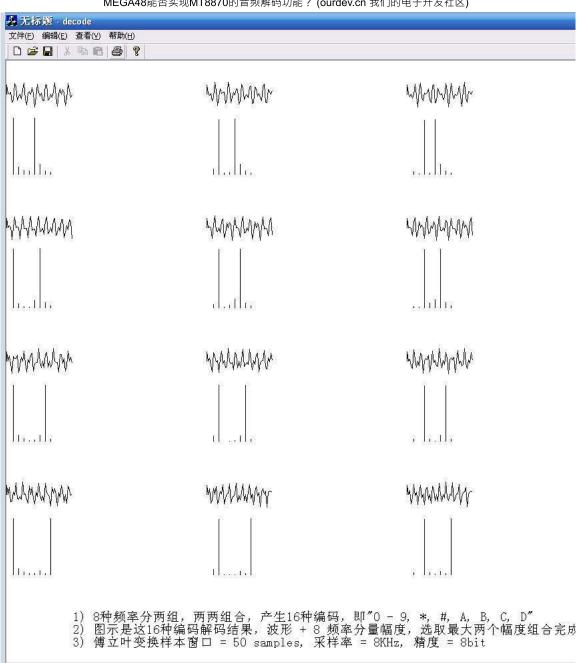
欢迎您·Oliver |登录 |快速注册 |退出 首页 酷贴 搜索 在线 个人功能 🔻 本网站由阿莫电子邮购独家赞助 论坛首页 -> 虚拟总论坛 -> MEGA48能否实现MT8870的音频解码功能? (属于论坛:AVR单片机) 跳转至: --虚拟总论坛 【楼主位】 ldh119 求教MEGA48能否实现MT8870的音频解码功能? 积分:25 派别: 等级:-----来自: 2008-10-13,12:55:32 资料 邮件 本主题已经审核通过 编辑 删除 【1楼】 ldh119 看来大家都不懂呀 积分:25 派别: 等级: -----来自: 2008-10-14,12:20:48 资料 邮件 本回复已经审核通过 编辑 删除 【2楼】 Oliver 可以的,有两种DTMF软解码法. 积分:1441 1.相位比较. 派别: 2.AD采样. 等级:-----来自: 我以前试过第二种,M8跑16MHZ,能解出来,但不是非常灵(可能和当时M8同时要跑别的功能有关) 2051都可以解不要说M48了,可以GOOGLE上搜资料看看. 2008-10-14,12:23:53 本回复已经审核通过 编辑 删除 资料 邮件 【3楼】 hendry 可以解码的·方法如下: 积分:116 派别: 波形-->A/D采样-->8频率点傅立叶变换-->选取最大两个频率分量组合编码-->查表,得DTMF解码结果 等级: -----来自: 我选取的参数如下: 1) A/D 采样频率8KHz·分辨率 8bit 2) 参与傅立叶变换的样本个数(转换窗口宽度) = 50 samples (采样点)3) 8个频率点:697Hz, 770, 852, 941, 1209, 1336, 1477, 1633Hz 4) 傅立叶变换所需的三角函数运算变成查表·表格大小=50*8=400字节 5) atmega8 @ 8MHz 实测结果: 1) 信号动态适应范围 <= 42dB 2) 频偏容忍度 <= +/- 10% 3) 误码率 < 1%·结合软件判断·误码率将为0 本贴被 hendry 编辑过,最后修改时间: 2008-10-14,13:22:00. 2008-10-14,13:16:34 资料 邮件 本回复已经审核通过 编辑 删除 【4楼】 hendry 积分:116 派别: 等级·____ 来自:





| 023/1/29 上午11:03 | MEGA48能否实现MT8870的音频解码功能? (ourdev.cn 我们原 | 的电于开发社区) |
|----------------------|---|-----------------------------|
| 【7楼】 hendry | 对于楼上疑问解释一下: | |
| 积分:116 派别: | 1)8KHz采样率·50个采样点·花费时间=0.125ms*50=6.25ms | |
| 等级: | 2) 为了保证 50 个点至少采一个完整周期,信号周期必须不大于 6.25ms, 即频率必须不小于 1/6.25ms = 160Hz 3) 低于 160Hz 的信号 · 50 个点都采不到一个完整周期,因此无法保证傅立叶计算的正确性 3) 实际上,在 50 个采样点内,上述 8 个频率信号分别可以采 4.4, 4.8, 5.3, 5.9, 7.6, 8.4, 9.2, 10.2 个周期,因此可以保证傅立叶计算正确性 4) 实际表明,采到 40 个点时,仍可以工作,采到 30 个点时,就不可靠了(无法适应两两频率的任意相位组合),最后选择了 50 点,保证了应用的可靠性 5) 当然,采样点越多,计算就越精确,但是,1是没有必要,因为只需要相对把 8个频率点区分开就可以了,没有必要真正计算其准确频谱,2是单片机资源有限,采样点越多,存储开销和计算开销就越大,最后不能实现。 | |
| 来自: | | |
| | | |
| | | |
| | | |
| | | |
| | 2008-10-15,09:52:21 | 资料 邮件 |
| 【8楼】 hendry | 市面上所售比较便宜的 DTMF 解码芯片·都是软解码·内部就一个小单片机。 | |
| 积分:116 派别: | 实际证明·软解码方法比硬解码(硬件带通滤波)效果更好·适应性更强·成本也更低。 | |
| 等级: | | |
| 来自: | 明白了软解码的原理,就不需要外置这些解码芯片了,从而降低成本,降低功耗,提高集成度和可靠性,采购t | 也简单了。 |
| 2008-10-15,10:06:07 | 资料 邮件 | 本回复已经审核通过 编辑 删除 |
| 【9楼】Oliver | 【8楼】 hendry ,讲得很好,以前把DTMF发射,充放电控制,2272解码全部放到M8里面,整个板子就一CPU. | |
| 积分: 1441 | | |
| 派别: 等级: | 之前的东西很庞大的,外面N多元件 | |
| 来自: | | |
| 2008-10-15,10:26:59 | 资料 邮件 | 本回复已经审核通过 编辑 删除 |
| 【10楼】 chendaon 嘛糖人 | 终于碰到了一个DTMF方面的高手了· | |
| | 请教一下hendry·软解码是否对消测音的适应要强一下·硬解码的时候本地语音一高·就不能解码了·或全是证 | 吴码 |
| 88891 | 本贴被 chendaon 编辑过.最后修改时间:2008-10-15,10:37:35. | |
| 9 g g U z) | | |
| 积分:717 | | |
| 派别: 等级: | | |
| 来自:浙江 | | |
| 2008-10-15,10:34:32 | 资料 邮件 | 本回复已经审核通过 编辑 删除 |
| 【11楼】34480016 晕来晕去 | 好帖好收起来 | |
| | | |
| 积分:980 | | |
| 派别: | 温馨提示:发帖前先把输入的内容复制到剪贴板Contrl+C,上坛有风险,发帖要谨慎! | |
| 等级: 来自: 东莞 | 小广告:出售一批全新台湾原产凌阳COG12864(无字库,有背光)点阵液晶屏自制模块42元,提供51口线模拟方式与Q.com, http://shop34480016.taobao.com/ | 总线方式的电路图和C51源代码.271885846@Q |
| 2008-10-15,11:09:01 | 资料 邮件 | 本回复已经审核通过编辑删除 |
| 【12楼】 hendry | 10楼:是的。 | |
| 积分:116 | // 北侧沙 抗胡加拉丁拉岭山伯迈 - 亚洲女子来源江过山山 - 四本林河社 | |
| 派别: 等级: | 经我测试、软解码抗干扰能力很强,我试过了在通话过程中,用音箱播放 DTMF 双音频录音,音箱距离电话1米开外,音量中等,DTMF 音从听筒透进电话,软解码能正确解出,这期间我一直在说话。 | |
| 来自: | 另外·当 DTMF 波形严重削顶失真(幅值饱和)时·也能正确解出号码·这令我很惊讶(因为我用的 A/D 是 8/ 为软件解码是基于信号的频谱·而不是波形·信号失真后·从频谱上看只是谐波分量增加了·但是基波分量还是解码算法·肯定就不行了! 从这点可以看出·软解码方法的稳定性。 | , |
| | | |

当然了,要全面测试软解码指标,还要用电话测试仪,不过我手头没有,所以也没有做。

2008-10-15,11:11:02

资料 邮件

本回复已经审核通过 编辑 删除

【13楼】 chendaon 嘛糖人

77778 0 8 8 8 9 1 【12楼】 hendry

谢谢了‧看来有必要全面学习软解码了‧

积分:717 派别:

等级:-----来自:浙江

2008-10-15,14:12:41

资料 邮件

本回复已经审核通过 编辑 删除

【14楼】 teddeng

积分: **28** 派别: 等级: -----来自: handry·我曾经对这个问题感兴趣·可惜没有真正动过手·我的一些理解写下。。

首先·8KHZ,50SAMPLE DFT的物理意义是算出输入信号在f(i)=8KHZ/50*i (i=0,1,2,,,49)这50个频率点上的幅度、相位分布。

这一点,不论采用什么技巧、算法,原理都是确定无误的。你的那个160HZ解释似乎没有道理。

就DMTF解码具体而言·相位是不用考虑的。当输入信号中有697HZ信号时·我们需要根据f(4)=640HZ这点的DFT幅度值判断;同样770HZ时·需要计算f(5)=800HZ这点的DFT幅度值来判断。问题来了·当输入852HZ时·显然不能根据f(6)=960HZ这点的DFT幅度值来判断了;因为后面还有个941HZ,也许han dry的算法是综合考虑f(5),f(6)的计算幅度吧。也许是可以的·但规范上的误差范围应该是根本谈不上了·误码率应该很大·当然可以在软件上参考同时计算的4个高频群频率综合判断·但总之理论上是不严密的·变成为解码而解码了。改成100sample要好很多·最少频率分辨率高了1倍·可以分辨差80HZ的相邻频率。但确实·先不考虑表格大小·最要命的可能还是计算出来的幅度值·考虑信号的动态范围·可能是个范围很大的数·计算复杂度大大增加了。

所以,实践中我知道的DTMF软解码一般都是按所谓1bit A/D DFT。。即不管什么信号统统整成方波,即非0即1.再做DFT计算。计算量非常小,可以说任何单 片机都可以实现。有兴趣的可以到网上搜搜,有PIC的例子,有HOLTEK的官方文档。不过按我的理解,用MATLAB模拟研究了很久,总觉不理想。事实上, 普通单片机的软解码主要还是用在最低端的民用产品上,如来电显示电话,误码多点问题也不大。

至于带A/D单片机的"正规"解码算法·源码·TI的MSP430有篇AN说得很详细·但计算量就不是普适的了。。

最后提下,电信的DTMF解码,除了8个基本频率,还要计算它们的谐波幅度,这样才能真正不被话音干扰,道理是话音中谐波丰富,而真正的DTMF,几乎没有 谐波。

2008-10-15,17:35:15

资料 邮件

本回复已经审核通过 编辑 删除

【15楼】 miraclelife 麦诺克科技有限公司



希望各位多多发表意见,最近在搞DTMF的软件码,实在是无从下手啊。

积分:967

派别: 等级:----

来自: 山东 蓬莱

2008-11-03,15:11:19

资料 邮件

本回复已经审核通过 编辑 删除

【16楼】 ldh119

积分:25 派别:

等级:-----

来自:

3楼的大侠真乃高人也· 我搞了好久也没解对DTMF码·

能否放个C源程序呀开源一下嘻嘻谢谢了!

2008-11-25,12:52:49

资料 邮件

本回复已经审核通过 编辑 删除

【17楼】 hendry

积分:116 派别:

等级:-----来自: 根据傅里叶变换定义:

$$F(s) = \int_{-\infty}^{+\infty} f(x)e^{-i2\pi sx} dx$$

(原文件名:1.JPG)

对于DTMF解码・f(x)是时间的实函数·记作f(t)。傅里叶变换要求信号持续的时间从过去无穷远到未来无穷远,实际上采集进来的DTMF信号总是持续一个有限时间。记作td·当t<0或t>td时·f(t)取值0。由于f(t) 既不是偶实函数·也不是虚奇函数·因此傅里叶变换结果一定是复的·即F(s)的虚部一定不等于0·所以必须计算虚部。

为了计算·将变换离散化:

$$F(s) = \sum_{0}^{N} f(n)e^{-i2\pi sn} = \sum_{0}^{N} f(n)\cos 2\pi sn - i\sum_{0}^{N} f(n)\sin 2\pi sn = R + iI$$

(原文件名:2.JPG)

sqrt(R*R + I*I) 是我们想要的计算结果,它表示了原信号中包含的某个频率分量的幅值大小。为了避免让单片机做开方运算,变通一下用 abs(R) + abs(I)。

稍后上参考代码,先过一下能否发图片。

2008-11-25,15:01:55

资料 邮件

本回复已经审核通过 编辑 删除

【18楼】 hendry

积分:116 派别:

等级:-----来自: 下面贴代码·mega8@8MHz通过。

注意:1)采样率 4000 bps (采样率很低·降低对系统要求·解码准确度一点不含糊·(*^_^*) 嘻嘻.....)

2) 采样精度 8bit, signed char, 注意是 "有符号数"啊!! 千万别搞错了, 如果输入无符号数 · 哭去吧..

3) 每输入 50 个采样点就得到一次 DTMF 解码 · 历时 12.5ms, 国标要求 DTMF 码持续时间 >= 40ms, 因此连续解码的话可以得到连续3个相同码值 · 实际上 · 我就是这样提高抗干扰性的 。

4) 属于本人原创·请勿用作商用·否则保留追究权利。

by hendry, 2008/11

//

// DTMF 解码程序

// author: hendry

// date: 2008/11

//

unsigned char s dtmf p;

long s_dtmf_real[8];

 $long \quad s_dtmf_image[8];$

const unsigned char g_sincos_len = 50;

const signed char $g_cos_table[400] = \{(signed char)0x7E_s(signed char)0x3A_s(signed char)0xB_s(signed char)0xB_s(signed char)0xD_s(signed char)0xD_s(signed char)0xD_s(signed char)0xB_s(signed char)0xB_s(signe$

(signed char)0x7C,(signed char)0x4E,(signed char)0xCB,(signed char)0xC1,(signed char)0xC1,(signed char)0x4E,(signed char)0x2P,(signed char)0xAD, (signed char)0x7E,(signed char)0x2D,(signed char)0xA1,(signed char)0x91,(signed char)0x10,(signed char)0x7B,(signed char)0x47,(signed char)0x88,(signed char)0x7B,(signed char)0x47,(signed char)0x8B,(signed char)0x8E,(signed char)0x5D,(signed char)0x5D,(signed char)0x5D,(signed char)0x82,(signed char)0xD5,(signed char)0x6E,(signed char)0x6E,(signed char)0x8E,(signed char)0x8E,(signed char)0x8E,(signed char)0x8E,(signed char)0x8E,(signed char)0x8E,(signed char)0x8E,(signed char)0x9E,(signed cha

(signed char)0x14,(signed char)0x7B,(signed char)0x44,(signed char)0x84,(signed char)0x87,(signed char)0x72,(signed char)0x5A,(signed char)0xCE,(signed char)0x82,(signed char)0xB2,(signed char

(signed char)0x84,(signed char)0x84,(signed char)0xCF,(signed char

(signed char)0x84,(signed char)0x95,(signed char)0x38,(signed char)0x76,(signed char)0x0E,(signed char)0x84,(signed char)0x8B,(signed char)0x7E,(signed char)0x0C,(signed char)0x84,(signed char)0x8D,(signed char)0x8B,(signed char

d char)0x37,(signed char)0x76,(signed char)0xDF,(signed char)0x84,

(signed char)0x7E,(signed char)0xD7,(signed char)0x9C,(signed char)0x69,(signed char)0x20,(signed char)0x82,(signed char)0x31,(signed char)0x5E,(signed char)0x92,(signed char)0x82,(signed char)0x82,(signed char)0x85,(signed char)0x5E,(signed char)0x92,(signed char)0x0E,(signed char)0x7D,(signed char)0xC7,(signed char)0xA1,(signed char)0x0E,(signed char)0x85,(signed char)0xFA,(signed char)0xCF,(signed char)0xCF,(signed char)0xCF,(signed char)0xFA,(signed char

(signed char)0x6A,(signed char)0x9E,(signed char)0xD5,(signed char)0x7E,(signed char)0xDA,(signed char)0x9B,(signed char)0x68,(signed char)0x23,(signed char)0x22,(signed char)0x2E,(signed char)0x2E,(signed char)0x7E,(signed char

(signed char)0x7D,(signed char)0xB3,(signed char)0xD0,(signed char)0x7D,(signed char)0xD2,(signed char)0x7D,(signed char)0xD3, (signed char)0xD2,(signed char)0xD6,(signed cha

(signed char)0xA1,(signed char)0x03,(signed char)0x5A,(signed char)0x52,(signed char)0x05,(signed char)0x9A,(signed char)0x7C,(signed char)0xBD,(signed char)0xDF,(signed char)0x6F,(signed char)0x89,(signed char)0x33,(signed char)0x32,(signed char)0x89,(signed char)0x70,(signed char)0xDE,(signed char)0xBE,(signed char)0x7C,(signed char)0x9A,(signed char)0x10,(signed char)0x51,(signed char)0x82,(signed char)0x5B,(signed char)0x22,(signed char)0xA2,(signed char

(signed char)0x6D,(signed char)0x82,(signed char)0x65,(signed char)0xD4,(signed char)0xE5,(signed char)0xE5,(signed char)0x85,(signed char)0xA1,(signed char)0xA1,(signed char)0xA2,(signed char

 $const signed char g_sin_table[400] = \{(signed char)0x00,(signed char)0x70,(signed char)0x67,(signed char)0xEE,(signed char)0x89,(signed char)0xA5,(signed char)0x24,(signed char)0x7C,(signed char)0xCC,(signed char)0xCE,(signed char)0xC1,(signed char)0x44,(signed char)0x7D,(signed char)0x2F,(signed char)0xAD,(signed char)0x86,(signed char)0xE2,(signed char)0x5F,(signed char)0x75,(signed char)0x0C,(signed char)0x96,(signed char)0x93,(signed char)0x06,(signed char)0x76,(signed char)0x2F,(signed char)0x2F,(signed$

(signed char)0xE8,(signed char)0x63,(signed char)0x73,(signed char)0x06,(signed char)0x93,(signed char)0x96,(signed char)0x06,(signed char)0x75,(signed char)0x5F,(signed char)0xE2,(signed char)0x00,(signed char)0x64,(signed char)0x67,(signed char)0xC5,(signed char)0xE3,(signed char

(signed char)0x84,(signed char)0x66,(signed char)0x6A,(signed char)0x65,(signed char)0xDD,(signed char)0x82,(signed char)0xA4,(signed char)0x86,(signed char)0x86,(signed char)0x86,(signed char)0x86,(signed char)0x86,(signed char)0x86,(signed char)0x86,(signed char)0x86,(signed char)0x86,(signed char)0x95,(signed char)0x95,(signed char)0x96,(signed char)0x36,(signed char)0x36,(signed char)0x86,(signed char

(signed char)0x7E,(signed char)0x15,(signed char)0x8C,(signed char)0x85,(signed char)0x52,(signed char)0x70,(signed char)0x82,(signed char)0x92,(signed char

(signed char)0x65,(signed char)0x8E,(signed char)0x8F,(signed char)0x2D,(signed char)0x7D,(signed char)0xE9,(signed char)0xE9,(signed char)0xE9,(signed char)0xFF,(signed char)0x7D,(signed char)0x7D,(signed char)0x18,(signed char)0x8P,(signed char)0xD2,(signed char)0xD2,(signed char)0xD2,(signed char)0xBF,(signed char)0xBF,(signed char)0xE9,(signed char

(signed char)0x00,(signed char)0x77,(signed char)0xB3,(signed char)0xBA,(signed char)0x7A,(signed char)0xF7,(signed char)0xE,(signed char)0x54,(signed char)0x54,(signed char)0xB3,(signed char)0xB4,(signed char)

(signed char)0x44,(signed char)0x4F,(signed char)0x8A,(signed char)0xFD,(signed char)0x78,(signed char)0xB8,(signed char)0x99,(signed char)0x79,(signed char)0x78,(signed char)0x8B,(signed char)0x90,(signed char)0x6D,(signed char)0x92,(signed char)0x02,(signed char)0x0C,(signed char)0x92,(signed char)0x92,(signed char)0x92,(signed char)0x92,(signed char)0x91,(signed char)0x05,(signed char)0x6A,(signed char)0x06,(signed char)0x08F,(signed char)0x08,(signed char)0x08,(signed char)0x08,(signed char)0x08F,(signed char)0x08,(signed ch

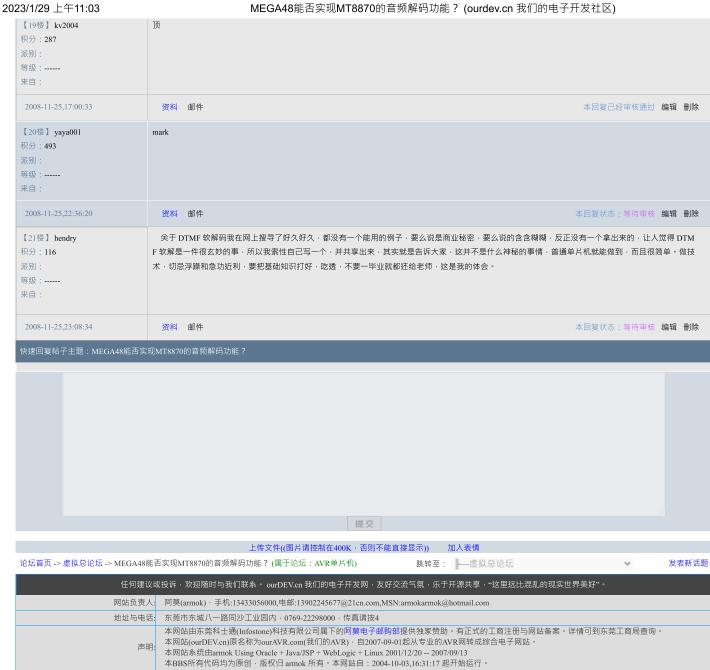
(signed char)0x10,(signed char)0x64,(signed char)0x8B,(signed char)0x11,(signed char)0x63,(signed char)0x8B,(signed char)0x13,(signed char)0x62,(signed char)0x8A,(signed char)0x14,(signed char)0x14,(signed char)0x61,(signed char)0x8A,(signed char)0x16,(signed char)0x69,(signed char)0x18,(signed char)0x18,(signed char)0x5F,(signed char)0x8B,(signed char)0x12,(signed char)0x5E,(signed char)0x7B,(signed char)0x7B,(signed char)0x00,(signed char)0x7D,(signed char)0x12,(signed char)0x12,(signed char)0x6E,(signed char)0x6E,(signed char)0x6E,(signed char)0x7B,(signed char)0x7D,(signed char

(signed char)0x53,(signed char)0x82,(signed char)0x58,(signed char)0x06,(signed char)0x040,(signed char)0x7D,(signed char)0xB6,(signed char)0xE8,(signed char)0x6B,(signed char)0x86,(signed char)0x86,(signed char)0x29,(signed char)0x8D,(signed char)0x74,(signed char)0xD6,(signed char)0xC6,(signed char)0x79,(signed char)0x95,(signed char)0x19,(signed char)0x4A,(signed char)0x83,(signed char)0x61,(signed char)0x7B,(signed char)0xAC,(signed cha

 $(signed\ char)0x3E, (signed\ char)0x07, (signed\ char)0x85, (sig$

gned char)0x82,(signed char)0x6C,(signed char)0xC8,(signed char)0xF1,(signed char)0x50,(signed char)0x88,(signed char)0x79,(signed char)0xAD,(signed char)0x12,(signed char)0x82,(signed char)0x ed~char)0x35, (signed~char)0x95, (signed~char)0x7E, (signed~char)0x97, (signed~char)0x31, (signed~char)0x16, (signed~char)0xAA, (signed~char)0x7A, (signed~char)0x94, (signed~char)0x9 $d \hspace{0.5mm} \hspace{0.5mm} char) 0x 4D, (signed \hspace{0.5mm} char) 0x F5, (signed \hspace{0.5mm} char) 0x C5, (signed \hspace{0.5mm} char) 0x 6E, (signed \hspace{0.5mm} char) 0x 8E, (signed \hspace{0.5mm} char) 0x 6E, ($ $d\; char)0x84, (signed\; char)0x74, (signed\; char)0xB9, (signed\; char)0x03$ **}**; // dtmf decode initiation void init_dtmf_decoder() $s_dtmf_p = 0;$ memset(s_dtmf_real, 0, sizeof(long) * 8); memset(s_dtmf_image, 0, sizeof(long) * 8); // dtmf decode program // input: sample = DTMF wave sample value, // sample rate = 4000 point per second // NOTE!! sample is "SIGNED" char, NOT "UNSIGNED" char !! // output: 0 = decoded not completed yet // '0123456789*#ABCD' = dtmf code unsigned char decode_dtmf(signed char sample) unsigned char i, low_freq, high_freq; unsigned short offset; for (i = 0, offset = s_dtmf_p; i < 8; i ++, offset += g_sincos_len) s_dtmf_real[i] += sample * g_cos_table[offset]; s_dtmf_image[i] += sample * g_sin_table[offset]; s dtmf p++; if $(s_dtmf_p == g_sincos_len) // 50$ sample has inputed, let's look at the result... // calculate |R| + |I|for (i = 0; i < 8; i ++)s_dtmf_real[i] = labs(s_dtmf_real[i]) + labs(s_dtmf_image[i]); // find max value of low freq for (low_freq = 0, i = 1; i < 4; i +++) $if \, (s_dtmf_real[i] > s_dtmf_real[low_freq])$ low freq = i; } // find max value of high freq for $(high_freq = 4, i = 5; i < 8; i ++)$ $if \, (s_dtmf_real[i] > s_dtmf_real[high_freq]) \\$ $high_freq = i;$ } // clean up for continuous decoding $s_dtmf_p = 0;$ $memset(s_dtmf_real,\,0,\,32);$ memset(s_dtmf_image, 0, 32); // combine low and high frequency and decode out dtmf code

```
if ((low_freq == 0) && (high_freq == 4))
                                       return '1';
                                       else if ((low_freq == 0) && (high_freq == 5))
                                       return '2';
                                       }
                                       else if ((low_freq == 0) && (high_freq == 6))
                                       return '3';
                                       else if ((low_freq == 0) && (high_freq == 7))
                                       return 'A';
                                       }
                                       else if ((low_freq == 1) && (high_freq == 4))
                                       return '4';
                                       else if ((low_freq == 1) && (high_freq == 5))
                                       return '5';
                                       else if ((low_freq == 1) && (high_freq == 6))
                                      else if ((low_freq == 1) && (high_freq == 7))
                                       return 'B';
                                       else if ((low_freq == 2) && (high_freq == 4))
                                       return '7';
                                       else if ((low_freq == 2) && (high_freq == 5))
                                       return '8';
                                       else if ((low_freq == 2) && (high_freq == 6))
                                       return '9';
                                       else if ((low_freq == 2) && (high_freq == 7))
                                       return 'C';
                                       }
                                       else if ((low_freq == 3) && (high_freq == 4))
                                       {
                                       return '*';
                                       else if ((low_freq == 3) && (high_freq == 5))
                                       return '0';
                                       else if ((low_freq == 3) && (high_freq == 6))
                                       return '#';
                                      else if ((low_freq == 3) && (high_freq == 7))
                                       return 'D';
                                      }
                                     return 0;
2008-11-25,16:29:39
                                                                                                                                                                  本回复已经审核通过 编辑 删除
                                       资料 邮件
```



本网站已经通过了东莞市公安局安全认证许可,请大家遵守国家法律法规。

法律顾问: 本网站尊重知识产权,自2007年08月15日起,魏龙律师担任ourDEV.cn(原ourAVR.com)常年法律顾问。