

# LD3320 Board 用户手册

## 一. 应用场景主要包括：

1. 电磁炉/微波炉/智能家电操作
2. 导航仪
3. MP3/MP4
4. 数码像框
5. 机顶盒/彩电遥控器
6. 智能玩具/对话玩具
7. PMP/游戏机
8. 自动售货机
9. 地铁自动售票机
10. 导游机
11. 楼宇电视的广告点播
12. 公共照明系统/卫生系统/智能家居的声控

## 二. LD3320 识别芯片介绍

LD3320 是一颗基于非特定人语音识别（SI-ASR: Speaker-Independent Automatic Speech Recognition）技术的语音识别/声控芯片。提供了真正的单芯片语音识别解决方案。

LD3320 芯片上集成了高精度的 A/D 和 D/A 接口，不再需要外接辅助的 Flash 和 RAM，即可以实现语音识别/声控/人机对话功能。并且，识别的关键词语列表是可以动态编辑的。基于 LD3320，可以在任何的电子产品中，甚至包括最简单的 51 作为主控芯片的系统中，轻松实现语音识别/声控/人机对话功能。为所有的电子产品增加 VUI（Voice User Interface）语音用户操作界面。

主要特色功能：

**非特定人语音识别技术：**不需要用户进行录音训练

**可动态编辑的识别关键词语列表：**只需要把识别的关键词语以字符串的形式传送进芯片，即可以在下次识别中立即生效。比如，用户在 51 等 MCU 的编程中，简单地通过设置芯片的寄存器，把诸如“你好”这样的识别关键词的内容动态地传入芯片中，芯片就可以识别这样设定的关键词语了。

**真正单芯片解决方案：**不需要任何外接的辅助 Flash 和 RAM，真正降低系统成本。

**内置高精度 A/D 和 D/A 通道：**不需要外接 AD 芯片，只需要把麦克风接在芯片的 AD 引脚上；可以播放声音文件，并提供 550mW 的内置放大器。

**高准确度和实用的语音识别效果。**

**支持用户自由编辑 50 条关键词语条：**在同一时刻，最多在 50 条关键词语中进行识别，终端用户可以根据场景需要，随时编辑和更新这 50 条关键词语的内容。

### 三. 主要技术参数

1. 内置单声道 mono 16-bit A/D 模数转换
2. 内置双声道 stereo 16-bit D/A 数模转换
3. 内置 20mW 双声道耳机放大器输出
4. 内置 550mW 单声道扬声器放大器输出
5. 支持并行接口或者 SPI 接口
6. 内置锁相电路 PLL，输入主控时钟频率为 2MHz - 34MHz
7. 工作电压：(VDD: for internal core) 3.3V
8. 48pin 的 QFN 7\*7 标准封装
9. 省电模式耗电：1uA

### 四. 模块的使用

在拿到模块后，可将模块直接接入 Open103Z 开发板（当然也很容易可以将测试程序移植到其它开发板）的 SPI 接口，同时接入串口模块用于与 PC 通信。下载“LD3320 测试程序 Open103Z”进行测试。

打开串口，将显示以下内容：



```
开始运行LD3320测试程序
1、流水灯
2、闪烁
3、按键触发
4、全灭
5、状态
```

对着 LD3320 Board 模块上的板载麦克风说出相关的指令，在模块正确识别后，将执行相应的命令。

例如：对着模块说流水灯。模块成功识别后，将在串口打印相应的数据，且开发板上的 LED 以流水灯的方式运行。

```
开始运行LD3320测试程序
1、流水灯
2、闪烁
3、按键触发
4、全灭
5、状态
进入中断12

识别码:1 流水灯 命令识别成功
```

用户只需修改几处代码，便可以增加新的指令。

1.LD2230\_config.h 文件中，增加相应的识别码。（必须）

```
1 #ifndef _LD3320_CONFIG_H_
2 #define _LD3320_CONFIG_H_
3
4 ///识别码（客户修改处）
5 #define CODE_LSD 1 //流水灯
6 #define CODE_SS 2 //闪烁
7 #define CODE_AJCF 3 //按键触发
8 #define CODE_QM 4 //全灭
9 #define CODE_JT 5 //状态
10
```

2.LD2230.c 文件中，在 LD3320\_main 函数中增加相应分支。（必须）

```
45 switch(nAsrRes) //对结果执行相关操作,客户修改
46 {
47 case CODE_LSD: //命令“流水灯”
48 printf("流水灯 指令识别成功\r\n");
49 break;
50 case CODE_SS: //命令“闪烁”
51 printf("闪烁 指令识别成功\r\n");
52 break;
53 case CODE_AJCF: //命令“按键触发”
54 printf("按键触发 指令识别成功\r\n");
55 break;
56 case CODE_QM: //命令“全灭”
57 printf("全灭 指令识别成功\r\n");
58 break;
59 case CODE_JT: //命令“状态”
60 printf("状态 指令识别成功\r\n");
61
62 default:break;
63 }
```

3.LD2230.c 文件中，在 LD\_AsrAddFixed 函数中修改相应代码。（必须）

DATA\_A 为指令的个数，DATA\_B 为指令的最大长度，sRecog 二维数组为相应指令的拼音集合（注意空格）。

pCode 为指令识别码的集合。

```

80 #define DATE_A 5 //数组二维数值
81 #define DATE_B 20 //数组一维数值
82 //添加关键词, 用户修改
83 uint8 sRecog[DATE_A][DATE_B] = {
84     "liu shui deng", \
85     "shan shuo", \
86     "an jian chu fa", \
87     "quan mie", \
88     "zhuang tai" \
89
90 };
91 uint8 pCode[DATE_A] = {
92     CODE_LSD, \
93     CODE_SS, \
94     CODE_AJCF, \
95     CODE_QM, \
96     CODE_JT \
97 }; //添加识别码, 用户修改

```

4.LD2230.c 文件中, 在 LD3320\_main 函数中修改相应代码。(建议)

```

20 printf("1.流水灯\r\n");
21 printf("2.闪烁\r\n");
22 printf("3.按键触发\r\n");
23 printf("4.全灭\r\n");
24 printf("5.状态\r\n");

```

5.LD2230.c 文件中, 在 Board\_text 函数中修改相应代码。(建议)

```

129 switch(Code_Val) //对结果执行相关操作
130 {
131     case CODE_LSD: //命令“流水灯”
132         Glide_LED();
133         break;
134     case CODE_SS: //命令“闪烁”
135         Flicker_LED();
136         break;
137     case CODE_AJCF: //命令“按键触发”
138         Key_LED();
139         break;
140     case CODE_QM: //命令“全灭”
141         Off_LED();
142         break;
143     case CODE_JT: //命令“状态”
144         Jt_LED();
145         break;
146     default:break;
147 }

```

由于在测试程序中没有加入相应的“垃圾关键词语”用以吸收错误识别。所以测试程序比较容易发生误识别。更多的 LD3320 使用方法, 可以参考《语音识别芯片 LD3320 高阶秘籍》