



朗睿电子
工业交互显示专家

工业智能液晶显示器

A F

系
列
通
用
指
令
集

日期	修订内容

目 录

1. 基础设置和指令格式	
1.1 设置 CAN 波特率.....	3
1.2 设置 CAN ID.....	3
1.3 设置串口波特率.....	4
1.4 背光控制.....	4
1.5 控制蜂鸣器.....	4
1.6 获取系统时间.....	5
1.7 在指定位置显示图片.....	5
2. 基本图形操作	
2.1 画点.....	6
2.2 绘制直线.....	6
2.3 绘制椭圆.....	7
2.4 绘制圆.....	8
2.5 绘制矩形.....	8
2.6 绘制圆弧（扇形）.....	9
3. 图片操作	
3.1 调用局部位图.....	10
3.2 调用全屏位图.....	10
3.3 全屏位图剪切.....	11
3.4 用串口发送程序转发到指定的页面.....	11
4. 文字操作	
4.1 清屏.....	12
4.2 设置文字颜色.....	12
4.3 设置文字背景颜色.....	12
4.4 自动判断、显示字符.....	13
4.5 在任意位置显示字符串.....	13
4.6 字库类型选择.....	14
4.7 设置文字放大倍数.....	14
5. 控件操作	
5.1 设置变成灰色按钮的灰色度指数.....	15
5.2 指令控制按钮，改变编辑框的值.....	15
5.3 激发一个页面的事件.....	16
5.4 仪表旋转.....	16
5.5 设置柱状图的高度.....	17

1. 基础设置和指令格式

指令组成格式:

指令	1B	XX	XX... .. XX	XXXX
说明	帧头	功能码	指令参数	校验和

1.1 设置 CAN 波特率

单片机发送指令	1B 0E 02 00 04(B PS_N)00 (SUM_H)2F(SUM_L)
命令解析	1B : 帧头
	0E : 表示设置 C A N 信息
	02 : 指令数据两个字节
	00 : 设置波特率
	04: 设 置 C A N 波 特 率 为 5 00kbps
	00 : 校验和高字节
	2F : 校验和低字节
提示	BPS_N: C A N 波 特 率 设 置 值; BPS_N=00, 50kbps BPS_N=01, 100kbps BPS_N=02, 125kbps BPS_N=03, 250kbps BPS_N=04, 500kbps BPS_N=05, 1000kbps *注意事项: 液晶重启后生效
使用函数	void LR_LCD_CanBoudS et(UCHAR BPS_N)

1.2 设置 CAN ID

单片机发送指令	1B 0E 02 01 04(i d V a l u e) 00(SUM_H)30(SUM_L)
命令解析	1B : 帧头
	0E : 表示设置 C A N 信息
	02 : 指令数据两个字节
	01: 设 置 I D
	04: 设 置 C A N 的 I D 为 04
	00 : 校验和高字节
	30 : 校验和低字节
提示	idValue: C A N 的 I D 号 *注意事项: 液晶重启后生效
使用函数	void LR_LCD_CanIdSet (UCHAR idValue)

1.3 设置串口波特率

单片机发送指令	1B 1D 01 04(BPS_N) 00(SUM_H) 3D(SUM_L)
命令解析	1B : 帧头
	1D : 表示设置设置串口波特率
	01 : 指令数据一个字节
	04 : 设置波特率为 115200
	00 : 校验和高字节 3D : 校验和低字节
提示	B P S _ N : 串口波特率设置值; /B PS_N=00, 4800; /B PS_N=01, 9600; /B PS_N=02, 19200; /B PS_N=03, 38400; /B PS_N=04, 115200;
使用函数	void LR_LCD_Boud Set(UCHAR BPS_N)

1.4 背光控制

单片机发送指令	1B 12 01 01 00(SUM_H) 38(SUM_L)
命令解析	1B : 帧头
	12 : 软件背光控制
	01 : 指令数据 1 个字节
	01 : 亮度等级设置为 10
	00 : 校验和高字节
	38 : 校验和低字节; 2E: 背光关; 2F: 背光明
提示	Light_Grade: 0-10 共 10 个等级, 液晶能自动保存亮度参数等级 00: 关闭背光, 下次重新启动后按照 1 的亮度等级, 以防止背光不亮无法操作。
使用函数	void LR_LCD_BackLightC TL(UCHAR Light_Grade)

1.5 控制蜂鸣器

单片机发送指令	1B 6A 01 01 00(SUM_H) 87(SUM_L)
命令解析	1B : 帧头
	6A : 在指定的位置显示 CRC 校验的值
	01 : 指令数据 1 个字节
	01: 01 蜂鸣器开, 00 蜂鸣器关
	00 : 校验和高字节
	87 : 校验和低字节
提示	Beep: beep=1 蜂鸣器响 beep=0 结束一个事务
使用函数	void LR_LCD_ControlB eep(UCHAR beep)

1.6 获取系统时间

单片机发送指令	1B 22 00 00 (SUM_H) 3D (SUM_L)
命令解析	1B : 帧头
	22 : 读取液晶内部当前时间
	00 : 指令数据 0 个字节
	00 : 校验和高字节
	3D : 校验和低字节
液晶屏回复	1B 10 07 DB 09 06 02 0E 1B 2D
回复命令解析	1B : 帧头
	10 : 返回液晶内部当前时间
	07 : 年高字节 DB : 年低字节
	09: 月 06: 日 02 : 星期
	0E: 时 1B: 分 2D: 秒
使用函数	void LR_LCD_GetCurTime(void)

1.7 在指定位置显示图片

单片机发送指令	1B 2F 07 32 04 00 06 01 09 00 00 97
命令解析	1B : 帧头
	2F : 控件信息
	07 : 指令数据 7 个字节
	32 : LOCBMP_ITEM 局部位图控件:参见 lr_lcd_api.h
	01 : 控件所在的页面
	05 : 控件在该页面中的编号
	50 : 局部位图显示的时候左上角在屏上的 x 坐标低字节
	00 : 局部位图显示的时候左上角在屏上的 x 坐标高字节
	60 : 局部位图显示的时候左上角在屏上的 y 坐标低字节
	00 : 局部位图显示的时候左上角在屏上的 y 坐标高字节
	00 : 校验和高字节
97 : 校验和低字节	
提示	controlType: 使用宏定 LOCBMP_ITEM ItemIndex: 局部位图 ID Sx : 局部位图显示的时候左上角在屏上的 x 坐标 Sy : 局部位图显示的时候左上角在屏上的 y 坐标
使用函数	void LR_LCD_LoadPartBmpEx (UCHAR controlType, USHORT ItemIndex, USHORT sx, USHORT sy)

2. 基本图形操作

2.1 画点

单片机发送指令	1B 40 05 50 00 60 00 FF 02(SUM_H) 0F(SUM_L)
命令解析	1B : 帧头
	40 : 画点
	05 : 指令数据五个字节
	50 : X 坐标低字节
	00 : X 坐标高字节
	60 : Y 坐标低字节
	00 : Y 坐标高字节
	FF : 点的颜色
	02 : 校验和高字节
0F : 校验和低字节	
提示	Sx: X坐标 Sy: Y坐标 clr: 点的颜色 颜色查看 lr_lcd_api.h 文件下的 COLOR_256
使用函数	void LR_LCD_SetPixel (USHORT sx,USHOR T sy,UCHAR clr)

2.2 绘制直线

单片机发送指令	1B 46 09 FF 00 00 00 00 50 00 50 00 02(SUM_H) 09(SUM_L)
命令解析	1B : 帧头
	46 : 直线类型, 0x 46 为实线, 0x 5F 为虚线
	09 : 指令数据九个字节
	FF : 直线颜色为白色
	00 : X 起 点坐标低字节
	00 : X 起 点坐标高字节
	00 : Y 起 点坐标低字节
	00 : Y 起 点坐标高字节
	50 : X 终 点坐标低字节
	00 : X 终 点坐标高字节
	50 : Y 终 点坐标低字节
	00 : Y 终 点坐标高字节
02 : 校验和高字节 09 : 校验和低字节	
提示	style: 直线类型 style=LINE_SOLID 为实线, style=LINE_DOT 为虚线 sx : 起点 x 坐标 sy : 起点 y 坐标 ex: 终点 x 坐标 ey: 终点 y 坐标 clr: 颜色颜色查看 lr_lcd_api.h 文件的 COLOR_256
使用函数	Void LR_LCD_Line(UCHARstyle, USHORTsx, USHORT sy, USHORT ex, USHORT ey, UCHAR clr)

2.3 绘制椭圆

单片机发送指令	1B 4E 02 64 41 01 10 1B 13 07 FF 50 00 60 00 30 00 02 (SUM_H) 14 (SUM_L)
命令解析	1B : 帧头
	4E : 设置长短轴比值
	02 : 指令数据二个字节
	64 : 水平轴比
	41 : 垂直轴比
	01 : 校验和高字节
	10 : 校验和低字节
	1B : 帧头
	13 : 圆的类型 ; 13 实心圆 , 59 空心圆
	07 : 指令数据七个字节
	FF : 圆的颜色
	50 : X 终点坐标低字节
	00 : X 终点坐标高字节
	60 : Y 终点坐标低字节
	00 : Y 终点坐标高字节
	30 : 垂直轴低字节
00 : 垂直轴高字节	
02 : 校验和高字节	
14 : 校验和低字节	
提示	<p>x : 圆心 x 坐标 y : 圆心 y 坐标 w : 水平半轴 h : 竖直半轴 clr: 颜色 style:style=SOLID_CIRCLE 画实心椭圆 style=HOLLO_CIRCLE 空心椭圆 轴比计算: if (w>=h) {chW=0x64; chH=0x64*h/w; } if (w<h) { chH=0x64; chW=0x64*w/h; } clr: 颜色 颜色查看 lr_lcd_api.h 文件下的 COLOR_256</p>
使用函数	void LR_LCD_Ellipse(USHORT x, USHORT y, USHORT w , USHORT h , UCHAR clr, UCHAR style)

2.4 绘制圆

单片机发送指令	1B 13 07 FF 50 00 60 00 30 00 02(SUM_H) 14(SUM_L)
命令解析	1B : 帧头
	13 : 圆的类型 ; 13 实心圆 , 59 空心圆
	07 : 指令数据七个字节
	FF : 圆的颜色
	50 : X 终点坐标低字节
	00 : X 终点坐标高字节
	60 : Y 终点坐标低字节
	00 : Y 终点坐标高字节
	30 : 半径低字节
	00 : 半径高字节
	02 : 校验和高字节
	14 : 校验和低字节
提示	<p>style:style=SOLID_CIRCLE 画实心圆 style=HOLLO_CIRCLE 空心圆 x : 圆心横坐标 y : 圆心纵坐标 r: 半径 clr: 颜色 颜色查看 lr_lcd_api.h 文件下的 COLOR_256</p>
使用函数	void LR_LCD_Circle(USHORT x ,USHORT y,USHORT r ,UCHAR clr,UCHAR style)

2.5 绘制矩形

单片机发送指令	1B 41 09 FF 00 00 00 00 50 00 6000 02(SUM_H) 14(SUM_L)
命令解析	1B : 帧头
	41:RECT_SOLID(41)实心 RECT_HOLLO(5A)空心 宏定义参见 lr_lcd_api.h
	09 : 指令数据 9 个字节
	FF : 矩形框颜色 LOCBMP_ITEM 局部位图控件
	00 : 矩形左上角的 X 坐标低字节
	00 : 矩形左上角的 X 坐标高字节
	00 : 矩形左上角的 Y 坐标低字节
	00 : 矩形左上角的 Y 坐标高字节
	50 : 矩形右下角的 X 坐标低字节
	00 : 矩形右下角的 X 坐标高字节
	60 : 矩形右下角的 Y 坐标低字节
	00 : 矩形右下角的 Y 坐标高字节
	02 : 校验和高字节
	14 : 校验和低字节

提示	Clr: 颜色 S x : 左上角 x 坐标 S y : 左上角 y 坐标 E x : 右下角 x 坐标 E y : 右下角 y 坐标 Style: style=RECT_SOLID 实心 style=RECT_HOLLO 空心 颜色以及宏定义参见 lr_lcd_api.h
使用函数	Void LR_LCD_Rectangle (USHORT sx, USHORT sy, USHORT ex, USHORT ey, UCHAR clr, UCHAR style)

2.6 绘制圆弧（扇形）

单片机发送指令	1B 16 0B FF 50 00 60 00 10 00 20 00 30 00 02 4B
命令解析	1B : 帧头
	16 : 表示画弧, 1 7 表示画扇形
	0B : 指令数据 1 1 个 字节
	FF: 颜色
	50 : 圆心 X 坐标低字节
	00 : 圆心 X 坐标高字节
	60 : 圆心 Y 坐标低字节
	00 : 圆心 Y 坐标高字节
	10 : 半径低字节
	00 : 半径高字节
	20 : 起始角度低字节
	00 : 起始角度高字节
	30 : 终止角度低字节
	00 : 终止角度高字节
02 : 校验和高字节	
4B : 校验和低字节	
提示	X : 圆心横坐标 Y : 圆心纵坐标 R : 半径 Stangle: 起始角 Endangle: 终止角 Clr : 颜色 Style: style=STYLE_PI 画扇形 , style=STYLE_ARC 圆弧
使用函数	void LR_LCD_Arc (USHORT x , USHORT y , USHORT r, USHORT stangle, UCHAR endangle, UCHAR color, UCHAR style)

3. 图片操作

3.1 调用局部位图

单片机发送指令	1B29 0A 10 05 50 00 60 00 30 0040 00 01(SUM_ H)82(SUM_L)
命令解析	1B : 帧头
	29 : 调用局部位图
	0A : 指令数据 10 个字节
	10 : 局部位图所在的块
	05 : 局部位图所在的页
	50 : 局部位图显示的时候左上角在屏上的 x 坐标低字节
	00 : 局部位图显示的时候左上角在屏上的 x 坐标高字节
	60 : 局部位图显示的时候左上角在屏上的 Y 坐标低字节
	00 : 局部位图显示的时候左上角在屏上的 Y 坐标高字节
	30 : 局部位图的宽度低字节
	00 : 局部位图的宽度高字节
	40 : 局部位图的高度低字节
	00 : 局部位图的高度高字节
01 : 校验和高字节	
82 : 校验和低字节	
提示	Block: 局部位图所在的块 Page: 局部位图的起始页 S x : 局部位图显示的时候左上角在屏上的 x 坐标 S y : 局部位图显示的时候左上角在屏上的 y 坐标 Height: 局部位图的高度 Width: 局部位图的宽度
使用函数	void LR_LCD_LoadPartBmp(UCHAR block,UCHAR page, USHORT sx,USHORT sy,USHORT width, USHORT height)

注：前提是局部位图已下载到 flash 中，在画面工程中---->工程设置----->局部位图，找到对应的存储位置，图片实际宽高

3.2 调用全屏位图

单片机发送指令	1B 19 01 05 00(S U M_H)3A(SUM_L)
命令解析	1B : 帧头
	19 : 调用全屏位图
	01 : 指令数据 1 个字节
	05 : 要调用的全屏位图所在的块
	00 : 校验和高字节
	3A : 校验和低字节
提示	N : 要调用的全屏位图所在的块
使用函数	void LR_LCD_Load FullBmp(UCHAR n)

3.3 全屏位图剪切

单片机发送指令	1B1A 0D 05 00 00 00 00 00 50 0060 00 30 00 3001 57
命令解析	1B : 帧头
	1A : 全屏位图剪切
	0D : 指令数据 1 3 个 字节
	05 : 选用第 5 幅全屏位图
	00 : 要截取背景位图部分左上角 X 坐标高字节
	00 : 要截取背景位图部分左上角 X 坐标低字节
	00 : 要截取背景位图部分左上角 Y 坐标高字节
	00 : 要截取背景位图部分左上角 Y 坐标低字节
	00 : 要截取背景位图部分右下角 X 坐标高字节
	50 : 要截取背景位图部分右下角 X 坐标低字节
	00 : 要截取背景位图部分右下角 Y 坐标高字节
	60 : 要截取背景位图部分右下角 Y 坐标低字节
	00 : 截取背景位图部分在屏上显示的 X 坐标高字节
	30 : 截取背景位图部分在屏上显示的 X 坐标低字节
	00 : 截取背景位图部分在屏上显示的 Y 坐标高字节
30 : 截取背景位图部分在屏上显示的 Y 坐标低字节	
01 : 校验和高字节	
57 : 校验和低字节	
提示	Page_N: 全屏位图存放的位置, 在朗睿上位机软件中定; sx , sy, ex, e y : 要截取背景位图 部分左上角坐标和右下角坐标; X , Y : 截取背景位图部分在屏上显示的位置
使用函数	void LR_LCD_FullBmpCut (UCHAR Page_N, USHORT sx, USHORT sy, USHORT ex, USHORT ey, USHORT X, USHORT Y)

注: Page_N:代表全屏位图存储的起始地址, 新版画面编辑软件, 在工程设置--->预置画面----->页面顺序“1_0.fullbmp.pgs”, “0”代表存储起始位置

3.4 用串口发送程序转发到指定的页面

单片机发送指令	1B 2F 03 64 65 01 01(S U M_H) 17(SUM_L)
命令解析	1B : 帧头
	2F : 向控件对象发送数据
	03 : 指令数据 3 个字节
	64 : 控件类型为页面 PAGE_ITEM
	65 : 控制功能为初始化页面 INIT_PAGE_INDEX
	01 : 指定的页面编号
	01 : 校验和高字节 17 : 校验和低字节
提示	cPageIndex : 页面索引
使用函数	void LR_LCD_ForwardPage(unsigned char cPageIndex)

4. 文字操作

4.1 清屏

单片机发送指令	1B 42 01 03 00(SUM_H) 61(SUM_L)
命令解析	1B : 帧头
	42 : 清屏
	01 : 指令数据 1 个字节
	03 : 清屏颜色为蓝色
	00 : 校验和高字节
61 : 校验和低字节	
提示	Clr : 清屏颜色 颜色定义参见 lr_lcd_api.h
使用函数	void LR_LCD_ScreenClear(UCHAR clr)

4.2 设置文字颜色

单片机发送指令	1B 43 01 03 00(S U M_H) 62(SUM_L)
命令解析	1B : 帧头
	43 : 设置文字颜色
	01 : 指令数据 1 个字节
	03 : 文字颜色为蓝色
	00 : 校验和高字节
62 : 校验和低字节	
提示	Clr : 文字颜色 颜色定义参见 lr_lcd_api.h
使用函数	void LR_LCD_SetF_Clr(UCHAR clr)

4.3 设置文字背景颜色

单片机发送指令	1B 49 01 03 00(SUM_H) 68(SUM_L)
命令解析	1B : 帧头
	49 : 设置文字背景颜色
	01 : 指令数据 1 个字节
	03 : 设置文字背景颜色为蓝色
	00 : 校验和高字节
68 : 校验和低字节	
提示	Clr : 文字背景颜色 (覆盖模式下有效)颜色定义参见 lr_lcd_api.h
使用函数	void LR_LCD_SetF_Clr(UCHAR clr)

4.4 自动判断、显示字符

单片机发送指令	1B 25 04 31 32 33 34 01(SUM_H) 0E(SUM_L)
命令解析	1B : 帧头
	25 : 自动判断、显示字符
	04 : 指令数据 4 个字节
	31 : 显示数字 1
	32 : 显示数字 2
	33 : 显示数字 3
	34 : 显示数字 4
	01 : 校验和高字节
0E : 校验和低字节	
提示	Buf : 存放要显示的字符缓冲区, 要以 0 结尾, 特别注意, b u f 的最后要 有一个 0 , 就是说多添加一个 0
使用函数	void LR_LCD_ShowString(UCHAR *buf)

4.5 在任意位置显示字符串

单片机发送指令	1B 48 04 50 00 60 00 01(SUM_H) 17 (SUM_L) 1B 25 04 31 32 33 34 01(SUM_H) 0E (SUM_L)
命令解析	1B : 帧头
	48 : 光标以像素为单位直接定位
	04 : 指令数据四个字节
	50 : X 坐标低字节
	00 : X 坐标高字节
	60 : Y 坐标低字节
	00 : Y 坐标高字节
	01 : 校验和高字节
	17 : 校验和低字节
	1B : 帧头
	25 : 自动判断、显示字符
	04 : 指令数据 4 个字节
	31 : 显示数字 1
	32 : 显示数字 2
	33 : 显示数字 3
	34 : 显示数字 4
	01 : 校验和高字节
	0E : 校验和低字节

提示	X : 要显示的位置的 x 坐标 Y : 要显示的位置的 y 坐标 Buf : 存放要显示的字符缓冲区, 要以 0 结尾, 特别注意, buf 的最后要有一个 0 , 就是说多添加一个 0
使用函数	void LR_LCD_ShowStringAt(USHORT x, USHORT y, UCHAR *buf)

4.6 字库类型选择

单片机发送指令	1B 6201 01 00 (SUM_H) 7F (SUM_L)
命令解析	1B : 帧头
	62 : 字库类型选择
	01 : 指令数据 1 个字节
	01: 24x24 字库 data 取值 00 01 02
	00 : 校验和高字节
	7F : 校验和低字节
提示	data 取值 00 01 02 分别对应 16x16 24x24 32x32 字库 , 默认字库 16*16
使用函数	void LR_LCD_font_select(UCHAR cFONT)

4.7 设置文字放大倍数

单片机发送指令	1B 4D 01 51 00 (SUM_H) BA (SUM_L)
命令解析	1B : 帧头
	4D : 设置文字放大倍数
	01 : 指令数据 1 个字节 放大倍数 cy=1, cx=1; 放大倍数 cy=1, cx=2; 放大倍数 cy=2, cx=1; 放大倍数 cy=2, cx=2; 除此之外 (03) : 指令数据 3 个字节
	51 : 放大倍数 cy = 1 , cx = 1 ; 52 : 放大倍数 cy = 1 , cx = 2 ; 53 : 放大倍数 cy = 2 , cx = 1 ; 54 : 放大倍数 cy = 2 , cx = 2 ; (55) : 除此之外的放大倍数
	00 : 校验和高字节 (03 : CX 放大倍数)
	BA : 校验和低字节 (03 : CY 放大倍数)
	(00) : 校验和高字节 (C6) : 校验和低字节
提示	Cx : 宽度 Cy : 高度
使用函数	void LR_LCD_FontTimes(UCHAR cx, UCHAR cy)

5. 控件操作

5.1 设置变成灰色按钮的灰度度指数

单片机发送指令	1B 2F 04 01 06 02 03 00(SUM_H) 54 (SUM_L)
命令解析	1B : 帧头
	2F : 设置控件命令
	04 : 指令数据 4 个字节
	01 : 设置的控件类型为按钮 BUTTON_ITEM
	06 : 设置按钮的灰度值
	02 : 灰度值分子 mulFactor
	03 : 灰度值分母 divFactor
	00 : 校验和高字节
	54 : 校验和低字节
提示	灰度指数设置的值为(mulFactor/divFactor) 备注: 使用函数 LR_LCD_SetPropsToControlEx(BUTTON_ITEM, ID_BTN_741, disableOrEnd, 0); 设置 ID=ID_BTN_741 的为禁用按钮后, 禁用的按钮变为灰色, 灰度度可用此函数来设置。
使用函数	Void LR_LCD_SetDisBtnFactor(unsigned char mulFactor, unsigned char divFactor)

5.2 指令控制按钮, 改变编辑框的值

单片机发送指令	1b 2f 06 02 03 00 00 32 36 00 BD
命令解析	1B : 帧头
	2F : 设置控件命令
	0F : 指令数据 6 个字节
	02: 设置的控件类型为文本框 EDITE_ITEM
	03: 发送的是数据值 DATA_TRAN
	00: 控件所在的页面编号 (ID 值对应的高字节)
	00: 在该页面内控件的顺序编号 (ID 值对应的低字节)
	32 36: 对应字符串是" 26", 文本框内显示 26
	00 : 校验和高字节
	BD : 校验和低字节
提示	pageIndex: 按钮所在页面 ctrIndex: 按钮在页面中的编号
使用函数	void LR_LCD_SetValueToEditeEx(unsigned short keyIndex, unsigned char *buf)

5.3 激发一个页面的事件

单片机发送指令	1B 2F 03 01 0A 05 00(S U M_H) 5D(SUM_L)
命令解析	1B : 帧头
	2F : 向控件对象发送数据
	03 : 指令数据 3 个字节
	01 : 控件类型为按钮 BUTTON_ITEM (controlType)
	0A : 控制功能为初始化按钮组 INIT_GROUP_CMD(eventType)
	05 : 按钮组编号 (p Index)
	00 : 校验和高字节
	5D : 校验和低字节
提示	controlType: 控件类型 eventType: 事件类型 pIndex: 页面索引
使用函数	Void LR_LCD_ExecutePageEvent(unsigned char controlType, unsigned char eventType, unsigned char pIndex)

5.4 仪表旋转

单片机发送指令	1B 2F 07 0A 0B 00 00 01 00 00 00 67
命令解析	1B : 帧头
	2F : 向控件对象发送数据
	07:指令数据 7 个字节
	0A: 控件类型为仪表 DIAL_ITEM(controlType)
	0B: 设置指针的旋转角度
	00: 控件所在的页面编号 (ID 值对应的高字节)
	00: 在该页面内控件的顺序编号 (ID 值对应的低字节)
	01: 旋转角度值 (高字节)
	00: 旋转角度值 (低字节)
	00: 仪表上面动态显示的文字长度 (0 表示没有)
	67 : 校验和低字节
提示	pIndex 页面编号 (ID 值对应的高字节) controlIndex 页面内控件的顺序编号 (ID 值对应的低字节) ang 两字节的旋转角度 *buf 仪表上面显示的动态文字
使用函数	Void LR_LCD_SetAutoValueToDial(unsigned char pIndex, unsigned char controlIndex, unsigned short ang, unsigned char *buf)

5.5 设置柱状图的高度

单片机发送指令	1B 2F 07 14 15 01 05 01 00 50 00(SUM_H) D1(SUM_L)
命令解析	1B : 帧头
	2F : 控制控件
	07 : 指令数据 7 个字节
	14 : 柱状图控件
	15 : 设置柱状图高度
	01 : 页面编号
	05 : 控件在页面中的编号
	01 : 子柱状图的编号
	00 : 柱状图高度数据高字节
	50 : 柱状图高度数据低字节
	00 : 校验和高字节
	D1 : 校验和低字节
提示	pIndex: 页面索引 controlIndex: 柱状图的索引 childIndex : 子柱状图的索引 barHeight: 子柱状图的高度
使用函数	Void LR_LCD_SetChildBarHeight(unsigned char pIndex, unsigned char controlIndex, unsigned char childIndex, unsigned short barHeight)

注：仍有未使用到的指令，在今后测试和开发过程中及时增加。