

# SHARC 与 MCU 控制协议 v0.4

2012.09.06

## 目录

ID=0：控制所有模块的状态	2
ID=1：控制噪声门（噪声门开启信号返回值请看最后一页）	2
ID=2：控制移频	2
ID=3：控制高低通滤波器	3
ID=4：控制 PEQ	3
ID=5：控制压限	4
ID=6：控制电平大小	4
ID=7：控制电平表（电平返回值请看最后一页）	4
ID=8：控制混音器	5
ID=9：控制延时	5
ID=10：控制正弦波发生器频率	6
ID=11：//往任意地址里面写任意值	6
ID=12：//从任意地址里面读任意值      此值会在“DSP 处理结果”字中返回	6
ID=7 控制电平表 DSP 数据返回格式	7
ID=1 控制噪声门（噪声门开启信号数据返回格式）	7
其他 ID 返回格式：	7

注：ID=11 和 ID=12 是在 v0.3 基础上添加的

MCU 发给 DSP 协议格式：（每个单元由一个字（4 个字节）组成），注意：发送时低字节先发

### ID=0：控制所有模块的状态

帧头	ID/长度	首地址	数据(int)	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外，所有字的个数	模块地址	模块状态： 0:无效;1:有效; 2:静音 3:旁路 整型	长度+首地址+数据长度+具体数据之和

例如：要改变 ScalerCH1\_1 模块的状态(Layout1\_ScalerCH1\_1 = 0x000C056A)为静音，需要发送给 DSP 的指令如下：

```
0xDEAD BEEF // 帧头 发送按照 EF BE AD DE 低字节先发送
0x0004 0000 //长度=4; ID=0
0x000C056A //模块地址
0x000000002 // 2:静音
0xxxxxxxx //CRC 校验=0x00040000+ 0x000C056A +0x2
```

### ID=1：控制噪声门（噪声门开启信号返回值请看最后一页）

帧头	ID/长度	首地址	threshold	attackTime	decayTime	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外，所有字的个数	噪声门模块地址；	-120---0 dB 数据类型： 浮点	20---1000ms 浮点	20---1000ms 浮点	ID/长度+首地址+数据长度+具体数据之和

例如：要设置 NoiseGate1 模块 (Layout1\_NoiseGate1=0x000C0500)，需要发送给 DSP 的指令如下：

```
0xDEAD BEEF // 帧头，发送按照 EF BE AD DE 低字节先发送
0x01060000 //长度=6; ID=1
0x000C0500 // 模块地址
0xC1200000 // threshold = -10; 因为 threshold 的类型是浮点，发给 DSP 的也要是 32 位表示的浮点数
0x 41A00000 // attackTime = 20.0
0x 41A00000 // decayTime = 20.0
0xxxxxxxx //CRC 校验=0x01040000+0x000C0500 +0xC1200000+0x 41A00000+0x 41A00000
```

### ID=2：控制移频

帧头	ID/长度	移频量	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外，所有字的个数	1---10Hz 整型	ID/长度+首地址+数据长度+具体数据之和

例如：要设置移频量是 1Hz，需要发送给 DSP 的指令如下：

```
0xDEAD BEEF // 帧头，发送按照 EF BE AD DE 低字节先发送
0x02030000 //长度=3; ID=2
0x00000001 // 移频量
0xxxxxxxx //CRC 校验同上
```

**ID=3: 控制高低通滤波器**

帧头	ID/长度	首地址	HLPF & Type & Slope	freq	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外，所有字的个数	高低通滤波器模块地址；	HLPF: 1---2 Type: 0---2; Slope: 0---6 整型	10--- 20k Hz 浮点	ID/长度+首地址+数据长度+具体数据之和

注: **HLPF: 1= HPF; 2= LPF; 位于低 8 位**

**Type: 0: Bessel; 1: Butterworth; 2: Link-Riley 位于高 8 位**

**Slope: 0: -6dB 1: -12dB 2: -18dB 3: -24dB 4: -36dB 5: -48dB 位于高 16 位**

例如: 要设置 HighPassCH1 模块 (\_Layout1\_HighPassCH1=0x000C052F), 需要发送给 DSP 的指令如下:

```

0xDEAD BEEF // 帧头
0x03050000 // 长度=5; ID=3
0x000C0500 // 模块地址
0x00030201 // HLPF =1:HPF; Type =2: Link-Riley; Slope=3: -24dB
0x 447A0000 // freq=1000.0
0xxxxxxx //CRC 校验同上

```

**ID=4: 控制 PEQ**

帧头	ID/长度	首地址	Ch&Type& Band	freq	gain	Q	CRC
0xDEADBEEF	ID<<24   Len<<16 Len= 除 帧 头 外，所有 字的个数	PEQ 模块地址	Ch : 0---31 Type: 3---5 Band =0---7 整型	10--- 20000Hz 浮点	-20--- 20dB 浮点	0.5---20 浮点	ID/ 长 度 + 首地址+数据长度+具体数据之和

注: **Ch=0---31: 0~15: 输入通道 1~16; 16~31: 输出通道 1~16, 位于低 8 位**

**Type=3---5: 3=PEQ; 4= lowShelf; 5= highShelf, 位于高 8 位**

**Band =0---30, 共 31 段, (注: 前面只有 0~14: 共 15 段), 位于高 16 位**

**Gain = 0 时, 表示此段 PEQ Bypass**

例如: 要设置 PEQ5BandCH1 (\_Layout1\_PEQ5BandCH1=0x000C0538), 需要发送给 DSP 的指令如下:

```

0xDEAD BEEF // 帧头
0x04070000 // 长度=7; ID=4
0x000C0538 // 模块地址
0x00010302 // Ch=2; Type=3= PEQ; Band =1;
0x 447A0000 // freq=1000.0
0x 00 // gain = 0.0
0x 3F800000 // Q = 1.0
0xxxxxxx //CRC 校验同上

```

**ID=5: 控制压限**

帧头	ID/长度	首地址	Thres hold	gain	Knee Depth	ratio	Attack Time	Decay Time	CRC
0xDEADBEEF	ID<<24   Len<<16 Len= 除帧头 外, 所有字的 个数	压限模块 地址;	-120--- 0 dB 浮点	0--- 100 浮点	0.1--- 60 浮点	1--- 100 浮点	20--- 1000ms 浮点	20--- 1000ms 浮点	ID/ 长度 + 首地址 + 数据长度 + 具体数 据之和

例如: 要设置 **Limiter1** 模块 (**\_Layout1\_AGCLimiterCore1=0x000C0554**), 需要发送给 **DSP** 的指令如下:

**0xDEAD BEEF** // 帧头, 发送按照 **EF BE AD DE** 低字节先发送

**0x05090000** // 长度=9; ID=5

**0x000C0554** // 模块地址

**0xC1200000** // threshold = -10;

**0x00000000** // gain = 0;

**0x3F800000** // kneedepth = 1;

**0x3F800000** // ratio= 1;

**0x 41A00000** // attackTime = 20.0

**0x 41A00000** // decayTime = 20.0

**0xxxxxxxx** //CRC 校验同上

**ID=6: 控制电平大小**

帧头	ID/长度	首地址	amp	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外, 所 有字的个数	电平模块 地址;	-1.0---1.0 浮点	ID/ 长度 + 首地 址 + 数据长度 + 具体数据之和

例如: 要设置 **ScalerCH1\_1** 模块 (**\_Layout1\_ScalerCH1\_1=0x000C056A**), 需要发送给 **DSP** 的指令如下:

**0xDEAD BEEF** // 帧头, 发送按照 **EF BE AD DE** 低字节先发送

**0x06040000** // 长度=4; ID=6

**0x000C056A** // 模块地址

**0x 3F800000** // amp =1.0

**0xxxxxxxx** //CRC 校验同上

**ID=7: 控制电平表 (电平返回值请看最后一页)**

帧头	ID/长度	首地址	attackTime	decayTime	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外, 所 有字的个数	电平表模 块的地址;	3---1000ms 浮点	3---1000ms 浮点	ID/长度+首 地址+数据 长度+具体 数据之和

例如: 要设置 **ScalerCH1\_1** 模块 (**\_Layout1\_CH1meter=0x000C05C2**), 需要发送给 **DSP** 的指令如下:

**0xDEAD BEEF** // 帧头, 发送按照 **EF BE AD DE** 低字节先发送

**0x07050000** // 长度=5; ID=7

**0x000C05C2** // 模块地址

**0x 41A00000** // attackTime = 20.0

**0x 41A00000** // decayTime = 20.0

**0xxxxxxxx** //CRC 校验同上

**ID=8: 控制混音器**

帧头	ID/长度	outch	Value1	Value2	Value3	Value4	Value5	CRC
0xDEADBEEF	ID<<24   Len<<16 Len= 除 帧头外, 所有字的个数	Ch1---ch16: 0---15 整型	整型	整型	整型	整型	整型	ID/长度 + 首地址 + 数据长度 + 具体数据之和

注:  $\text{Value1} = v1 \ll 24 \mid v2 \ll 16 \mid v3 \ll 8 \mid v4$ ;

$\text{Value2} = v5 \ll 24 \mid v6 \ll 16 \mid v7 \ll 8 \mid v8$ ;

$\text{Value3} = v9 \ll 24 \mid v10 \ll 16 \mid v11 \ll 8 \mid v12$ ;

$\text{Value4} = v13 \ll 24 \mid v14 \ll 16 \mid v15 \ll 8 \mid v16$ ;

$\text{Value5} = v17 \ll 24 \mid v18 \ll 16$ ;

某个输出通道 **Outch** 的电平 =  $v1 \cdot in1 + v2 \cdot in2 + \dots + v17 \cdot in3 + v18 \cdot in4$

**v1—v18** 分别表示 **18** 个输入通道的电平值

发送时  $v \cdot 100$ , 比如界面上混音值是 **0.05**, 则发送值是 **5**

则混音的最大值只能是 **2.55** (乘以 **100** 后, 就是 **255**),

注: 此协议只支持 **18\*16** 的 **Mixer** 模块, 如要控制输入输出数不一样的 **Mixer**, 需要修改协议

例如: 要设置混音器, 需要发送给 **DSP** 的指令如下:

**0xDEADBEEF** // 帧头, 发送按照 **EF BE AD DE** 低字节先发送

**0x08080000** // 长度=8; ID=8

**0x 0** // outch = 0

**0x010001000** // v1=1; v2=0; v3=1; v4=0

**0x 010001000** // v5=1; v6=0; v7=1; v8=0

**0x 010001000** // v9=1; v10=0; v11=1; v12=0

**0x 010001000** // v13=1; v14=0; v15=1; v16=0

**0x 010000000** // v17=1; v18=0;

**0xxxxxxxx** //CRC 校验同上

**ID=9: 控制延时**

帧头	ID/长度	首地址	delay	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外, 所有字的个数	延时模块地址;	1.34—maxDelay ms 浮点	ID/长度+首地址+数据长度+具体数据之和

例如: 要设置 **DelayOffChipCH1** 模块 (**\_Layout1\_DelayOffChipCH1=0x000C05B2**), 需要发送给 **DSP** 的指令如下:

**0xDEADBEEF** // 帧头, 发送按照 **EF BE AD DE** 低字节先发送

**0x09040000** // 长度=4; ID=9

**0x000C05B2** // 模块地址

**0x41A00000** // delay = 20ms

**0xxxxxxxx** //CRC 校验同上

**ID=10: 控制正弦波发生器频率**

帧头	ID/长度	首地址	频率	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外, 所有字的个数	正弦模块地址;	0---24k 浮点	ID/长度+首地址+数据长度+具体数据之和

例如: 要设置 **TestGenSine** 模块 (**\_Layout1\_TestGenSine=0x000B4208**) 频率为 **1000Hz**, 需要发送给 **DSP** 的指令如下:

```

0xDEAD BEEF // 帧头, 发送按照 EF BE AD DE 低字节先发送
0x0a040000 // 长度=4; ID=a
0x000B4208 // 模块地址
0x447A0000 // 1000Hz
0xxxxxxxxx //CRC 校验同上

```

**ID=11: //往任意地址里面写任意值**

帧头	ID/长度	首地址	具体数据	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外, 所有字的个数	音频模块的参数地址;	具体写入什么数据	ID/长度+首地址+数据长度+具体数据之和

例 1: 要把 **level** 模块的 **amps** (**\_Layout1\_levelamps=0x000B88DC**) 设成 **1.0**, 需要发送给 **DSP** 的指令如下:

```

0xDEAD BEEF // 帧头, 发送按照 EF BE AD DE 低字节先发送
0x0B040000 // 长度=4; ID=11
0x000B88DC // amps 参数地址
0x3F800000 // 1.0; 因为 amp 的类型是浮点, 所以发给 DSP 的也要是 32 位表示的浮点数
0x 3F8F88DC //CRC 校验=0x00040000+0x000B88DC +0x3F800000

```

**ID=12: //从任意地址里面读任意值**

此值会在“**DSP 处理结果**”字中返回

帧头	ID/长度	地址	CRC
0xDEADBEEF	ID<<24   Len<<16 Len=除帧头外, 所有字的个数	模块参数的地址	长度+首地址+数据长度+具体数据之和

例如: 要读取 **level** 模块的当前值(**\_Layout1\_levelamps=0x000B88DC**), 需要发送给 **DSP** 的指令如下:

```

0xDEAD BEEF // 帧头
0x0C03 0000 // 长度=3; ID=12
0x000B88DC // 参数地址
0xxxxxxxxx //CRC 校验

```

**数据返回格式:****ID=7 控制电平表 DSP 数据返回格式****DSP 返回 MCU 协议格式:**

帧头	长度<<16	Cur	Pk	CRC
0xDEADBEEF	0x00040004	浮点 单位: dB	浮点 单位: dB	ID/ 长度 + 处理结果之和

例如: **0xDEAD BEEF** // 帧头  
**0x00040004** //长度=4, 固定值 4  
**0x3F800000** // Cur = 1.0  
**0x3F800000** // Pk = 1.0  
**0xxxxxxxx** //CRC 校验同上

**ID=1 控制噪声门 (噪声门开启信号数据返回格式)****DSP 返回 MCU 协议格式:**

帧头	长度<<16	NoiseGateOn	CRC
0xDEADBEEF	0x00030004	0: 噪声门关闭 1: 噪声门开启	ID/ 长度 +处理结果之和

例如: **0xDEAD BEEF** // 帧头  
**0x00030004** //长度=3, 固定值 4  
**0x00000001** // NoiseGateOn = 1  
**0xxxxxxxx** //CRC 校验同上

**其他 ID 返回格式:****DSP 返回 MCU 协议格式:**

帧头	长度<<16	DSP 处理结果	CRC
0xDEADBEEF	0x00030004	0---2	ID/ 长度 + 处理结果之和

例如: **0xDEAD BEEF** // 帧头  
**0x00030004** //长度=3, 固定值 4  
**0x0** // DSP 处理结果 = 0: 没错  
**0xxxxxxxx** //CRC 校验同上

**DSP 处理结果:****0: NO\_ERROR****1: CRC\_ERROR****2: NOT\_HANDLED**