5.1 声道 DTS/杜比数码 AC-3 音频解码 KC32C 用户手册

◆ 特性

- 采用多核双精度浮点数（64 位）DSP 计算处理器，所有音频处理都是双精度 64 位。
- 支持最高 192K 杜比数码 AC-3、AC-3 HD、DTS、DTS HD 等 5.1（I^2S 数字输出 7.1）声道解码。
- 三路 SPDIF 数码输入,可根据需要设计为光纤或同轴输入（同轴输入需要放大线路）。
- 6 声道 DAC 输出及双声道立体声输入，可选带 I^2S 数字 PCM 输出接口，用户可自行选配 DAC 及 ADC 以达到更理想的模式音频输出。
- 支持 U 盘及 SD/TF 卡多媒体文件播放或升级更新文件。
- 两路话筒 MIC 输入，支持最大 500 毫秒的多路延时混响及高低音或多段 EQ 均衡器，只需要增加一个运放就能实现话筒的全部功能。
- I2S 数字 PCM 输出时支持 7.1 声道及 MIC 与模拟输入交换功能，方便不使用 MIC 时代替模拟输入，这样可以仅增加外置的 DAC 芯片就可以获取更好的音质。
- 多达 16 段 EQ 频率均衡器，可以自行分配到各个声道及 MIC 输入之中，无需要外置任何音调及 EQ 电路即可以调整音色。
- 内置 2048 点 FFT 频谱取样 AI 算法，可以为主机输出最大 256x64 像素点的频谱或频率直接输出，无需要外置任何电路即可以驱动大点阵屏作声音动态显示。
- 内置 LIN SYNC 齿音同步功能，可以调节所有声道延迟最大 200 毫秒。
- 主声道 HPF 高通滤波器及超低音 LPF 低通滤波器频率可以任意调节，更容易匹配不同的低音炮。
- 兼容 Cirrus Logic 完整的低音管理结构，支持全部杜比标准低音配置及各种大小喇叭组合。
- 内置 AI 算法进行模拟输入静音，当模拟没有信号时可自动进行静音，免除了外接检测电路。
- 所有声道都可以进行+/-10dB 的声道微调。
- 内置常用的音量芯片控制程序，可以选择及定制音量 IC 及音量步数。
- 通讯接口直接升级固件或通过 UsI2c 硬件下载固件，可以配合用户主机现有的系统实现云升级功能，可以在线直接升级控制程序，极大方便调试及生产维护。
- 使用 I^2C 从机接口，用户主机无需增加额外接口即可使用现成的 I^2C 接口。
- I^2C 接口可以与其他 I^2C 设备并联使用。KC32C 与 24C01 等 I^2C 设备完全相同，非常容易进行二次开发。
- I^2C 通讯带有 INT 中断输出端口，用户主机可以在 INT 变化时才读取相应的数据，减少了用户主机的通讯占用时间。
- 全部寄存器带有掉电记忆，用户主机写入的数值都可以读取寄存器后还原，用户不需要使用记忆芯片，所有记忆位置使用 AI 算法，可以无限次数重复使用，随时随地写入就可以了。
- 提供独立的 64 字节的记忆体空间，与 24C01 的功能完全相同，用户可以省略例如 24C01、93C46 等记忆芯片。
- 数码与模拟地线独立，降低对主板 PCB LAYOUT 的要求，获得更好的性能。
- 直接板上安装，可与音频板组成一体化产品，改善传统解码板的连线，提高了可靠性及增加了产品的可观性。

◆ 应用范围

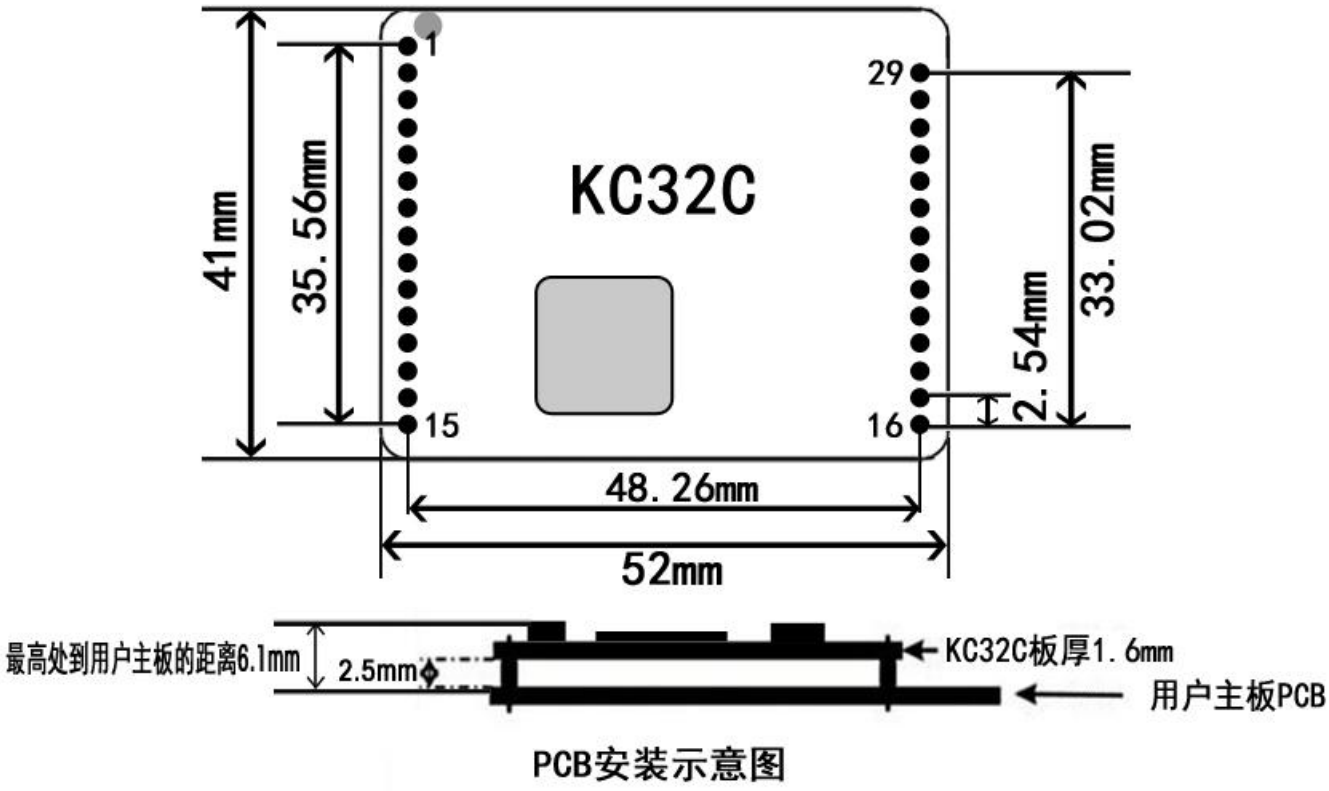
- ✓ 数字音频解码器或模拟音频解码器。
- ✓ 多声道 AV 接收功放。
- ✓ 带解码的多声道多媒体有源音箱。
- ✓ 各种高清影音设备。

深圳市酷唱科技有限公司
Hard & Soft Technology Co., LTD.
地址:深圳市宝安区西乡共乐城 F 栋 2210　　技术支持：support@HSAV.com
电话:0755-27950879　　QQ：1005231106　　业务联系：sales@HSAV.com　　2020 年 6 月 10 日

◆ 地线注意事项

AGND 与 GND 在 KC32C 内未有连接通，需要在用户板上连通，如果+5V 的供电地线与模拟部分的地线不在电源端连通，则接合点在 KC32C 引脚处较近的位置，GND 与金属外壳的地线相连，接线时可连接，使地线阻抗更低，以获得更好的效果，否则在电源供电处连通。建议在 KC32C 较近的位置连通，以取得较好的效果。

◆ KC32C 尺寸图



PCB安装示意图

◆ KC32C 插座端口连接详解

1) SWP SD 卡 SWP 输入及输出。
2) SDO SD 卡 SD0 数据输入及输出。
3) UD+ USB 串行数据 D+输入及输出。
4) UD- USB 串行数据 D-输入及输出。
5) C2K 控制外置音量芯片的串行数据输入输出端，与调试下载的时钟复用。
6) C2D 控制外置音量芯片的串行数据输入输出端，与调试下载的数据复用。
7) EIO 大小喇叭设置或通用的输入输出端口.
8) SCL 通用的输入输出端口，I^2C 通讯端口的 SCL 端口。
9) SDA 通用的输入输出端口，I^2C 通讯端口的 SDA 端口。
10) INT 通用的输入输出端口，I^2C 通讯端口的 INT 端口。
11) 3V3 3.3V 数字输入供电。
12) GND 数码地线输入及输出。
13) RX3 第 3 路 SPDIF 数字 TTL 电平输入，如果用于同轴输入需要增加放大电路。
14) RX2 第 2 路 SPDIF 数字 TTL 电平输入，如果用于同轴输入需要增加放大电路。
15) RX1 第 1 路 SPDIF 数字 TTL 电平输入，如果用于同轴输入需要增加放大电路。

16) MUTE 静音控制信号输出。当静音有效时输出高电平，正常放音为低电平。
17) SCK SD 卡 SCK 时钟输入及输出。
18) SCM SD 卡 SCM 输入及输出。
19) MI2 模拟第 2 路话筒（MIC）输入。
20) MI1 模拟第 1 路话筒（MIC）输入；话筒数码输入时为 I^2S 串行 MADC 模拟话筒转换数据输入。
21) LCH 模拟左声道信号输入；数码输出时为 I^2S 串行数据 ADC 立体声模拟转换数据输入

22) RCH 模拟右声道信号输入；数码输出时为 I²S 串行数据 DA3 后置左右声道音频数据输出。
23) AGND 模拟地线输入及输出，与数码地线并没有连通，必须在外面连通数码地线。
24) FL 模拟前置左声道信号输出；数码输出时为 I²S 串行数据 DA0 前置左右声道音频数据输出。
25) FR 模拟前置右声道信号输出；数码输出时为 I²S 串行数据 DA1 中置超低音声道音频数据输出。
26) CE 模拟中置声道信号输出；数码输出时为 I²S 串行数据 DA2 环绕左右声道音频数据输出。
27) SW 模拟超低音声道信号输出；数码输出时为 I²S 串行 WCK 帧时钟输出。
28) SL 模拟环绕左声道信号输出；数码输出时为 I²S 串行 BCK 主时钟输出。
29) SR 模拟环绕右声道信号输出；数码输出时为 I²S 串行 MCK 主时钟输出。

◆ I2S 音频数字 PCM 输入输出说明

当使用 I2S 数字 PCM 输出接口时，输出的通道最大为 7.1 声道，除了 ADC 为模拟的数据输入外，其余全部为输出，用户需要 I2S 从机与模块连接，以下说明以音频取样 48KHz 为例子说明每个引脚。

MCK 主时钟输出，频率为 48KHz 的 256 倍 12.288MHz。

BCK 位时钟输出，频率为 48KHz 的 64 倍 3.072MHz。

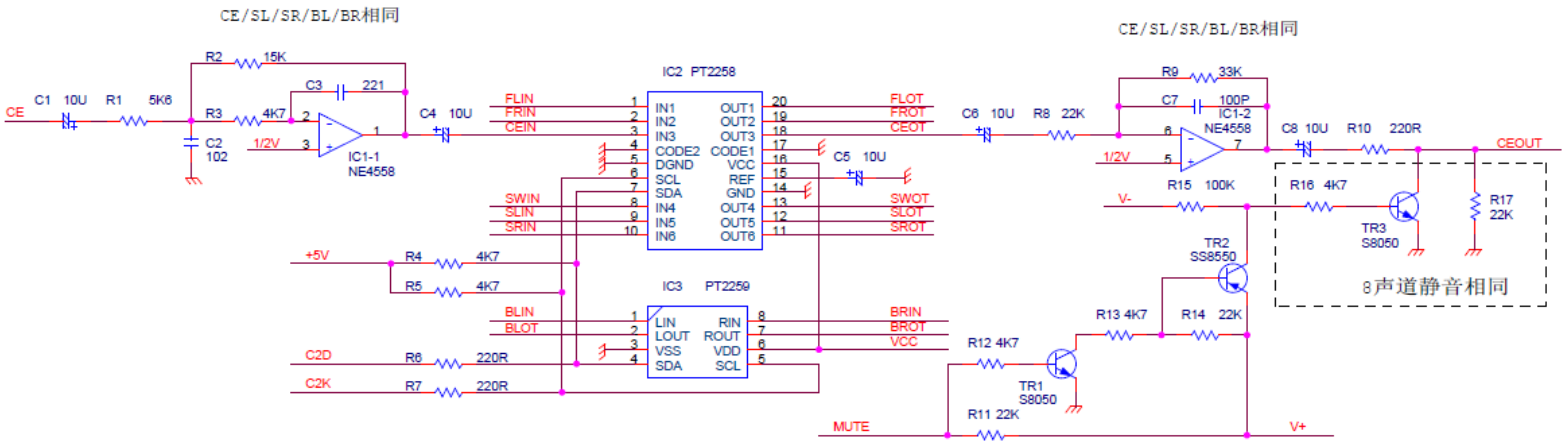
WCK 声道时钟，频率为 48KHz，当 WCK 为低时，传输为通道的左声道。

DA0-DA3 分别为各声道的 32 位数据输出。

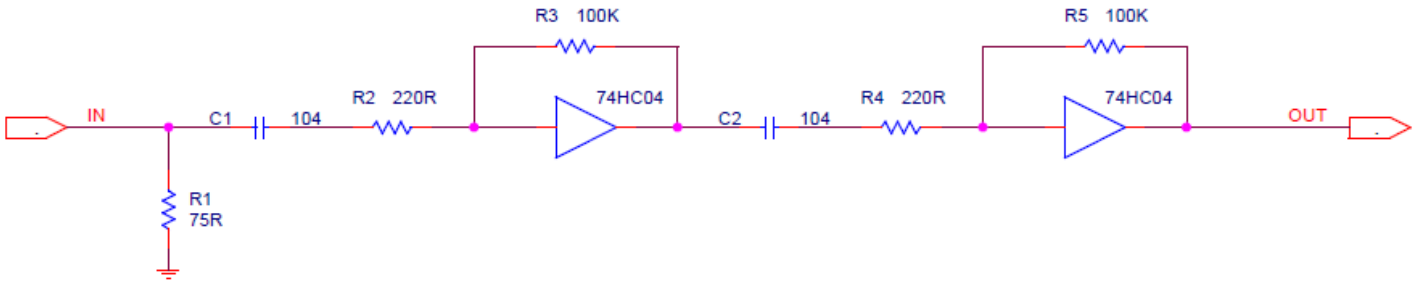
ADC 为模拟的数据输入率，支持最大 32 位。

◆ 音频处理说明

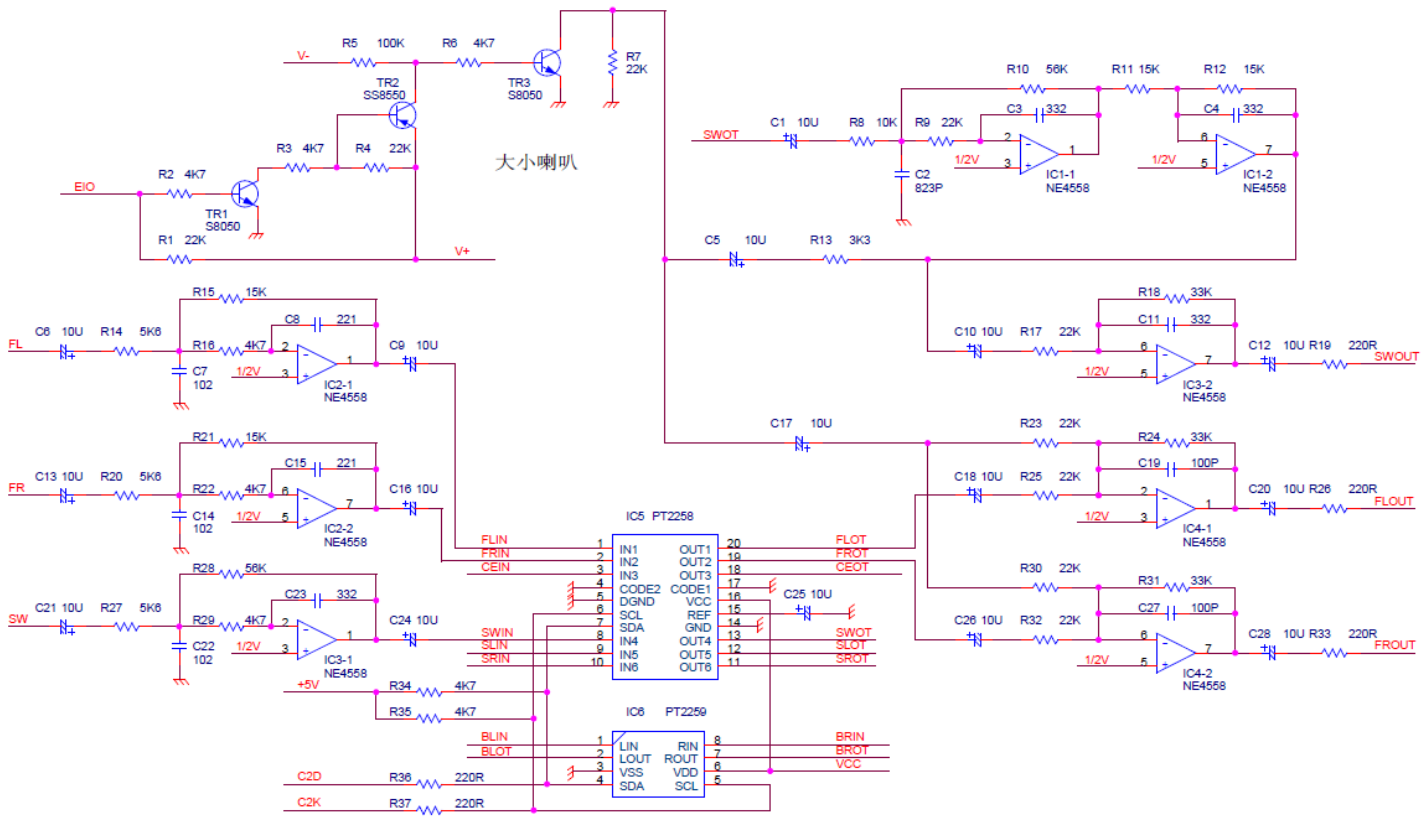
如果整机带有正负电源供电，则最好采用正负电源以获得更好的效果。否则可以采用单电源供电，这时运放的正输入接 1/2 电源，如图下所示。



中置及环绕声低通滤波器（采用单电源供电）、音量放大及静音原理图



同轴输入放大器原理图



前置及超低音通道原理图

◆ 电气规格

序号	项目	最小值	典型	最大值
1	+5V 电源电压	+4.6V	+5V	+5.5V
2	+5V 工作电流	720mA	750mA	770mA
3	数字 RX 输入	0.1V(P-P)	0.5V(P-P)	1.0V(P-P)
4	模拟输入有效检测电平	0.8 Vrms	-	-
5	信噪比(CIR)	-	88dB	-
6	分离度(CIR)	-	87dB	-
7	电平输出@0dB	-	1V	-
8	模拟电平输入	-	1.2V	1.5V
9	频率响应（20Hz-20KHz）	-	+/-1dB	-

◆ 软件二次开发说明

KC32C 提供用户主机订制功能，可以单独完成整机的功能。如果用户产品本身带有单片机时，可以选择采用 I^2C 总线通讯。

如果是两个字节组成 16 位的参数，则第 1 个字节为低位，第 2 个字节为高位。

如果是 4 个字节组成 32 位的参数，则第 1 个字节为低位，第 4 个字节为高位。

0xnn 表示所描述的值不确定，可能为任意值。但其值为原先约定的范围，例如指令长度为 2～137。

B7 表示位于字节的第 7 位，B6 表示位于字节的第 6 位，以此类推；B7:4 使用第 7 至第 4 位。

用户主机写入模块的 I^2C 地址为 0xcc 即 11001100，读取的 I^2C 地址为 0xcd 即 11001101，标准的 I^2C 地址叫法为 0x66，实际是左移了一位。

寄存器长度一般为 8 位，用户主机只需要一个字节的读写即可。另外标注字节长度的寄存器，则需要多个字节读写的，应根据需要进行多字节的读写。

KC32C 写寄存器示意图:

开始	器件地址（写）	寄存器地址	写数据 0	写数据 X（多字节时）	结束

KC32C 读寄存器示意图:

开始	器件地址（写）	待读取的寄存器地址	结束

开始	器件地址（读）	读数据 0	读数据 X（多字节时）	结束

先使用写的器件地址写入待读取的寄存器地址，再使用读的器件地址读入相应的数据。

在对 I^2C 写入每个字节包括数据及地址时，需要接收第 9 位 ACK 位，ACK 位由 KC32C 输出 0。用户主机依靠 ACK 可以获得 KC32C 是否正常工作的信息。

在对 I^2C 读取时每个字节时，需要发送第 9 位 ACK 位，ACK 位由用户主机输出 0。但最后一个字节则需要发送第 9 位 NAK 位，NAK 位由用户主机输出 1。

◆ SDK 开发下载地址

使用标准 8051 的 SDK，带 Windows 电脑必须编译器、编辑器、下载器，只需要一个串就可以在开发板上直接调试程序:

http://www.hsav.com/download/kc3xm51.zip

https://gitee.com/hsav20/kc3xm51.git

https://github.com/hsav20/kc3xm51.git

其余不同平台的 SDK 后面会继续补充。

◆ I^2C 通讯用户主机寄存器地址简表

地址	名称	描述
0x01	KCM_READ_IRQ	读中断请求寄存器，16位寄存器
0x03	KCM_CLEAR_IRQ	清除中断请求寄存器，16位寄存器
0x05	KCM_POWER_ON	用户主机上电寄存器
0x06	KCM_SRC_FORMAT	数码信号输入格式及通道信息指示
0x07	KCM_SRC_FREQ	采样频率及码流率指示
0x08	KCM_VOLUME_MUTE	音频静音及音量加减控制
0x09	KCM_TEST_TONE	噪音测试控制
0x0d	KCM_WIFI_STATUS	WIFI 状态指示
0x0e	KCM_PLAY_STATUS	多媒体文件播放状态指示
0x0f	KCM_PLAY_OPERATE	控制多媒体文件播放模式
0x10	KCM_PLAY_INDEX	控制读取播放文件索引编号，16 位寄存器
0x12	KCM_PLAY_TIME	读取多媒体文件正在播放的时间，16 位寄存器
0x1c	KCM_SRC_VALID	有效的音源输入改变，16 位寄存器
0x1f	KCM_SRC_DETECT	检测所有有效的音源一次

以下读写寄存器带上电记忆，每次上电会自动恢复上次关机的数值

0x20	KCM_INPUT_SOURCE	输入音源选择
0x21	KCM_INPUT_VIDEO	输入视频源选择
0x23	KCM_DYN_COMPRES	杜比数码动态压缩，1 为打开(夜间模式)；0 为正常模式
0x24	KCM_SPK_CONFIG	喇叭设置
0x25	KCM_LPF_FREQ	超低音通道 LPF 低通滤波器频率
0x26	KCM_HPF_FREQ	主声道小喇叭 HPF 高通滤波器频率
0x28	KCM_LIP_SYNC_SET	齿音同步延迟时间，修正对画面与声音不同步
0x29	KCM_LIP_SYNC_MAX	齿音同步最大的延迟时间
0x2b	KCM_LISTEN_MODE	聆听模式选择
0x2c	KCM_EQ_SELECT	多段 EQ 均衡音效处理选择
0x2e	KCM_VOLUME_MAX	设置音量最大值
0x2f	KCM_VOLUME_CTRL	音量值设置
0x30	KCM_FL_TRIM	前置左声道微调
0x31	KCM_FR_TRIM	前置右声道微调
0x32	KCM_CE_TRIM	中置声道微调
0x33	KCM_SW_TRIM	超低音声道微调
0x34	KCM_SL_TRIM	环绕左声道微调
0x35	KCM_SR_TRIM	环绕右声道微调
0x36	KCM_BL_TRIM	后置左声道微调
0x37	KCM_BR_TRIM	后置右声道微调
0x38	KCM_MIC_DELAY	话筒延迟时间，每步 20 毫秒
0x39	KCM_MIC_VOLUME	话筒 1 及话筒 2 音量比例
0x3a	KCM_MIC_ECHO_EQ	话筒回声比例及话筒多段 EQ 均衡音效处理选择
0x3b	KCM_MIC_REPEAT	话筒重复及话筒直达声比例
0x3c	KCM_MIC_REVERB	话筒混响 1 及话筒混响 2 比例
0x3d	KCM_MIC_WHISTLE	话筒啸叫声音反馈模式
0x40	KCM_EXTR_MEMORY	扩展给用户主机的掉电记忆空间，0x40-0x7f 共 64 字节

以下为多字节读写寄存器，这些寄存器不会自动增加寄存器的索引值

0x80	KCM_CUSTOM_CODE	设置用户自定义功能寄存器
0x81	KCM_RD_INFO	读取模块信息寄存器
0x82	KCM_FW_UPGRADE	升级模块固件寄存器
0x83	KCM_RD_RAM	读取指定地址的 RAM 内容
0x86	KCM_MAX_DELAY	读取所有声道最大可用的延迟时间
0x87	KCM_DELAY_TIME	设置所有声道的延迟时间
0x88	KCM_PROGUCE_SIGNAL	模块内部产生的信号配置
0x8b	KCM_EQ_SETUP	多段 EQ 均衡音效处理设置
0x8c	KCM_EQ_VALUE	多段 EQ 均衡音效处理数值
0x8d	KCM_MIC_EQ_VALUE	话筒多段 EQ 均衡音效处理数值
0x8e	KCM_WR_SPECTRUM	设置频谱模式
0x8f	KCM_RD_SPECTRUM	频谱数值读取
0x90	KCM_WR_FLASH	写入 512 字节 FLASH 掉电记忆空间，不可以单独写入
0x91	KCM_RD_FLASH	读取 512 字节 FLASH 掉电记忆空间，不可以单独读取
0x96	KCM_RD_SD_QTY	读取 SD 卡文件总数量，共 2 字节
0x97	KCM_RD_UDISK_QTY	读取 U 盘文件总数量，共 2 字节
0x99	KCM_RD_FILE_TIME	读取多媒体文件正在播放的总时间，共 2 字节单位秒
0xa0	KCM_COMMAND_RCV	云通讯指令接收，0xa8-0xaf 共 8 字节
0xa1	KCM_COMMAND_SEND	云通讯指令发送，0xb0-0xbf 共 16 字节

在 SDK 之中，定义了 typedef enum KC3X_REGISTER，可以直接#include "Kc3xType.h"使用。

◆ 寄存器说明及应用例子

※KCM_READ_IRQ 读中断请求控制，0x01 及 0x02（16 位）读写寄存器

※KCM_CLEAR_IRQ 清除中断请求控制，0x03 及 0x04（16 位）读写寄存器

当用户检测到 INT 端口变低后，需要读取"KCM_READ_IRQ"寄存器用于判断所产生中断的类型。而且需要写入相应的中断类型到"KCM_CLEAR_IRQ"寄存器以清除对应的中断。

寄存器中断位说明:

中断位	名称	描述
B0	KCM_IRQ_SYSTEM_INIT	模式初始化完成中断，需要写入"KCM_POWER_ON"寄存器
B1	KCM_IRQ_FORMAT_INFO	数码信号输入格式改变中断，需要读取"KCM_SRC_FORMAT"寄存器
B2	KCM_IRQ_VOLUME	音量调节改变中断，需要读取" KCM_VOLUME_CTRL"寄存器获取更新的音量值
B3	KCM_IRQ_SRC_VALID	有效的音源输入改变中断，需要读取"KCM_SRC_VALID"寄存器
B4	KCM_IRQ_FIRMWARE	固件更新，需要读取"KCM_RD_INFO"寄存器
B5	KCM_IRQ_PLAY_STATUS	多媒体文件播放改变，需要读取"KCM_PLAY_STATUS"寄存器
B6	KCM_IRQ_PLAY_TIME	多媒体播放时间改变，需要读取"KCM_PLAY_TIME"寄存器
B8	KCM_IRQ_WIFI_RCV	接收到WIFI指令中断，需要读取"KCM_COMMAND_RCV"寄存器获取更新的音量值
B9	KCM_IRQ_WIFI_STATUS	WIFI 状态指示变化中断，需要读取"KCM_WIFI_STATUS"寄存器

Kc3xType.h 定义了 typedef enum KC3X_IRQ_TYPE。

例子:

```
if (!HAL_KCM_I2C_INT()){                                    // INT 端口变低
    BYTE gLocal_1 = MKCM_ReadRegister(KCM_READ_IRQ) ;
    MKCM_WriteRegister(KCM_CLEAR_IRQ, gLocal_1);
    if (gLocal_1 & KCM_IRQ_SYSTEM_INIT){
        // 模块上电，需要读取相应的寄存器以恢复记忆
        // 写入 KCM_CUSTOM_CODE 的值
    }
    if (gLocal_1 & KCM_IRQ_FORMAT_INFO){
        BYTE gLocal_2 = MKCM_ReadRegister(KCM_SRC_FORMAT) ;
        // 数码信号输入格式或者通道信息改变，相应的显示
    }
}
```

※KCM_POWER_ON 用户主机上电及模块重新启动，0x05 只写寄存器

在用户主机上电后，需要写入 0x01 到这个寄存器，然后在收到 KCM_IRQ_SYSTEM_INIT 中断请求（KCM_READ_IRQ 寄存器的 B0）后，再执行相应的初始化动作，这样可以有效地防止通讯双方不同步引起的问题。

写入 0x55 时，模块会重新启动。一般在升级模块固件后需要重新启动。

※KCM_SRC_FORMAT 数码信号输入格式及通道信息指示，0x06 只读寄存器

B7:4 为音源通道信息指示	
数值	说明
0x00	2/0 Lt/Rt Dolby Surround compatible
0x01	1/0 C
0x02	2/0 L/R
0x03	3/0 L/C/R
0x04	2/1 L/R/S

B3:0 为数码音频格式	
数值	说明
0x00	没有信号输入
0x01	模拟信号输入
0x02	PCM 信号输入
0x03	AC3 信号输入
0x04	AC3 HD 信号输入

0x05	3/1 L/C/R/S
0x06	2/2 L/R/SL/SR
0x07	3/2 L/C/R/ SL/SR
0x08	3/3 L/C/R/ SL/SR /CS
0x09	3/4 L/C/R/LS/RS/BL/BR
0x0a	2/3 L/R/LS/RS/CS
0x0b	2/4 L/R/LS/RS/BL/BR

0x05	DTS 信号输入
0x06	DTS HD 信号输入
0x07	AAC 信号输入
0x08	LPCM 信号输入
0x09	HDCD 信号输入
0x0a	MP3 信号输入
其余	保留

Kc3xType.h 定义了 typedef enum KC3X_SRC_TYPE。

※KCM_SRC_FREQ 采样频率及码流率指示，0x07 只读寄存器

B7:3 为音源的码流率

数值	说明		数值	说明		数值	说明		数值	说明
0x00	保留		0x08	112K bps		0x10	448K bps		0x18	2560K bps
0x01	32K bps		0x09	128K bps		0x11	512K bps		0x19	保留
0x02	40K bps		0x0a	160K bps		0x12	640K bps		0x1a	保留
0x03	48K bps		0x0b	192K bps		0x13	896K bps		0x1b	保留
0x04	56K bps		0x0c	224K bps		0x14	1024K bps		0x1c	保留
0x05	64K bps		0x0d	256K bps		0x15	1280K bps		0x1d	保留
0x06	80K bps		0x0e	320K bps		0x16	13792K bps		0x1e	保留
0x07	96K bps		0x0f	384K bps		0x17	20480K bps		0x1f	保留

B2:0 为音源的采样频率，000=保留；001=192 KHz；010=176.4KHz；011=96KHz；100=88.2 KHz；101=48KHz；110=44.1KHz；111=其他频率；

※KCM_VOLUME_MUTE 音频静音及音量加减控制，0x08 写寄存器

B5 为控制音量的加减；只有在 B5 为 1 时 B4 才有效；

B4 为 1 表示音量值加 1，B4 为 0 表示音量值减 1；

B1 为控制音频的静音；只有在 B1 为 1 时 B0 才有效；

B0 为控制整机音频的静音，B0=1 静音打开，这时模块的 MUTE 脚也相应变高；B0=0 静音关闭，这时模块的 MUTE 脚也相应变低。

※KCM_TEST_TONE 噪音测试控制，0x09 写寄存器

B4 为打开噪音测试，B2:0 为对应的通道输出，0-7 依次是 FL、 FR、 CN、 SW、 SL、 SR、 BL、 BR 通道。

例子:

```
MKCM_WriteRegister(KCM_TEST_TONE,  0x12);  // 中置声道噪音测试
MKCM_WriteRegister(KCM_TEST_TONE,  0x00);  // 关闭噪音测试，返回正常的播音模式
```

※KCM_SRC_VALID 有效的音源输入改变， 0x1c 及 0x1d（16 位）读寄存器

位	名称	描述
B0	KCM_SRC_VALID_ANALOG	有信号的音源输入：模拟输入
B1	KCM_SRC_VALID_RX1	有信号的音源输入：数码 1
B2	KCM_SRC_VALID_RX2	有信号的音源输入：数码2
B3	KCM_SRC_VALID_RX3	有信号的音源输入：数码 3
B5	KCM_SRC_VALID_SD	有文件的音源输入：SD插入
B6	KCM_SRC_VALID_UDISK	有文件的音源输入：U盘插入
B7	KCM_SRC_VALID_MIC	有信号的音源输入：话筒插入
B8	KCM_SRC_VALID_HDMI1	有信号的音源输入：HDMI1
B9	KCM_SRC_VALID_HDMI2	有信号的音源输入：HDMI2
B10	KCM_SRC_VALID_HDMI3	有信号的音源输入：HDMI3

B12	KCM_SRC_VALID_UCARD	有信号的音源输入：USB 声卡
B13	KCM_SRC_VALID_E8CH	有信号的音源输入：外置 7.1 声道
B14	KCM_SRC_VALID_BT	有信号的音源输入：蓝牙音频
B15	KCM_SRC_VALID_WIFI	有信号的音源输入：WIFI 音频

Kc3xType.h 定义了 typedef enum KC3X_SRC_VALID。

※KCM_PLAY_STATUS 多媒体文件播放状态，0x0e 读寄存器

B0 正在播放;
B1 正在暂停;
B3:2 重复播放标志，
00 没有重复;
01 重复当前文件;
10 重复当前文件夹;
11 重复所有文件;

※KCM_PLAY_OPERATE 控制多媒体文件播放模式，0x0f 写寄存器

B7:4 控制方式:
0001 停止;
0010 暂停/播放;
0011 只暂停;
0100 只播放;
0101 前一首;
0110 后一首;
1000 快进（速度）;
1001 快退（速度）;
1010 随机播放（时间）;
1010 重复播放（类型）;
B3:0 一些需要多级速度、时间、类型的参数;
Kc3xType.h 定义了 typedef enum KC3X_PLAY_OPERATE。

※KCM_SRC_DETECT 检测所有有效的音源一次，0x1f 写寄存器

寄存器写入检测的指定时间，单位为 100 毫秒。会检测所有的输入音源一次，直到指定的时间到达。完成后会收到 KCM_IRQ_SRC_VALID 中断，读取 KCM_SRC_VALID 寄存器可以获取所有有效的音源。

※KCM_INPUT_SOURCE 输入音源选择，0x20 写寄存器

B6:4 为输入类型选择（B3:0 为对应的通道）:

值	名称	描述
000	KCM_INPUT_ANALOG	音源选择模拟输入
001	KCM_INPUT_DIGITAL	音源选择数码输入，B3:0 为对应的通道，0 为 RX1， 1 为 RX2，2 为 RX3
011	KCM_INPUT_NETWORK	音源选择U盘、TF卡、蓝牙、WIFI、USB声卡、网络等输入
100	KCM_INPUT_SIGNAL	音源选择内部产生信号，B3:0=1为正弦波配合PROGUCE_SIGNAL寄存器可以产生各种不同的正弦波用于生产测试;

Kc3xType.h 定义了 typedef enum KC3X_INPUT_TYPE。

例子:

```
MKCM_WriteRegister(KCM_INPUT_SOURCE,  0x00);   // 选择模拟输入
MKCM_WriteRegister(KCM_INPUT_SOURCE,  0x12);   // 选择为数码 RX2
```

※KCM_SPK_CONFIG 喇叭设置，0x24 读写寄存器

B7:6 为后置喇叭，0 为没有使用、1 为小喇叭、2 为大喇叭;

B5:4 为环绕声喇叭，0 为没有使用、1 为小喇叭、2 为大喇叭;

B3:2 为中置喇叭，0 为没有使用、1 为小喇叭、2 为大喇叭;

B1 为前置喇叭，0 为小喇叭、1 为大喇叭;

B0 为超低音喇叭，0 为没有超低音、1 有超低音。

其中小喇叭表示相应的通道带高通滤波器，只输出高频信号大喇叭为全频输出。

例子:

设置前置大喇叭，中置及环绕声小喇叭，有超低音。

```
MKCM_WriteRegister(KCM_SPK_CONFIG,  0x17)
```

※KCM_LPF_FREQ 超低音通道 LPF 低通滤波器频率，0x25 读写寄存器

超低音的低通滤波器的高频截止频率，有效数值范围 40Hz 至 250Hz，一般推荐 70Hz。

※KCM_HPF_FREQ 主声道小喇叭 HPF 高通滤波器频率，0x26 读写寄存器

当选择小喇叭时，相应的声道就使用本寄存器设置的频率，为高通滤波器的低频截止频率有效数值范围 40Hz 至 250Hz，一般推荐 70Hz。

※KCM_LIP_SYNC_SET 齿音同步延迟时间，修正画面与声音不同步，0x28 读写寄存器

用于修正画面与声音不同步的现像，可以将所有声道的声道一起延迟输出，寄存器的值为延迟时间设置，每步为 2ms，最大时间可以从齿音同步最大的延迟时间寄存器获取。

※KCM_LIP_SYNC_MAX 齿音同步最大的延迟时间，0x29 读写寄存器

齿音同步最大的延迟时间获取，每步为 2ms。

※KCM_LISTEN_MODE 聆听模式选择，0x2b 写寄存器

B5:4 为聆听模式类型选择:

值	名称	描述
00	KCM_LISTEN_STEREO	选择为双声道立体声，B0 为 0 关闭超低音; 为 1 打开超低音;
01	KCM_LISTEN_MODE	选择多声道模式，B1:0 为各种不同算法的多声道模式;
11	KCM_LISTEN_EFECT	选择多声道音效，B1:0为各种不同算法的多声道音效;

例子:

```
MKCM_WriteRegister(KCM_LISTEN_MODE,  0x01);    // 双声道立体声，打开超低音
MKCM_WriteRegister(KCM_ LISTEN_MODE,  0x10);   // 多声道模式
```

※KCM_EQ_SELECT 音效高低音音调或多段 EQ 均衡器通道选择，0x2c 读写寄存器

0 为停止使用音效，1 至 4 分别为 4 组预置音效高低音音调或多段 EQ 均衡器。

※KCM_VOLUME_MAX 设置音量最大值，0x2e 读写寄存器

使用指定的音量芯片节，如果不使用音量芯片则寄存器无效，音量总步数设置，推荐使用 80，表示总音量最大为 80 步。

※KCM_VOLUME_CTRL 音量值设置，0x2f 读写寄存器

使用指定的音量芯片节，如果不使用音量芯片则寄存器无效，音量写入或者读出。0x00 为最小音量，最大音量与 VOLUME_MAX 对应; 一般应用建议使用 VOLUME_MUTE 控制整机音量。

※KCM_FL_TRIM 前置左声道微调，0x30 读写寄存器
※KCM_FR_TRIM 前置右声道微调，0x31 读写寄存器
※KCM_CE_TRIM 中置声道微调，0x32 读写寄存器
※KCM_SW_TRIM 超低音声道微调，0x33 读写寄存器
※KCM_SL_TRIM 环绕左声道微调，0x34 读写寄存器
※KCM_SR_TRIM 环绕右声道微调，0x35 读写寄存器
※KCM_BL_TRIM 后置左声道微调，0x36 读写寄存器
※KCM_BR_TRIM 后置右声道微调，0x37 读写寄存器

所有声道的音量微调，在音量芯片之中调节，如果不使用音量芯片则寄存器无效。
B4 为 0 时 B3:0 的值为增益+0 至+15dB，为 1 时 B3:0 的值为衰减-0 至-15dB。
B3:0 为 dB 的数值不包括符号，0 为增益或衰减 0dB，15 为增益或衰减 15dB。

※KCM_MIC_DELAY 话筒延迟时间，0x38 读写寄存器

话筒延迟时间设置，每步为 20ms。

※KCM_MIC_VOLUME 话筒 1 及话筒 2 音量比例，0x39 读写寄存器

B7:4 为话筒 1 音量，最大为 100%，最小为关闭音量;
B3:0 为话筒 2 音量，最大为 100%，最小为关闭音量。

※KCM_MIC_ECHO_EQ 话筒回声比例及话筒多段 EQ 均衡音效处理选择，0x3a 读写寄存器

B7:4 为话筒多段 EQ 均衡音效处理选择，0001 打开 EQ 音效，0000 关闭音效;
B3:0 为回声比例，最大为 100%，最小为关闭回声的合成。

※KCM_MIC_REPEAT 话筒重复及直达声比例，0x3b 读写寄存器

B7:4 为话筒直达声合成的比例，最大为 100%，最小为关闭直达声的合成;
B3:0 为话筒重复比例，最大为 100%，最小为关闭重复的合成。

※KCM_MIC_REVERB 话筒混响 1 及混响 2 比例，0x3c 读写寄存器

B7:4 为话筒混响 1 合成的比例，最大为 100%，最小为关闭混响 1 的合成;
B3:0 为话筒混响 2 合成的比例，最大为 100%，最小为关闭混响 2 的合成。

※KCM_MIC_WHISTLE 话筒啸叫声音反馈模式，0x3d 读写寄存器

0 为话筒啸叫声音反馈模式关闭。

※KCM_CUSTOM_CODE 设置用户自定义功能寄存器，0x80 读写寄存器

一般应用，当有些用户自定义的功能时使用:
共 4 个字节，每个客户型号都不相同，演示版本为 0x12 0x12 0x00 0x00。读取时，4 个字节与写入的完全相同。
字节 0 及字节 1 为客户型号，每个客户型号都不相同，没有对应的型号可以打不开后面的设置。

字节 2	B2:0 为互换输出声道; B3 为 5.1 的系统之中使用 7.1 功能，额外多了后置的左右声道; B6:4 为设置音量芯片类型。000 为不使用模块内部的音量; 001 为使用模块内部的 DSP 数码音量; 010 为使用模块内部的 DAC 数码音量; 011 为使用 PT2258 + PT2259 或者兼容的音量芯片; 100 为使用 M62446 或者兼容的音量芯片; B7 为每个输入通道单独记忆聆听模式及多段 EQ 均衡音效选择;
字节 3	B2:0 保留为 0 B3 话筒 MIC 与模拟输入交换; B4 使用话筒声音混合功能;

	B6:5 保留为 0 B7 选择 BCK 及 WCK 为输入;

※KCM_RD_INFO 读取模块信息寄存器，0x81 只读寄存器

读取 9 字节的模块信息，可以在固件升级时确认升级成功。

字节 0，模块型号：0x31 为 KC32C；0x53 为 KC35H。

字节 1 至 2 为 16 位的固件时间 time：时=time/1800；分=(time%1800)/30；秒=(time%1800) %30。

字节 3 至 4 为 16 位的固件日期 date：年=(time/372)+2010；月=(time%372)/31+1；日=(time%372)%31+1。

字节 5 至 9 为 32 位的版本号 version(a.b.c)：a=version/1000000；b=version%1000000/1000；c=version%1000000%1000。

※KCM_FW_UPGRADE 升级模块固件寄存器，0x82 读写寄存器

高级应用，当需要通过主板升级模块的固件时使用：

步骤 1、本寄存器写入升级文件的前面 16 字节。

步骤 2、循环读取本寄存器 1 字节，返回 6(正确，继续下一步)、7(文件类型出错)。

步骤 3、本寄存器写入整个升级文件的所有字节。

步骤 4、循环读取本寄存器 1 字节，返回 0(正确) 、1(文件校验出错) 、2(文件长度出错)。

步骤 5、KCM_POWER_ON 寄存器写入 0x55 重启模块。

步骤 6、收到中断 KCM_IRQ_SYSTEM_INIT 后读取 KCM_RD_INFO，对比字节 1 至 8 与文件的字节 12 至 19 相同表示已经使用了新的固件。

※KCM_RD_RAM 读取指定地址的 RAM 内容，0x83 只读寄存器

写入 4 字节指定的 RAM 地址后，可以读取任意长度从指定地址开始的 RAM 内容。可以用于升级固件时校验。

※KCM_MAX_DELAY 读取所有声道最大可用的延迟时间，0x86 只读寄存器

字节 0 至字节 7 分别是前置左、前置右、中置、超低音、环绕左、环绕右、后置左、后置右声道声道的最大可以延迟时间，单位毫秒。

※KCM_DELAY_TIME 设置所有声道的延迟时间，0x87 读写寄存器

字节 0 至字节 7 分别是前置左、前置右、中置、超低音、环绕左、环绕右、后置左、后置右声道声道的最大可以延迟时间，单位毫秒。

※KCM_PROGUCE_SIGNAL 模块内部产生的信号配置，0x88 读写寄存器

字节 0 频率。

※KCM_EQ_SETUP 多路均衡 EQ 音效处理设置，0x8b 读写寄存器

内置 4 组预置均衡 EQ 音效音效记忆，每组可以分别调节不同的音效值，而且独立记忆，使用者只需要选择 KCM_EQ_SELECT 寄存器，即调用相应已记忆的均衡 EQ 音效。用户整机只需要一个简单的音效选择按键，就可以将原来调节的均衡 EQ 音效调出来了，无需写大量的寄存器。

字节	作用	说明
字节 0	选择 4 组预置音效位置	0 为停止使用音效，1 至 4 分别为 4 组预置音效
字节 1	使用均衡的声道标志	对应的通道标志为 1 则使用均衡，0 则停止使用均衡 B4 MIC 声道使用 EQ; B3 后置通道使用 EQ; B2 环绕声通道使用 EQ; B1 中置通道使用 EQ; B0 前置通道使用 EQ;

字节 2	主声道滤波器模式及各个通道段数量	B7:6 为主声道滤波器模式; B5:0 为各个使用声道数量 * 各个通道段数量，不能超出总段数
字节 3	MIC 声道滤波器模式及 EQ 段数量	B7:6 为 MIC 声道滤波器模式; B5:0 为 MIC 声道段数量，不能超出总段数
字节 4	Q 值，数值越大滤波特性越尖	B7:4 为 MIC 声道滤波器 Q 值，0-15; B3:0 为主声道滤波器 Q 值，0-15;

当选择为两段均衡 EQ 音效时，自动转换为音调调节，只调节低音及高音。

当滤波器模式选择 B7:6 为 00 时所有段滤波器相同，B7 则最高段使用 HSF 滤波器，能通过所有高于频率设定值的信号，与音调的高音作用一样；B6 为 1 则最低段使用 LSF 滤波器，能通过所有低于频率设定值的信号，与音调的低音作用一样。

※KCM_EQ_VALUE 多路段均衡 EQ 音效处理数值，0x8c 读写寄存器

字节 0 选择 4 组预置音效位置，0 为停止使用音效，1 至 4 分别为 4 组预置音效;

字节 1 开始每 3 个字节为两段 12 位的 EQ 设置值，其中第一字节为第 1 段低 8 位，第二字节为第 2 段低 8 位，第三字节的 B3:0 为第 1 段高 4 位，B7:4 为第 2 段高 4 位;

主声道每 3 个字节为两段 12 位的 EQ 设置值，紧跟着是 MIC 声道同样的格式。

每段 12 位的格式如下:

B11:8 为 dB 的数值不包括符号，0 为增益或衰减 0dB，15 为增益或衰减 15dB。

B7 为 0 时 B11:8 的值为增益+0 至+15dB，为 1 时 B11:8 的值为衰减-0 至-15dB。

B6:5 为频率的单位范围，B4 至 B0 为频率计算值

B6 至 B5	说明	B4 至 B0
00	20 至 175Hz 频段 每步 5Hz	00000 = 0 * 5 + 20 = 20Hz 至 11111 = 31 * 5 + 20 = 175Hz
01	150 至 1700Hz 频段 每步 50Hz	00000 = 0 * 50 + 150 = 150Hz 至 11111 = 31 * 50 + 150 = 1700Hz
10	1500 至 4600Hz 频段 每步 100Hz	00000 = 0 * 100 + 1500 = 1500Hz 至 11111 = 31 * 100 + 1500 = 4600Hz
11	4.5 至 20KHz 频段 每步 500Hz	00000 = 0 * 500 + 4500 = 4500Hz 至 11111 = 31 * 500 + 4500 = 20000Hz

有效范围为 20Hz 至 20000Hz。

例子:

选择及预置音效第 1 组，前置左、前置右及中置加入 5 段均衡 EQ 音效，频率分别为 50Hz，300 Hz，1000Hz，5000Hz，15000 Hz；增益衰减分别为+15dB，-2dB，-12dB，+5dB，+10dB；Q 值为 128；所有段滤波器相同;

```
BYTE set[] = {
    1, 0x03, 5, 128, 0,
};
BYTE value[] = {
    1,
    0x00|0x00|6, 0x80|0x20|3, 2<<4|15,      // 50Hz, 300Hz  +15dB, -2dB
    0x80|0x20|6, 0x00|0x40|17, 5<<4|12,         // 1000Hz, 5000Hz  -12dB, +5dB,
    0x00|0xc0|21, 0x00|0x20|3, 10,          // 15000Hz,    +10dB,
};
  MKCM_WriteNByte (KCM_EQ_SETUP,  5,  set);
  MKCM_WriteNByte(KCM_EQ_VALUE,  10,  value);
  MKCM_WriteRegister(KCM_EQ_SELECT,  1)。
```

※KCM_WR_SPECTRUM 设置频谱模式，0x8e 读写寄存器

字节 0 选择频谱方式，0 为停止使用频谱；1 为频率电平方式；2 为低速取样点方式；3 为高速取样点方式；

字节 1 至字节 3 为组成两个 12 位的显示缓冲宽度及高度像素，字节 1 的为宽度低 8 位；字节 2 的为高度低 8 位；字节 3 的 B3:0 为宽度高 4 位；字节 3 的 B7:4 为高度高 4 位。显示缓冲高度像素，只支持 8、16、32、64、128、256;，宽度可以是任意值。

如果选择频率电平方式，则从字节 4 开始每两个字节组成 16 位的的频率点，字节数量为宽度像素*2。

※KCM_RD_SPECTRUM 频谱数值读取，0x8f 读寄存器

使用频谱后，用户主机定时（一般 50 至 100 毫秒）像素缓冲用于显示就可以了。

高度	范围	说明
8	每 3 个字节为一组，共 8 列，每组可以装载 8x8 像素点	字节 0 的 B2:0 为第 1 列，B5:3 为第 2 列，字节 1 的 B2:0 为第 3 列，B5:3 为第 4 列，字节 2 的 B2:0 为第 5 列，B5:3 为第 6 列，字节 0、1、2 的 B6 为第 7 列，B7 为第 8 列。
16	每个字节装载两列	B3:0 为第 1 列，B7:4 为第 2 列。
32	每 5 个字节为 1 组，共 8 列，每组可以装载 5x32 像素点	字节 0 至字节 4 的 B4:0 分别为第 1 至第 5 列，B5 为第 6 列，B6 为第 7 列，B7 为第 8 列。
64	每 6 个字节为一组，共 8 列，每组可以装载 6x64 像素点	字节 0 至字节 5 的 B5:0 分别为第 1 列至第 6 列，B6 为第 7 列，B7 为第 8 列。
128	每 7 个字节为一组，共 8 列，每组可以装载 7x128 像素点。	字节 0 至字节 6 的 B6:0 分别为第 1 列至第 7 列，B7 为第 8 列。
256	8 个字节装载 8 个点	字节与像素点完全对应

※调节与寄存器对应关系

一般来说，用户主机在收到上电中断后，从 I2C 相应的寄存器读取上次修改的数值，因为寄存器的数值可能不是连贯的，这时通常写两个函数与之对应，可以参考 SDK 的 kcm_sub.c 之中的方法，用查表法做转换。

1、从寄存器读取的值，调用 MSUB_FromRegister 后，转换到本机处理的值，一般只是上电做一次。

```
BYTE MSUB_FromRegister(BYTE index, BYTE value);
```

例如本机有模拟输入，数码 1 输入及数码 2 输入，对应的寄存器值为 0x30，0x00，0x01，修改表格如下所示

```
CONST_CHAR Tab_InputSwitch[] = {
 0x30, 0x01, 0x02,
}
BYTE gLocal_1 = MKCM_ReadRegister(KCM_INPUT_SRC);
gDIP_InputSource = MSUB_FromRegister(KCM_INPUT_SRC, gLocal_1);
```

gDIP_InputSource 的值就是 0 对应 0x30，1 对应 0x00，2 对应 0x01 了。连贯操作 gDIP_InputSource 就可以轻松做显示了。

2、当需要将修改的值写入寄存器时，调用 MSUB_ToRegister 转换后再写入。

```
BYTE MSUB_ToRegister(BYTE index, BYTE counter);
```

在按键处理处修改 gDIP_InputSource 的值后，如下调用转换函数

```
BYTE gLocal_1 = MSUB_ToRegister (KCM_INPUT_SRC,  gDIP_InputSource );
MKCM_WriteRegister(KCM_INPUT_SRC,  gLocal_1);
```

gLocal_1 的值返回为 0x30， 0x00 或者 0x01，符合寄存器的要求。

3、通过配合整机身功能修改 MSUB_FromRegister 及 MSUB_ToRegister 这两个转换函数，就可以在记忆与本机处理的值

之间轻松转换了。

4、如果在整机设计时有些功能是不能调节的，就需要在初始化时设置为固定的，以避免不同模块时参数被改变的情况。

例如，整机需要固定时所有喇叭都为小喇叭，有超低音的，可以在初始化时

```
BYTE gLocal_1 = MKCM_ReadRegister(KCM_SPK_CONFIG);
if (gLocal !=0x15) { // 如果模块的值与设置的不同
     MKCM_WriteRegister(KCM_SPIC_CONFIG, 0x15);
}
```