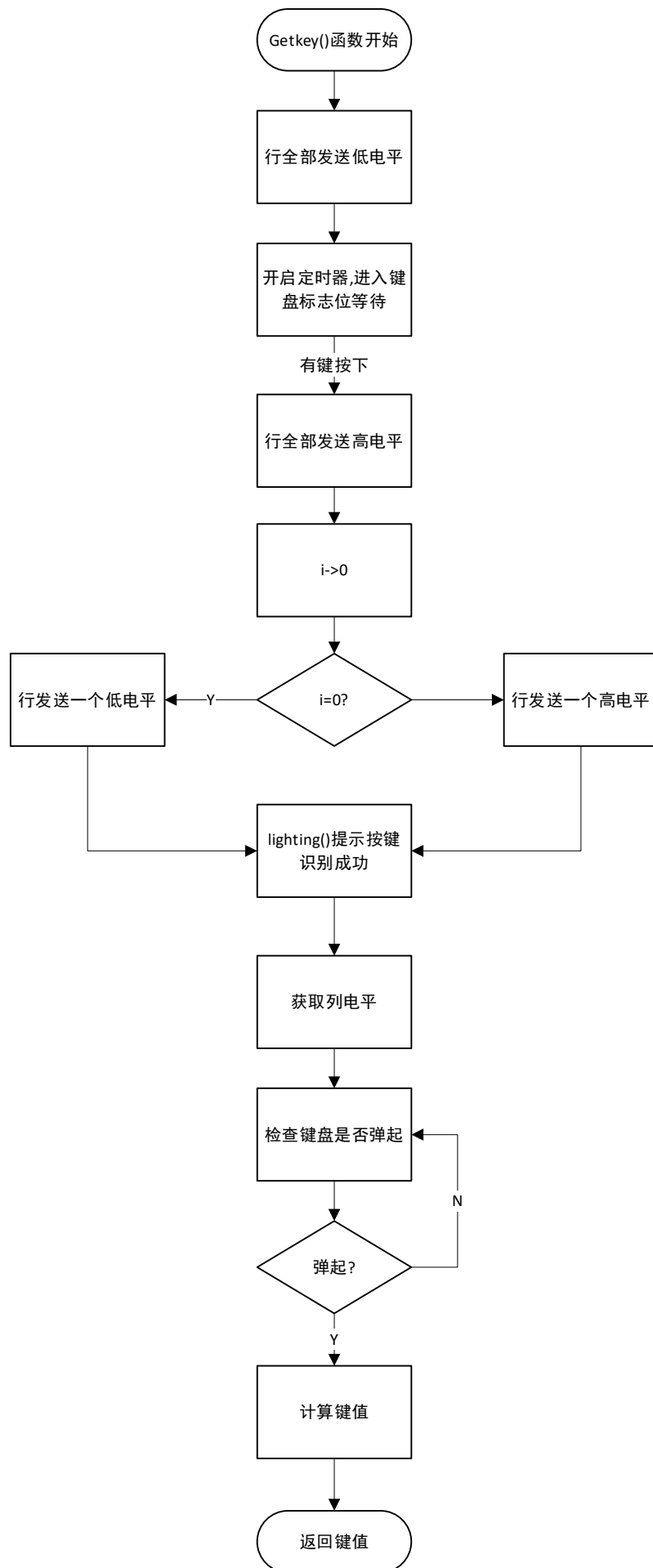


7. void KeySendByte(uchar datt); //向 74HC164 发送 1Byte 数据控制行扫描



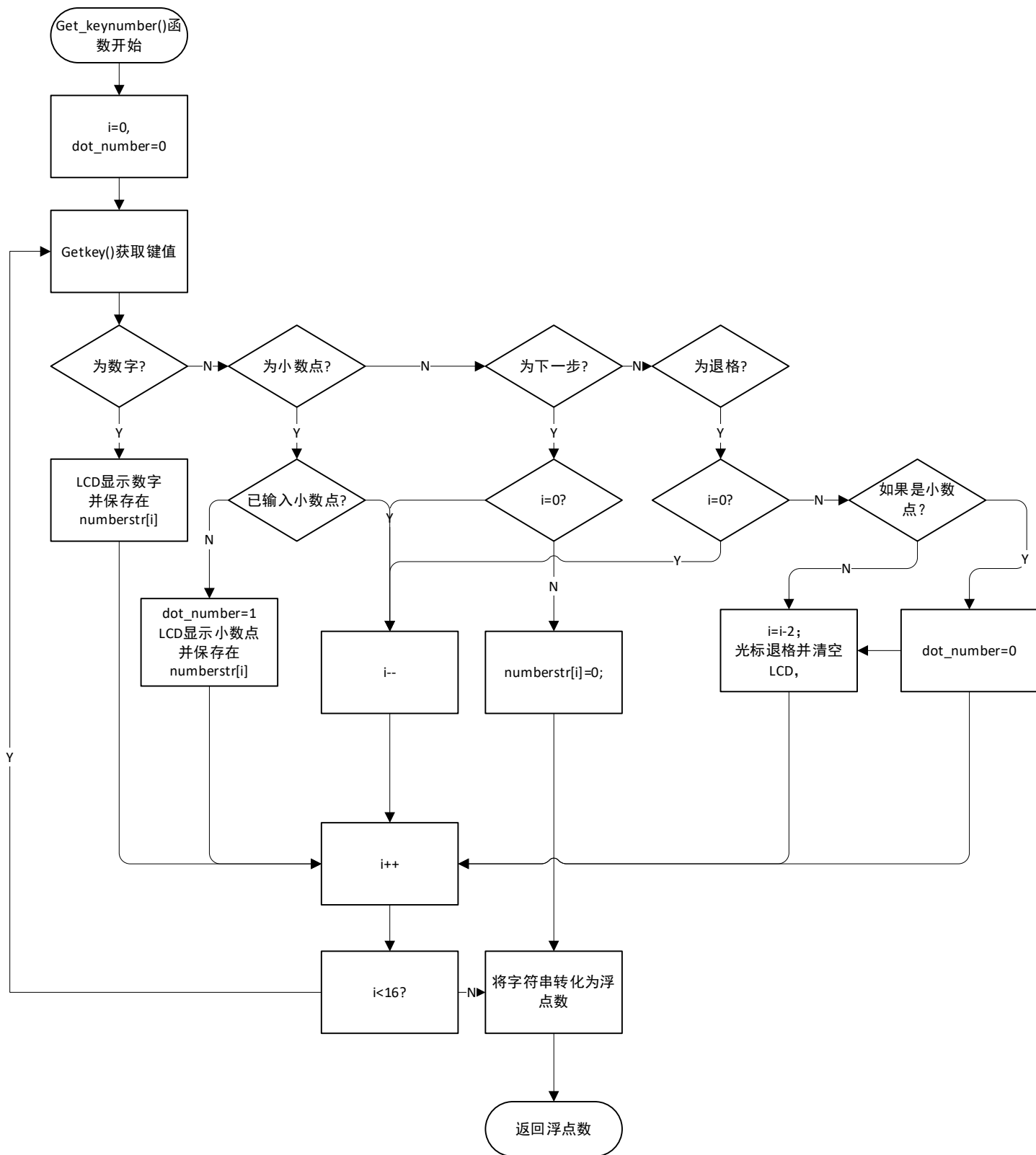
8. uchar Getkey();

//获得当前按键值

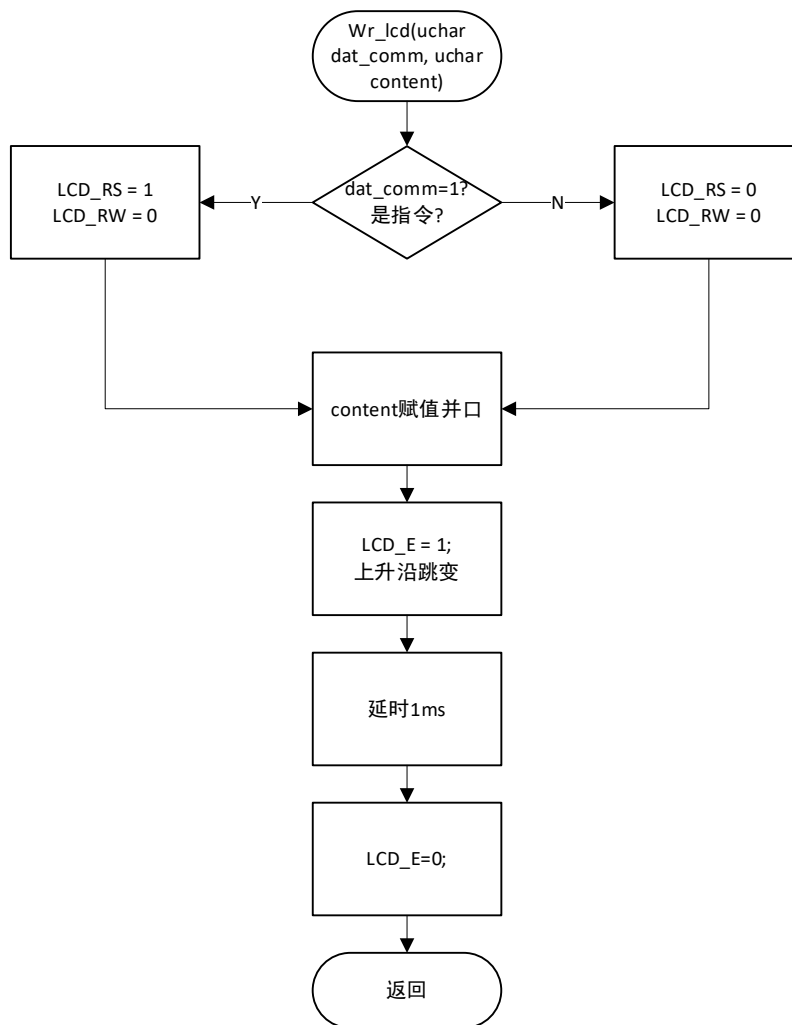


9. float Get_keynumber(void);

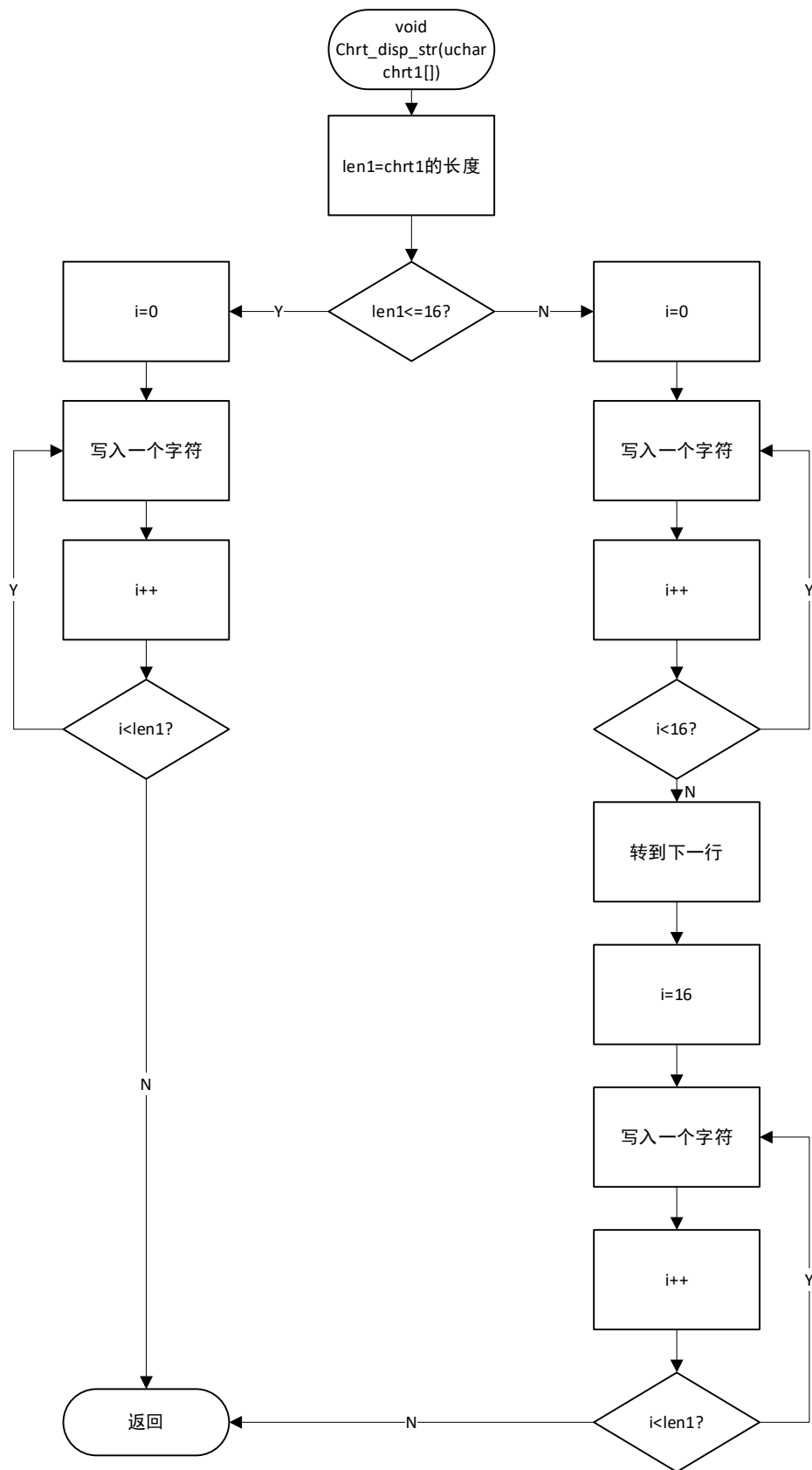
//获取当前输入的浮点数,并依次显示在 LCD 上



10. void Wr_lcd(uchar dat_comm, uchar content); //在 AC 位置写入一个字符



11. `void Chrt_disp_str(uchar chrt1[]);` //在 AC 位置写入字符串



五、源程序 （含文件头说明、资源使用说明、语句行注释）

略……

六、程序测试方法与结果

1. 复位后，按 F1 进入营养评价系统，再按 F1 新建数据，选择编号 1，依次输入 0（男），0（year），10（week），74（height），9.5（weight）（其中多次退格以验证退格程序），可

依次看到四个评价都为“median……”。然后重复操作，选择编号 2，输入一个比较极端的值，可以看到黄色 LED 对于危险评价可以报警。

2. 在主功能界面按 F1，再按 F2 查看旧数据，选择编号 1，可以查看之前输入的数据和评价，验证其正确性。
3. 在主功能界面按 F3 修改当前时间为 18:42:00，退回主功能界面，等待 10s 后进入待机界面，可以看到时间显示正常
4. 3. 在主功能界面按 F4 后，按 1 打开闹钟，修改当前时间为 18:43:00，退回主功能界面，等待 10s 后进入待机界面，当时间到达 18:43:00 时可以听到 2s 左右的报警声

七、心得与体会

自高中学习 VB 语言以来，我对编程有着浓厚的兴趣，但是由于缺乏系统的训练，所以没有什么进步。在大学学习了 C 语言之后，我也是深深的感受到了编程语言的迷人之处，所以便积极的报名了当时的“win32 窗口科学计算器”专题，得到了老师的表扬，并获得了大作业免做的优惠，这是我的第一次专题报告，前后共用了 3 天时间。

在我报这次专题之前，我原以为也可以 1 个星期写完代码，毕竟暑假考过计算机三级——嵌入式系统开发，然后还自购单片机开发板进行开发。然而在我仔细分析电路图之后才发现，事情没有我们想象的那么简单，首先键盘行扫描接入的是 74HC165 串行转并行，串行口未接在 UART 接口上，而且键盘不能用中断，这就是在考验我们对电路的控制能力。然后有关 LCD 的部分由于自购单片机上面有教程，所以我就对这些轻车熟路，大约用了一个半星期写完了硬件程序。

第二个困扰我的问题是 WHO 数据的建立和算法的实现，还好在我和唐永前充分浏览 WHO 网站之后发现 Z 分数的计算公式，于是我们的算法就有依次比较变成了直接计算，数据库文件变为了之前的 3/7，这对于我们算法的运行速度有了极大的提升。

第三个困扰我的问题便是差异化的问题，基础功能很简单大家也都差不多，但是这样会导致同质化严重。然而一开始我也想不到有什么功能，而且时间还不多。后来知道，验收当天上交板子，觉得机不可失不再来，然后就加入了儿童编号和复查的功能，本来是想实现断电存储的，后来发现不可以。后来在一个教室里，其他班上一个同学和另一个同学提建议加入时钟功能，另一个同学拒绝了，然而好战的我把它记在心里。终于在一个失眠的夜晚，构思了他的全部框架，第二天便实现了。

最后我认为，我的优势在于良好的写代码习惯和清晰的思路，写出来的代码除了输入有误之外没有出现逻辑上的错误，于是 DEBUG 只用了两个小时，后来加各种功能更是一次性成功，验收前在南七楼临时加入了小数点输入验证（即每次只能输入一个小数点，多输自动无效，退格后可重新输入）功能，也是一次成功。然后由于我是用 Visual Studio Code 进行代码编辑，一键全文排版、高亮显示、代码变量自动提示、函数定义一键查看等功能，极大地提高了我的效率，也提高了代码的可读性。

大考在即，尚未复习，写的匆忙，还请见谅。

设计人签名：_____ [手写签名]