

7.1 声道 DTS/杜比数码 AC-3 音频解码 KC32E 用户手册

◆ 特性

- 采用 32 位多核音频 DSP, CS8422 192KHz 数字接收器, 192KHz/24bit ADC 及 DAC。
- 三路 SPDIF 数码输入, 可根据需要设计为光纤或同轴输入。
- 带 I²S 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。
- 模拟音频输出和数码音频输出, 可自行选配不同的低通滤波电路, 以获得不同的效果。
- 支持杜比数码、杜比定向逻辑、DTS、PCM、HD-CD、LPCM 等格式解码, 多种聆听效果重放。
- 环绕声模式可以选择立体声、自动、PROLOGIC IIX、DTS NEO6 及自定义的 DSP 等多种模式。
- 内置 WIFI 接口, 支持云控制及云音乐, 支持电脑及手机 APP 远程控制。
- 带 4 组预置均衡及高低音音调控制, 用户可以做成 4 种音效选择, 每种音效可以独立调整音调或多段频率均衡 (手机 APP 支持)。
- 完整的喇叭设置支持前置喇叭大小, 超低音喇叭安装/不安装, 中置、环绕声及后置喇叭大、小或不安装, 全部可以通过寄存器选择。提供 I/O 口作前置大喇叭/小喇叭叠加控制, 符合杜比及 DTS 标准。
- 音频信号采样功能, 将输入的音频信号频率采样, 可实现多段频谱显示, 带频谱 ALC 增益控制输出, 可实现宽范围的频谱显示, 用户主机直接读取 I2C 数据就可以实现漂亮的频谱显示。
- 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。
- 所有声道都可以进行+/-10dB 的声道微调。
- 小喇叭的高通滤波器截止频率及超低音喇叭的低通滤波器截止频率可以线性调节。
- 云音乐播放功能 (预备中)。
- 环绕声 WIFI 无线传输功能 (预备中), 可以将环绕声道的声音通过 WIFI 发送到无线 WIFI 音箱之中, 用户整机可以不需要使用环绕声左右功放, 最终用户也不需要费心地将功放的输出接到后面的音箱之中。
- 提供静音控制信号输出, 直接用硬件进行静音。
- 可以在线直接升级控制程序, 使用更新的程序可以在网上获得, 极大方便调试及生产维护。
- 使用 I²C 从机接口, 用户主机无需增加额外接口即可使用现成的 I²C 接口。
- I²C 接口可以与其他 I²C 设备并联使用。KC32E 与 24C01 等 I²C 设备完全相同, 非常容易进行二次开发。
- I²C 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。
- 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。
- 全频频宽 (20Hz~20KHz), 用户可以通过低通滤波器调节频宽。
- 内置模拟输入静音, 当模拟没有信号时可自动进行静音, 免除了外接检测电路。
- 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。
- 有多个通用的输出端口, 可以用内置的 CPU 完成整机功能。
- 内置模拟输入静音, 当模拟没有信号时可自动进行静音, 免除了外接检测电路。
- 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。

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

2017 年 12 月 22 日



◆ 应用范围

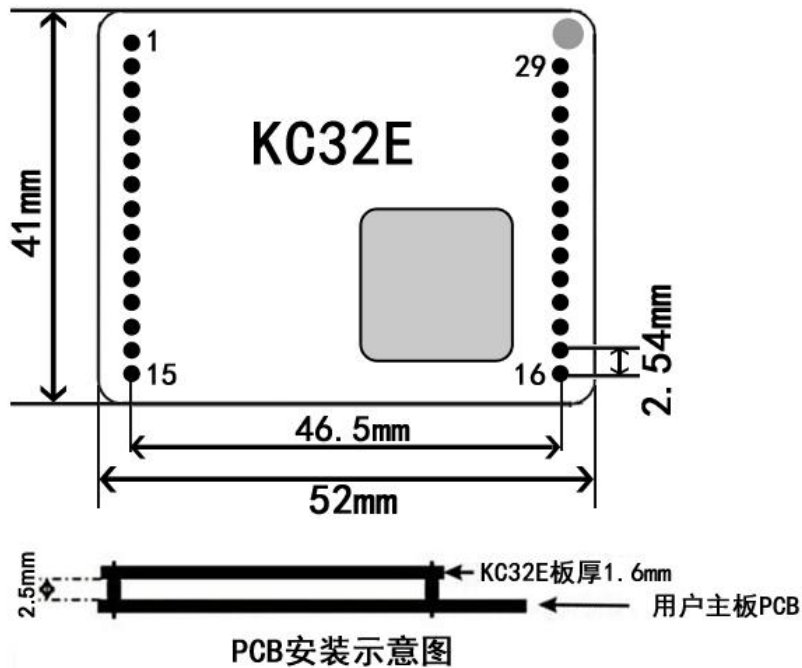
- ✓ 数字音频解码器或模拟音频解码器。
- ✓ 多声道 AV 接收功放。
- ✓ 带解码的多声道多媒体有源音箱。
- ✓ 各种高清影音设备。

◆ 地线注意事项

AGND 与 GND 在 KC32E 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 KC32E 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 KC32E 较近的位置连通，以取得较好的效果。



◆ KC32E 尺寸图



◆ KC32E 插座端口连接详解

- 1) **EB1** 通用的输入输出端口。
- 2) **EBO** 通用的输入输出端口。
- 3) **ADC** 通用的输入输出端口，ADC 输入，一般用于频谱的音频输入。
- 4) **LSW** 通用的输入输出端口，WIFI 配置按键及 LED 显示。
- 5) **C2K** 通用的输入输出端口，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** 通用的输入输出端口，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** 通用的输入输出端口。
- 8) **SCL** 通用的输入输出端口，I²C 通讯端口的 SCL 端口，普通应用为红外线遥控接收输入。
- 9) **SDA** 通用的输入输出端口，I²C 通讯端口的 SDA 端口。
- 10) **INT** 通用的输入输出端口，I²C 通讯端口的 INT 端口，简单的应用也可不用。
- 11) **+5V** 供电+5V 输入。
- 12) **GND** 数码地线输入及输出。
- 13) **RX3** 第 3 路 SPDIF 数字输入。
- 14) **RX2** 第 2 路 SPDIF 数字输入。
- 15) **RX1** 第 1 路 SPDIF 数字输入。

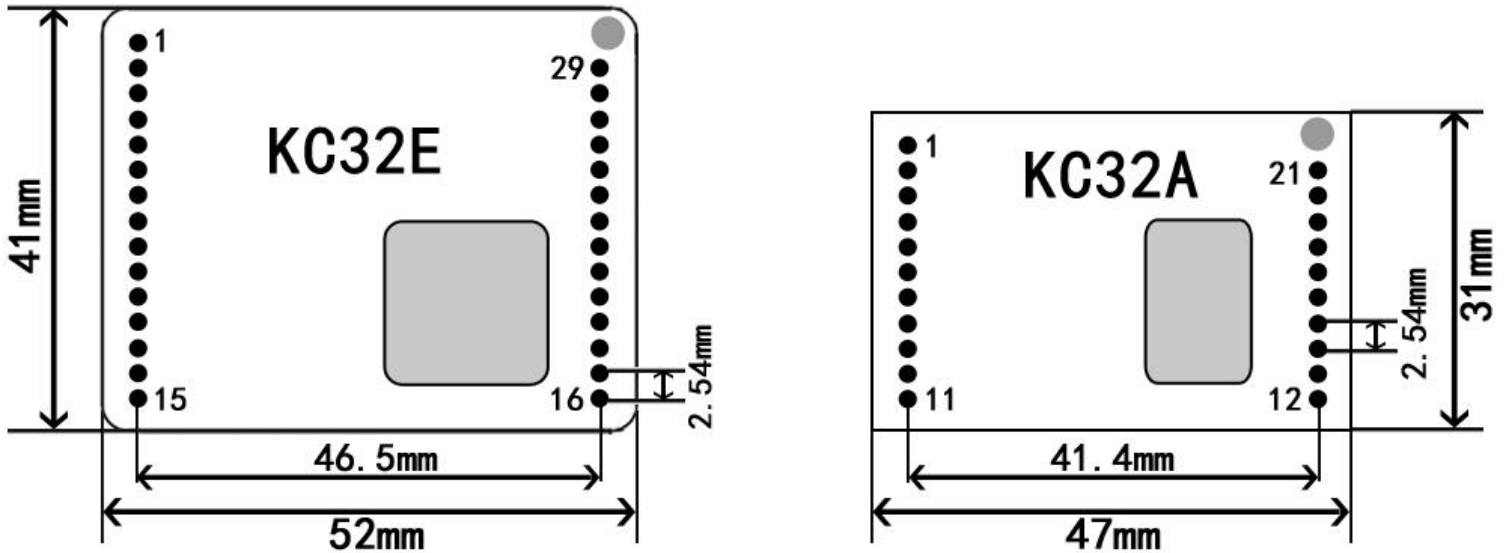
(17、18、19、20、21、22、28、29 黑色详解为模拟输出的说明，蓝色详解为数码音频输出的说明。)

- 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) **SW** 超低音声道信号输出。(SD2 I2S 串行低音中置声道音频数据输出)
- 18) **CE** 中置声道信号输出。(SD1 I2S 串行环绕左右声道音频数据输出)
- 19) **SR** 环绕右声道信号输出。(SD0 I2S 串行前置左右声道音频数据输出)
- 20) **SL** 环绕左声道信号输出。(WCK I2S LRCK 帧时钟: LRCK 的频率=采样频率; 一般使用格式为 32 位, 但实际音频为 24 位)
- 21) **FR** 前置右声道信号输出。(BCK I2S SCLK 位时钟: SCLK 的频率=2*采样频率*采样位数)
- 22) **FL** 前置左声道信号输出。(MCK I2S MCLK 主时钟: MCLK 的频率=256 倍采样频率, 例如 48KHz 时 12.288MHz)

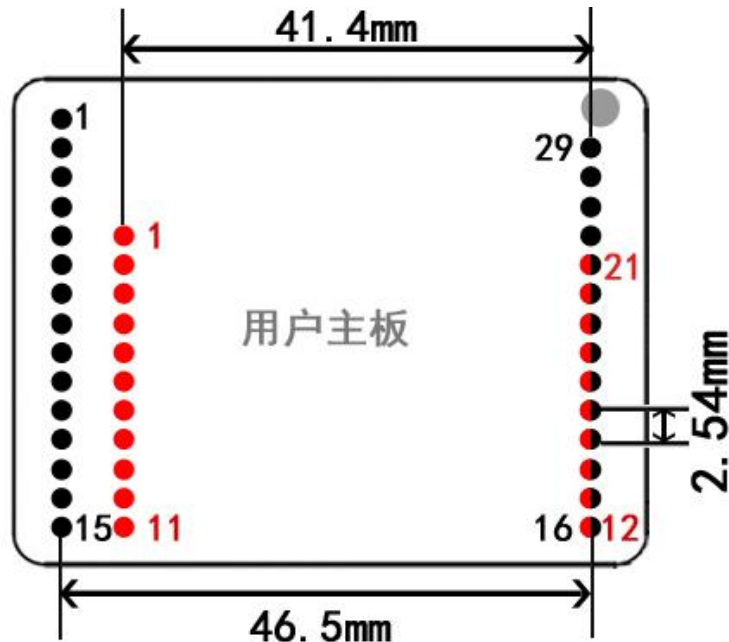


- 23) **AGND** 模拟地线音频输出及供电输入，与数码地线并没有连通，必须在外边连通数码地线。
- 24) **RCH** 模拟右声道信号输入。
- 25) **LCH** 模拟左声道信号输入。
- 26) **PWM** 通用的输入输出端口，PWM 输出，一般用于频谱的 ALC 控制。
- 27) **XIO** 通用的输入输出端口。
- 28) **BR** 模拟后置右声道信号输出。(DAT I2S 串行 DAT 数据输入)
- 29) **BL** 模拟后置左声道信号输出。(SD3 I2S 串行后置左右声道音频数据输出)

◆ KC32E 与 KC32A 的尺寸对比



KC32E与KC32A的尺寸对比

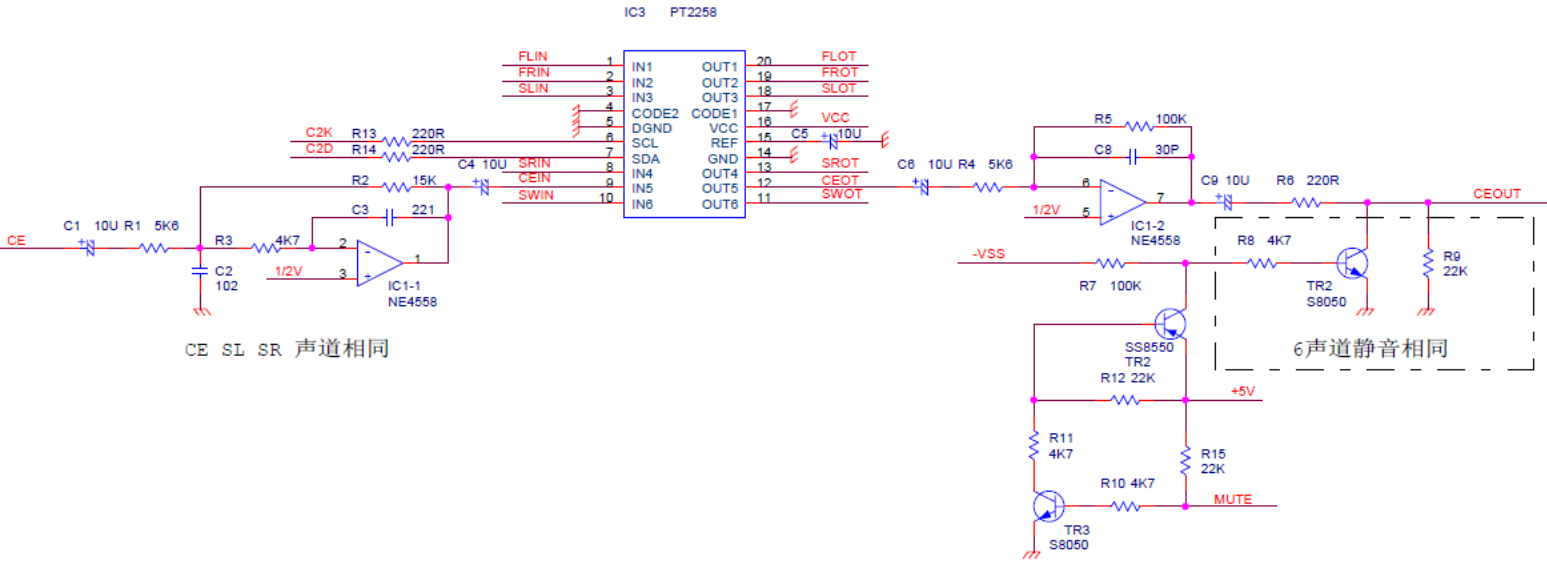


注：用户在设计自己主板的时候，可以预留一个安插 KC32A 尺寸的位置（红色部分）。如果 KC32E 用户不再需要用到 7.1 解码的时候，就可以直接用 5.1 解码的 KC32A 安插在预留的位置上，不需要重新画 PCB 板就能直接使用！安装的时候对照上图红色部分安装就可以了！

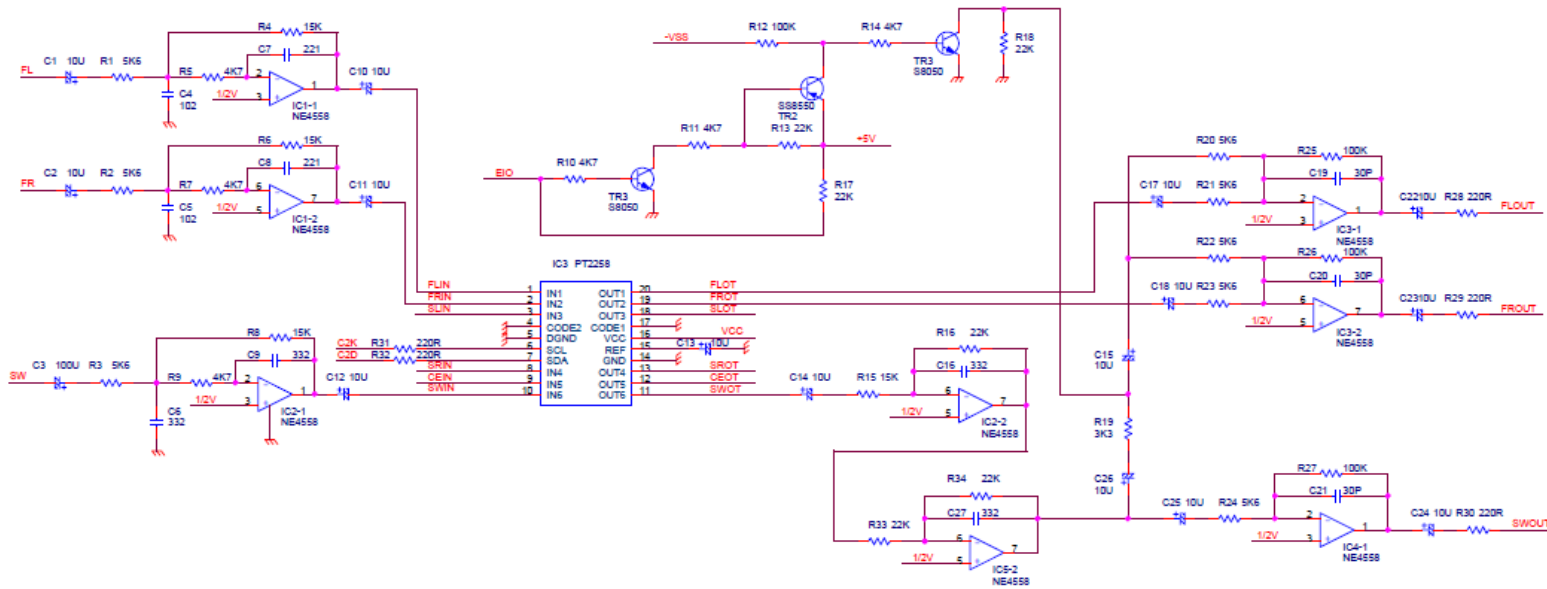


◆ 音频处理说明

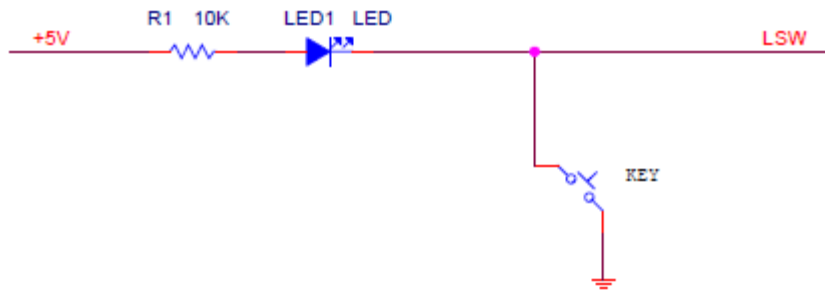
如果整机带有正负电源供电，则最好采用正负电源以获得更好的效果。否则可以采用单电源供电，这时运放的正输入接 1/2 电源，如图下所示。



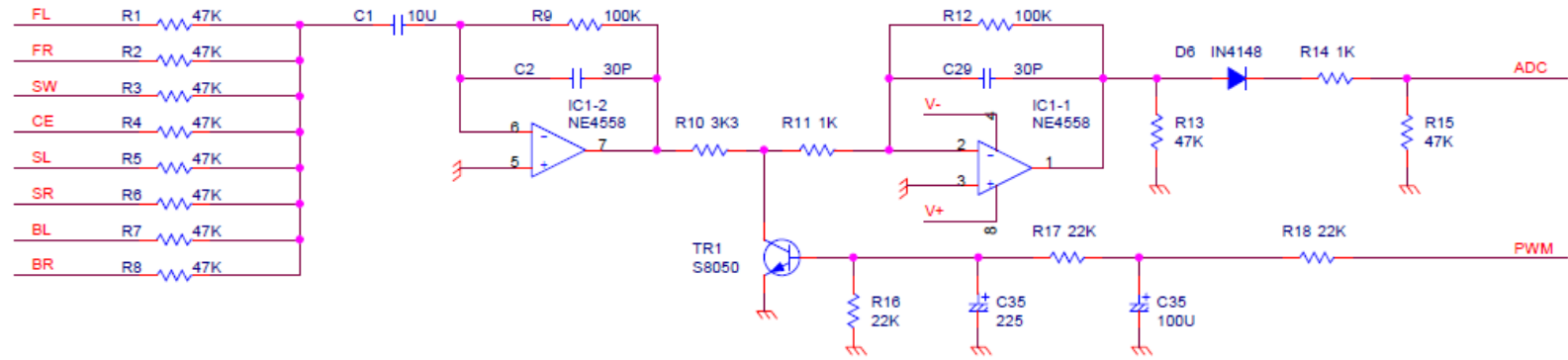
中置及环绕声低通滤波器（采用单电源供电）、音量放大及静音原理图



前置及超低音通道原理图



WIFI 配置原理图



频谱原理图

◆ 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

KC32E 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 I²C 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。

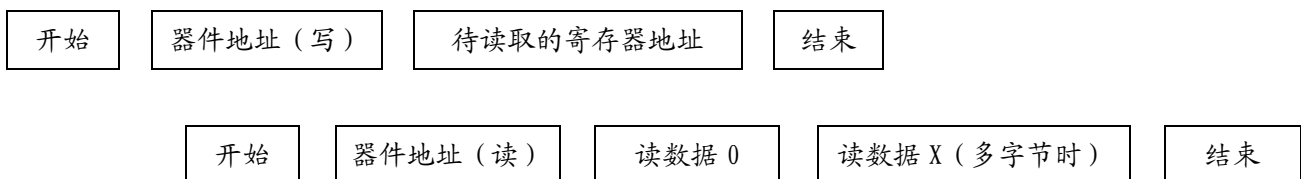
用户主机写入 KC32E 的 I²C 地址为 0xcc 即 11001100，读取的 I²C 地址为 0xcd 即 11001101。

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。

KC32E 写寄存器示意图:



KC32E 读寄存器示意图:





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 I²C 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 KC32E 输出 0。用户主机依靠 ACK 可以获得 KC32E 是否正常工作信息。

在对 I²C 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。

◆ I²C 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
0x03	KCM_CLEAR_IRQ	清除中断请求寄存器
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示
0x07	KCM_SRC_FREQ	采样频率及码流率指示
0x08	KCM_INPUT_SOURCE	输入音源选择
0x09	KCM_INPUT_VIDEO	输入视频源选择
0x0b	KCM_LISTEN_MODE	聆听模式选择
0x0c	KCM_VOLUME_MUTE	音频静音及音量加减控制
0x0d	KCM_VOLUME_CTRL	音量值设置
0x0e	KCM_TEST_TONE	噪音测试控制
0x0f	KCM_WIFI_STATUS	WIFI 状态指示
0x10	KCM_MODEL_TYPE	模块型号
0x11	KCM_VOLUME_MAX	设置音量最大值
0x12	KCM_SPECTRUM_CHIP	设置频谱及音量芯片类型
0x13	KCM_SPK_CONFIG	喇叭设置
0x14	KCM_NOISE_SIGNAL	噪音信号类型
0x15	KCM_FILTER_FREQ	超低音 LPF 及各声道小喇叭 HPF
0x16	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
0x18	KCM_DLT_LIPSYNC	所有声道一起延迟，修正对画面与声音不同步(齿音同步功能)
0x19	KCM_DLT_FRONT	前置声道的延迟时间
0x1a	KCM_DLT_CENTER	中置声道的延迟时间
0x1b	KCM_DLT_SURROUND	环绕声道的延迟时间
0x1c	KCM_DLT_BACK	后置声道的延迟时间
0x20	KCM_FL_TRIM	前置左声道微调
0x21	KCM_FR_TRIM	前置右声道微调
0x22	KCM_CE_TRIM	中置声道微调
0x23	KCM_SW_TRIM	超低音声道微调
0x24	KCM_SL_TRIM	环绕左声道微调
0x25	KCM_SR_TRIM	环绕右声道微调
0x26	KCM_BL_TRIM	后置左声道微调
0x27	KCM_BR_TRIM	后置右声道微调
0x2f 至 0x4f	KCM_PRESET_EQ	多路均衡或音调音效处理及通道选择
0x50 至 0x5f	KCM_SRC_LISTEN	为每个输入通道单独记忆聆听模式使用的掉电记忆空间，共 16 字节
0x60 至 0x9f	KCM_EXTR_MEMORY	扩展给用户主机的掉电记忆空间，共 64 字节
0xa0 至 0xa7	KCM_RD_SPECTRUM	频谱数值读取，共 8 字节
0xa8 至 0xaf	KCM_COMMAND_RCV	云通讯指令接收，共 8 字节
0xb0 至 0xbf	KCM_COMMAND_SEND	云通讯指令发送，共 16 字节



◆ 寄存器说明及应用例子

※KCM-READ-IRQ、KCM-CLEAR-IRQ 中断请求控制， 0x01、0x03 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
B5	KCM_IRQ_WIFI_STATUS	WIFI 状态指示变化中断，需要读取"KCM_WIFI_STATUS"寄存器
B4	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
B3	KCM_IRQ_VOLUME	音量调节改变中断，需要读取"VOLUME_CTRL"寄存器获取更新的音量值
B2	KCM_IRQ_INPUT_SRC	输入音源自动改变中断，需要读取"KCM_INPUT_SRC"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
B0	KCM_IRQ_SYSTEM_INIT	模式初始化完成中断，需要写入"KCM_POWER_ON"寄存器

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 & 0x01) {
        // 解码板上电，需要读取相应的寄存器以恢复记忆
    }
    if (gLocal & 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型号读取， 0x05 读写寄存器

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 0x00 表示进入待机状态。

※KCM-SRC-FORMAT 信号格式及通道信息， 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
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

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	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



※KCM_SRC_FREQ 采样频率及码流率指示, 0x07 只读寄存器

B7: 4 为音源的码流率	
数值	说明
0x00	free
0x01	32K bps
0x02	40K bps
0x03	48K bps
0x04	56K bps
0x05	64K bps
0x06	80K bps
0x07	96K bps
0x08	112K bps
0x09	128K bps
0x0a	160K bps
0x0b	192K bps
0x0c	224K bps
0x0d	256K bps
0x0e	320K bps
0x0f	bad

B3: 0 为音源的采样频率	
数值	说明
0x00	保留
0x01	8 KHz
0x02	11.025 KHz
0x03	12 KHz
0x04	16 KHz
0x05	22.05 KHz
0x06	24 KHz
0x07	32 KHz
0x08	44.1KHz
0x09	48 KHz
0x0a	64 KHz
0x0b	82.2 KHz
0x0c	96 KHz
0x0d	128 KHz
0x0e	176.4 KHz
0x0f	192 KHz

※KCM_INPUT_SOURCE 输入端口选择, 0x08 写寄存器

B5: 4 为输入类型选择:

值	名称	描述
0	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
1	保留	
2	KCM_INPUT_NETWORK	音源选择云音乐输入, 播放网络的声音文件
3	KCM_INPUT_ANALOG	音源选择模拟输入, B3: 0 为对应的通道

例子:

```
MKCM_WriteRegister(KCM_INPUT_SOURCE, 0x30); // 选择模拟输入
MKCM_WriteRegister(KCM_INPUT_SOURCE, 0x02); // 选择为数码 RX2。
```

※KCM_LISTEN_MODE 聆听模式, 0x0b 写寄存器

B5: 4 为聆听模式类型选择:

值	名称	描述
0	KCM_LISTEN_STEREO	B0 为 0 选择为两声道立体声
0	KCM_LISTEN_AUTO	B0 为 1 选择多声道自动模式
1	KCM_LISTEN_PL2X	杜比定向逻辑 II, B0 为 0 选择为 KCM_PL2X_MUSIC, B0 为 1 选择为 KCM_PL2X_MOVIE
2	KCM_LISTEN_NE06	声场矩阵 DTS NE06 模式, B0 为 0 选择为 KCM_NE06_MUSIC, B0 为 1 选择为 KCM_NE06_CINEMA
3	KCM_LISTEN_DSP	DSP EFFECT 声场效果, B3: 0 选择从 0-4 共 5 种不同的模式

例子:

```
MKCM_WriteRegister(KCM_LISTEN_MODE, 0x32); // DSP 多声道效果 2
MKCM_WriteRegister(KCM_LISTEN_MODE, 0x10); // 选择杜比定向逻辑 II MUSIC 聆听模式
```



※KCM-VOLUME-MUTE 音频静音及音量加减控制, 0x0c 写寄存器

B2 为控制音量的加减; 只有在 B2 为 1 时, B1 才有效

B1 为 1 表示音量值加 1, B1 为 0 表示音量值减 1;

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-VOLUME-CTRL 音量值设置, 0x0d 写寄存器

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-TEST-TONE 噪音测试, 0x0e 写寄存器

B3 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, 0x0a); // 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-VOLUME-MAX 设置音量最大值, 0x11 读写寄存器

音量总步数设置, 一般为 63, 表示总音量最大为 64 步。

※KCM-SPECTRUM-CHIP 设置频谱及音量芯片类型, 0x12 读写寄存器

B7: 4 为设置频谱类型, 0 为不使用频谱; 1 为五段频谱, 每段使用 3 位的值。如果使用其他频谱类型, 可以联系订制相应的驱动软件。

B3: 0 为设置音量芯片类型, 0 为不使用模块控制音量; 1 为使用模块内部的 DAC 或者 DSP 数码音量; 2 为使用 PT2258 + PT2259 或者兼容的音量芯片; 3 为使用 M62446 或者兼容的音量芯片;

如果使用其他频谱类型或音量芯片, 可以联系订制相应的驱动软件。

※KCM-SPK-CONFIG 喇叭设置, 0x13 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-NOISE-SIGNAL 噪音信号类型, 0x14 读写寄存器

0x00 = White Noise; 0x01 = Dolby Pink Noise; 0x02 = THX Pink Noise; 0x03 = Bass Noise; 0x04 = Sine Wave;
0x05 = Square Wave; 0x06 = DC Wave; 0x07 = Impulse Wave

一般常规使用 Dolby Pink Noise

※KCM-FILTER-FREQ 超低音 LPF 及各声道小喇叭 HPF, 0x15 读写寄存器

有效数值范围为 0x00 到 0x13。0x00 对应频率为 40Hz, 0x13 对应频率为 250Hz, 每步 5Hz。当选择小喇叭时, 相应的声道就使用这里设置的频率, 为高通滤波器的低频截止频率。同时作为超低音的低通滤波器的高频截止频率。



※KCM-PRESET_EQ 多路均衡或音调音效处理及通道选择, 0x2f-0x4f 读写寄存器

内置 4 组预置音效选择, 每组可以分别调节音调, 选择 PRESET_EQ 寄存器, 即调用相应已记忆的音调。同样地, 这时修改 EQ-TONE 寄存器也会修改为前的音调设置。用户整机只需要一个简单的按键, 就可以将原来调节的音调或均衡调出来了, 无需写大量的寄存器。

PRESET_EQ 类型选择:	0	1	2	3
均衡段对应地址:	寄存器地址	寄存器地址	寄存器地址	寄存器地址
KCM_EQ_B1_CONFIG	0x30	0x38	0x40	0x48
KCM_EQ_B2_B3	0x31	0x39	0x41	0x49
KCM_EQ_B4_B5	0x32	0x3a	0x42	0x4a
KCM_EQ_B6_B7	0x33	0x3b	0x43	0x4b
KCM_EQ_B8_B9	0x34	0x3c	0x44	0x4c
KCM_EQ_B10_B11	0x35	0x3d	0x45	0x4d
KCM_EQ_B12_B13	0x36	0x3e	0x46	0x4e
KCM_EQ_B14_B15	0x37	0x3f	0x47	0x4f

音调或均衡, 4 位的最高位 B3 为 1 代表负数, B2: 0 为每步 2dB 的数值, 最大值是 7, 表示+或者-14dB。

例子:

使用预置音效第 3 组, 同时将低音选择为+14dB, 中音选择为-2dB, 高音选择为-4dB

```

MKCM_WriteRegister(KCM.PRESET_EQ, 0x02);
MKCM_WriteRegister(KCM_EQ_B1_CONFIG + (2 * 8), 0x70);
MKCM_WriteRegister(KCM_EQ_B2_B3 + (2 * 8), 0xa9)。

```

※KCM-RD_SPECTRUM 频谱数值读取, 0xa0-0xa7 读寄存器

类型 0x00: 共分为 100、330、1K、3K3、10K 五段频谱, 每个频谱用 3BIT, 值从 0x00-0x07。使用 0x50-0x51 寄存器组成 16 位。B14:12 第 5 段(10K) B11:9 第 4 段(3K3) B8:6 第 3 段(1K) B5:3 第 2 段(330 Hz) B2:0 第 1 段(100Hz) 如果使用其他频谱类型, 可以联系订制相应的驱动软件。

※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_1 != 0x15) { // 如果模块的值与设置的不同  
    MKCM_WriteRegister(KCM-SPIC_CONFIG, 0x15);  
}
```

◆ KC32E 的三种模拟说明（可以根据自己的需要修改）

- ①用内部的 ADC 和 DAC 时，电阻排 R3、R4 焊上，R2、R5、R1 不焊；
为 7.1 模拟输出，立体声模拟（LCH、RCH）输入。
- ②使用内部 ADC，外部 DAC 时，电阻排 R2、R5 焊上，R3、R4、R1 不焊；
为 I²S 输出，立体声模拟（LCH、RCH）输入。
- ③使用外部 ADC 和 DAC 时，电阻排 R2、R5 和电阻 R1（R1 电阻用 0R 或 10R）焊上，R3、R4 不焊，拆了 CE2826；
为 I²S 输出，立体声 I²S 输入。