

AN070225

ZY1730 语音模块在公交车报站系统中的应用

Rev 1.0 Date: 2009/05/15

文件信息

类别	内容
关键词	P89LPC922, ZY1730, 公交车报站系统
摘要	本文介绍了 ZY1730 在公交车报站系统中的设计与应用。

技术支持

如果您对文档有所疑问，您可以在办公时间（星期一至星期五上午 8:30~11:50；下午 1:30~5:30；星期六上午 8:30~11:50）拨打技术支持电话或 E-mail 联系。

网 址： www.zlgmcu.com

联系电话： +86 (020) 22644358 22644359 22644360 22644361

E-mail: 80c51mcu@zlgmcu.com

销售与服务网络

广州周立功单片机发展有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4 邮编：510630

电话：(020)38730972 38730976 38730916 38730917 38730977

传真：(020)38730925

网址：<http://www.zlgmcu.com>

广州专卖店

地址：广州市天河区新赛格电子城 203-204 室

电话：(020)87578634 87569917

传真：(020)87578842

南京周立功

地址：南京市珠江路 280 号珠江大厦 2006 室

电话：(025)83613221 83613271 83603500

传真：(025)83613271

北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座 1207-1208 室（中发电子市场斜对面）

电话：(010)62536178 62536179 82628073

传真：(010)82614433

重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦（赛格电子市场）1611 室

电话：(023)68796438 68796439

传真：(023)68796439

杭州周立功

地址：杭州市天目山路 217 号杭州电子科技大楼 502 室

电话：(0571)28139611 28139612 28139613

传真：(0571)28139621

成都周立功

地址：成都市一环路南二段 1 号数码同人港 401 室（磨子桥立交西北角）

电话：(028)85439836 85437446

传真：(028)85437896

深圳周立功

地址：深圳市深南中路 2070 号电子科技大厦 C 座 4 楼 D 室

电话：(0755)83781788（5 线）

传真：(0755)83793285

武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室华中电脑数码市场

电话：(027)87168497 87168297 87168397

传真：(027)87163755

上海周立功

地址：上海市北京东路 668 号科技京城东座 7E 室

电话：(021)53083452 53083453 53083496

传真：(021)53083491

西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室

电话：(029)87881296 83063000 87881295

传真：(029)87880865

目 录

第 1 章 ZY1730 在公交车报站系统中的应用	1
1.1 系统概述.....	1
1.2 ZY1700 语音录放芯片介绍	1
1.2.1 概述.....	1
1.2.2 功能特点.....	2
1.2.3 选型指南.....	2
1.2.4 开发工具.....	3
1.3 SPI协议介绍.....	3
1.3.1 SPI接口综述.....	3
1.3.2 SPI处理格式.....	4
1.3.3 SPI操作时序.....	4
1.4 ZY1730 操作命令与寄存器	5
1.4.1 主要操作命令.....	5
1.4.2 寄存器详解.....	5
1.5 公交车报站系统设计.....	7
1.5.1 硬件设计.....	7
1.5.2 软件设计.....	8
1.6 总结.....	11
A.1 程序清单.....	12
A.2 版本信息.....	19
A.3 版权声明.....	19

第1章 ZY1730 在公交车报站系统中的应用

1.1 系统概述

随着科学技术的发展，智能化的电子产品逐渐取代了重复的人工劳动，而在需要语音提示的场合，如：公交车报站、银行叫号、自动取款机等，见图 1.1，一款可靠的语音芯片可以方便、高效地提供各种提示和警示服务，从而减轻工作人员的劳动强度，提高工作效率。



图 1.1 语音模块的应用

本系统主要介绍了如何利用 ZY1730 语音芯片实现公交车报站，结构框图如图 1.2 所示，系统主要通过按键来控制单片机，当单片机检测到有键按下时，便控制语音芯片执行相应的放音功能。



图 1.2 系统结构框图

1.2 ZY1700 语音录放芯片介绍

1.2.1 概述

广州致远电子有限公司最新推出的语音模块——ZY1700 系列，是 ZY1420 语音芯片的升级产品，它不仅具有 ZY1420 的所有优良性能，如大容量的存储器，消噪的麦克风前置放大器，自动增益调节 AGC 电路，专用语音滤波电路，高稳定性的时钟震荡电路和语音处理电路，而且还提供多项新功能，如多条信息管理，两种操作模式（按键操作模式和 SPI 操作模式），灵活的分段录、放音操作，音量控制以及擦除等功能，并且在音质、录放音时间长度等方面也都有了很大的改善。

ZY1700 系列语音模块采用模拟处理存储方式，音频数据无须数字压缩，直接存储在固体存储器中；这种存储方式基本消除了常见的背景噪音，从而可以提供更优质的语音；同时，语音内容即使在电路断电后也不会丢失，从而保证了数据的完整和系统的稳定。

一个最小的 ZY1700 系列录放系统仅由一个麦克风、一个喇叭、两个按钮、一个电源就

可以组成，使用非常方便，管脚见图 1.3。

1	/REC	/SS	24
2	/FT	SCLK	23
3	/PLAY	MOSI	22
4	Vss	MISO	21
5	SP+	/RESET	20
6	SP-	/FWD	19
7	Vcc	/EREASE	18
8	MIC-	/VOL	17
9	MIC+	nINT/RDY	16
10	AnaIn	AUD/AUX	15
11	/LED	NC	14
12	NC	NC	13

图 1.3 语音芯片管脚图

1.2.2 功能特点

- 可擦除/录音 10 万次，存储内容断电可以保留 100 年；
- 两种操作模式（按键操作和 SPI 操作）；
- 两种录音输入方式（AGC/Anain）和两种放音输出方式（AUD/AUX）；
- PWM 级别 D 扬声器放大器、可直接驱动一个 8Ω 的扬声器或典型的蜂音器；
- 录音时间 30S（ZY1730）、120S（ZY120）、240S（ZY17240）；
- 灵活的分段录、放音操作（最少段操作时间为 156ms）；
- 有丰富多样的工作状态提示；
- 音质好，性能高，物美价廉；
- 体积小，使用方便，应用灵活；
- 宽工作电压范围：2.4V~5.5V，极限电压 6V；
- 静态电流：0.5 μA~1 μA；
- 工作电流：20mA。

1.2.3 选型指南

ZY1700 系列语音芯片现有三款，选型指南见表 1.1。

表 1.1 选型指南

型号	电压范围	工作电流	静态电流	录音时间	采用频率	放音方式	录音方式
ZY1730	2.4~5.5V	20mA	0.5~1 μA	30S	6.4kHz	AUD/AUX	AGC/Anain
ZY17120	2.4~5.5V	20mA	0.5~1 μA	120S	6.4kHz	AUD/AUX	AGC/Anain
ZY17240	2.4~5.5V	20mA	0.5~1 μA	240S	6.4kHz	AUD/AUX	AGC/Anain

1.2.4 开发工具

ZY VoiceProgrammer 是广州致远电子有限公司开发的专用语音下载器，通过配套软件可以将多个音频文件（wav 文件）下载到 ZY 系列语音芯片中，并将指定的文件下载到指定的存储位置上，使得 ZY 系列语音模块的应用变得非常方便，实物如图 1.4 所示。

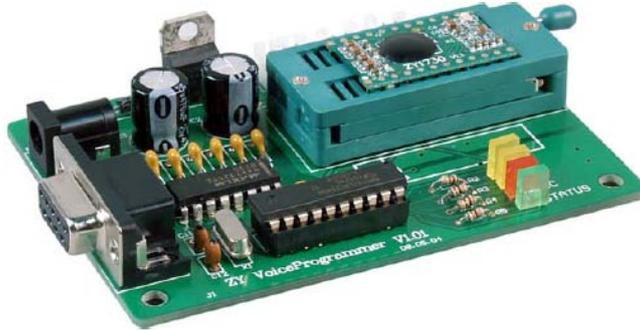


图 1.4 ZY1730 语音编程器

1.3 SPI协议介绍

系统使用 SPI 通信协议完成单片机与语音芯片之间的通信，从而实现报站功能。SPI 是一个同步串行接口，需要四根信号线（SCLK, MOSI, MISO, /SS）与 ZY1730 相连。ZY1730 作为一个外设从机，几乎所有的操作都可以通过 SPI 接口实现。

1.3.1 SPI接口综述

采用 SPI 模式对 ZY1730 语音芯片的操作应遵循以下协议。

- (1) 一个 SPI 处理开始于 /SS 管脚的下降沿；
- (2) 在一个完整的 SPI 指令传输周期，/SS 管脚必须保持低电平；
- (3) 数据在 SCLK 上升沿被锁存在芯片的 MOSI 管脚，在 SCLK 的下降沿从 MISO 管脚输出，并且首先移出低位；
- (4) SPI 指令操作码包括命令字节，数据字节和地址字节，决定了 ZY1730 的指令类型；
- (5) 当命令字节及地址数据输入到 MOSI 管脚时，同时状态寄存器中的内容从 MISO 管脚移出；
- (6) 一个 SPI 指令传输完成时，/SS 管脚变为高电平；
- (7) 在完成一个 SPI 命令的操作后，会启动一个中断信息，并且中断位持续保持为低，直到芯片收到 CLR_INT 命令或者芯片复位。

1.3.2 SPI处理格式

为确保数据可以被准确接收，使用 SPI 通信方式向语音芯片发送命令和数据时，需要按照一定的命令处理格式，具体处理格式如表 1.2。

表 1.2 SPI 处理的格式

	第一字节	第二字节	第三字节	第四字节	第五字节	第六字节	第七字节
MOSI	命令	数据 1	数据 2 或开始地址 1	数据 3 或开始地址 2	结束地址 1	结束地址 2	结束地址 3
MISO	SR0 低字节	SR0 高字节	数据 1 或 SR0 低字节	数据 2 或 SR0 高字节	SR0 低字节	SR0 高字节	SR0 低字节

MOSI 是 SPI 接口的“主机输出从机接收端”。数据在 SCLK 的上升沿锁存进芯片，并且低位首先移出。ZY1730 的 SPI 指令格式依赖于命令类型，根据不同类型的命令，指令可能是两个字节（如：上电命令、复位命令、清中断命令等）或 7 个字节（如：分段录音命令、分段放音命令等）。

MISO 即“主机接收从机发送端”，数据在 SCLK 的下降沿从 MISO 管脚输出，并且低位首先移出。伴随着指令码的输入，MISO 会在前两个字节返回芯片的当前状态和行地址信息<A10:A0>。

1.3.3 SPI操作时序

语音芯片作为一个 SPI 从器件，它有自己特定的 SPI 通信方式。在时钟信号的下降沿，语音芯片将自己目前的状态信息返回给单片机，而在时钟信号的上升沿，单片机将命令发送给语音芯片，具体时序见图 1.5。

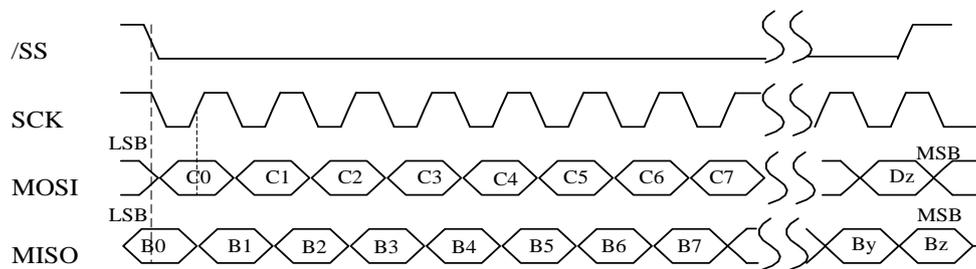


图 1.5 SPI 操作时序

1.4 ZY1730 操作命令与寄存器

1.4.1 主要操作命令

在系统设计中主要用到了以下指令实现单片机和语音芯片之间的通信，具体内容见表 1.3。

表 1.3 主要命令操作

命令	操作码	描述
PU	0x01,0x00	芯片上电
STOP	0x02,0x00	停止当前操作
RESET	0x03,0x00	器件复位
CLR_INT	0x04,0x00	读当前状态并清 INT 位和 EOM 位
RD_STATUS	0x05,0x00,0x00	读状态
PD	0x07,0x00	器件掉电并进入待机模式
PLAY	0x40,0x00	器件从当前播放指针开始播放，直到它到达 EOM 或者收到 STOP 命令
REC	0x41,0x00	器件从当前录音指针开始录音
ERASE	0x42,0x00	器件从当前播放指针开始擦除，当到达 EOM 时停止
WR_APC2	0x65,<D7:D0>,<xxxxxD11:D8>	装载 APC 寄存器，音量由<D2:D0>确定
SET_PLAY	0x80,0x00,<S7:S0>,<00000S10:S8>,<E7:E0>,<00000E10:E8>,0x00	从起始地址开始放音到结束地址停止
SET_REC	0x81,0x00,<S7:S0>,<00000S10:S8>,<E7:E0>,<00000E10:E8>,0x00	从起始地址开始录音到结束地址停止
SET_ERASE	0x82,0x00,<S7:S0>,<00000S10:S8>,<E7:E0>,<00000E10:E8>,0x00	从起始地址开始擦除到结束地址停止

注：<S7:S0>为起始地址低 8 位，<S10:S8>为起始地址高 3 位。<E7:E0>为结束地址低 8 位，<E10:E8>为结束地址高 3 位。

1.4.2 寄存器详解

ZY1730 中的状态寄存器可以返回器件的内部状态，通过状态寄存器的内容，可以知道上一条命令是否发送成功、芯片是否准备好、当前操作是否完成等重要信息。以下是对寄存器 SR0、SR1、APC 的位描述。

1. 状态寄存器（SR0）

SR0 是一个 16 位的状态寄存器，由 MISO 返回。它包括 5 个状态位（D4:D0）以及 11 个地址位（A10:A0），见表 1.4。

表 1.4 SR0 的位定义

SR0		位数：16 位			类型：读				
第一字节	位序列	D7	D6	D5	D4	D3	D2	D1	D0
	名称	A2	A1	A0	INT	EOM	PU	FULL	CMD_ERR
第二字节	位序列	D15	D14	D13	D12	D11	D10	D9	D8
	名称	A10	A9	A8	A7	A6	A5	A4	A3
描述:		器件状态寄存器							
访问:		在 MISO 端每个 SPI 命令的开始两个字节返回 SR0							

在 MISO 输出信息中，SR0 第一个字节的状态位提供了重要信息，该信息表明了上一个 SPI 命令发送后的结果。下面详细地介绍了 SR0 寄存器每一位的作用，具体内容见表 1.5。

表 1.5 SR0 的位描述

SR0			
	位	名称	描述
第一字节	7	A2	当前行地址位 2
	6	A1	当前行地址位 1
	5	A0	当前行地址位 0
	4	INT	当前操作完成时该位置 1，可被 CLR_INT 清除
	3	EOM	当检测到 EOM 时该位置 1，可被 CLR_INT 清除
	2	PU	当器件工作在 SPI 模式且上电时，该位置 1
	1	FULL	当该位为 1 时，说明存储器已满，这意味着不能再录音，除非擦除旧信息。该位只有在按键录音和擦除时才有效
	0	CMD_ERR	该位置 1 时说明前一条 SPI 指令无效，如果微控制器发送了少于 5 个字节的行地址，SPI 指令将会被解码，而不是被忽略
第二字节	15~8	A15~A8	当前行地址 15~8 位

2. 状态寄存器 (SR1)

SR1 是一个 8 位的状态寄存器，它反映了语音芯片的当前状态。具体内容由 MISO 返回，见表 1.6。

表 1.6 SR1 的位定义

SR1		位数：8 位				类型：读		
位序列	D7	D6	D5	D4	D3	D2	D1	D0
名称	SE4	SE3	SE2	SE1	REC	PLAY	ERASE	CMD_ERR
描述:	器件状态寄存器 1							
访问:	RD—STATUS 命令，<D7~D0>是 MISO 的第三字节							

在 MOSI 输出信息中，通过 SR1 状态寄存器的内容，可以判断当前操作是否完成。下面详细地介绍了 SR1 寄存器每一位的作用，具体内容见表 1.7。

表 1.7 SR1 的位描述

SR1		
位	名称	描述
7	SE1	音效 1 录音时该位为 1，当擦除时为 0
6	SE2	音效 2 录音时该位为 1，当擦除时为 0
5	SE3	音效 3 录音时该位为 1，当擦除时为 0
4	SE4	音效 4 录音时该位为 1，当擦除时为 0
3	REC	该位为 1 时表示当前操作正在录音
2	PLAY	该位为 1 时表示当前操作正在放音
1	ERASE	该位为 1 时表示当前操作正在擦除
0	RDY	在按键模式下，RDY=1 说明器件准备好接收命令 在 SPI 模式下，如果 RDY=1 说明器件准备接收新命令，例如，REC、PLAY、ERASE。如果 RDY=0，说明器件正忙不接收新命令，除了 RESET、CLR_INT、RD_STATUS、PD。但是录音、播放执行时将会接收 STOP 命令。如果其它命令被发送，将会被忽略，且置 CMD_ERR 为 1 对分段控制命令，RDY=1 表示缓冲器为空，SPI 可以接收同类型的分段控制命令，如果主机发送其它命令，将被忽略且 COM_ERR 为 1，除非新命令是 RESET、CLR_INT、RD_STATUS 和 PD。同样，在 SET_PLAY 和 SET_REC 时，将会接收 STOP 命令

3. APC寄存器

APC 寄存器可以将模拟信号通道配置为多种信号通道，包括：录音信号源、输入信号的混音、输入到输出信号的播放混合及直通信号到输出信号。

APC 寄存器的描述见表 1.8。

表 1.8 APC 寄存器的描述

APC	位数：12 位	类型：读/写
位序列：	<D11~D0>（见下表）	
描述：	模拟路径配置寄存器	
访问：	读：RD_APC；写：LD_APC	

1.5 公交车报站系统设计

1.5.1 硬件设计

报站系统以 NXP 公司的 P89LPC922 单片机作为主控芯片，语音模块的设计采用广州致远电子有限公司的 ZY1730 语音芯片，电路原理如图 1.6。

单片机的 P0¹，P0²，P0³，P0⁴ 口和语音芯片的 SPI 接口相连。按键与单片机的 P1⁴ 口相接，主控芯片一旦检测到有键按下，则执行相应的报站命令。

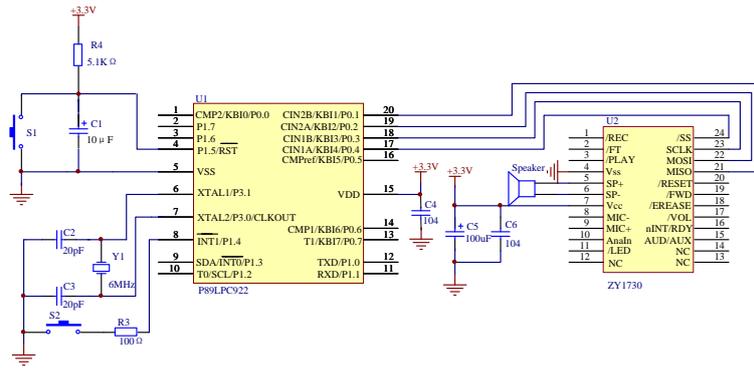


图 1.6 系统原理图

1.5.2 软件设计

1. 指令范例

(1) 上电命令

在语音模块播放报站信息之前，应首先发送上电命令唤醒 ZY1730，使它进入 SPI 空闲状态。图 1.7 是使用逻辑分析仪抓取的上电命令时序图，从图中可以很清楚地看出数据的收发过程。

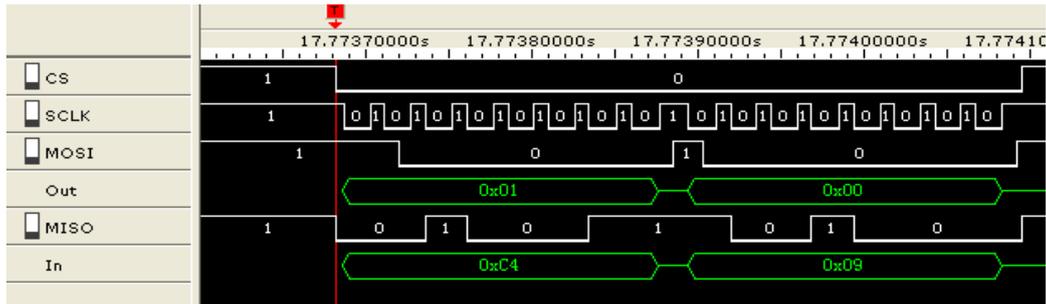


图 1.7 上电命令波形图

给语音芯片发送上电命令的程序如下。

程序清单 1.1

```

/*****
** 函数名称: RW_Byte
** 函数功能: 对 ZY1730 芯片的 SPI 口进行读写一个字节操作
** 入口参数: com 向语音芯片发送的数据
** 出口参数: revdat 语音芯片向主机返回的数据
*****/
uchar RW_Byte(uchar com)
{
    uchar i;
    uchar revdat = 0, senddat = 0;
    for(i = 0; i < 8; i++)
    {
        SCLK = 0;
        /* 在时钟信号上升沿将数据发送到 */

```

```

                                                                    /* 总线上 */
                                                                    */
Delay_us(8);
senddat = com & 0x01;
                                                                    /* 首先发送最低位 */
                                                                    */
com >>= 1;
if(senddat == 0x01)
{
    MOSI = 1;
}
else
{
    MOSI = 0;
}
if(MISO)
{
    revdat = revdat | (0x01 << i);
                                                                    /* 从最低位开始接收数据 */
                                                                    */
}
SCLK = 1;
Delay_us(8);
}
MOSI = 1;
return(revdat);
}

/*****
** 函数名称: ZY1730_PU
** 函数功能: 对 ZY1730 芯片进行上电操作
** 入口参数: 无
** 出口参数: 无
*****/

void PU()
{
    uchar temp;
    do{
        SS = 0;
                                                                    /* SPI 启动 */
                                                                    */
        RW_Byte(0X01);
        RW_Byte(0x00);
        SS = 1;

        SS = 0;
        temp = RW_Byte(0x15);
                                                                    /* 读状态寄存器 */
                                                                    */
        RW_Byte(0x00);
        RW_Byte(0x00);
        SS = 1;
    }while(temp & 0x01);
                                                                    /* 判断命令是否被成功接收 */
                                                                    */
}

```

(2) 分段放音命令

本系统中,每站的站名存储在不同的地址中,报站时需要对相应地址中的语音进行读取。分段放音命令允许 SPI 主机通过指定起始和结束地址,随机地访问存储器中的任意位置,芯片从起始地址开始播放操作,到结束地址停止,时序见图 1.8。

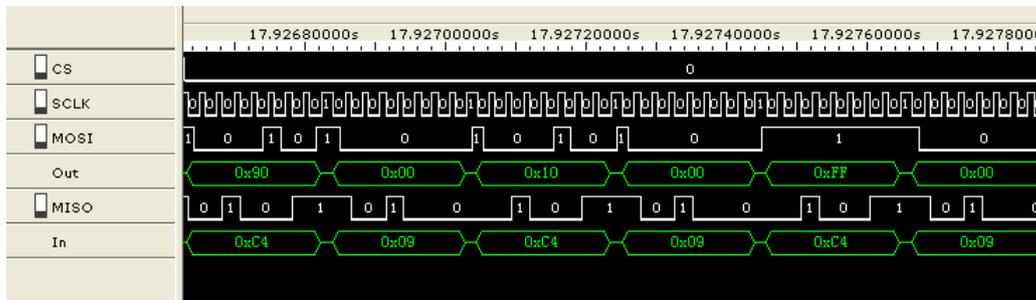


图 1.8 分段放音命令波形图

另外,设计中还用到了清中断命令、读状态寄存器命令、写配置寄存器命令、掉电命令等。其命令的发送过程与上电命令相同。

2. 放音流程

ZY1730 语音芯片可以同时接收两条分段放音命令,当第一段语音播放到结束地址时不会停止,而是自动地跳到第二段语音的开始地址继续播放操作。连续使用两条分段放音命令可以缩小两段录音信息之间的空白时间,使两段独立的录音信息平滑连接。在本系统中,每次报站可以分为三段:【xx 站】、【到了】、【下车的乘客请您从后门下车】。因此进行三段放音设计,程序流程图见图 1.9。

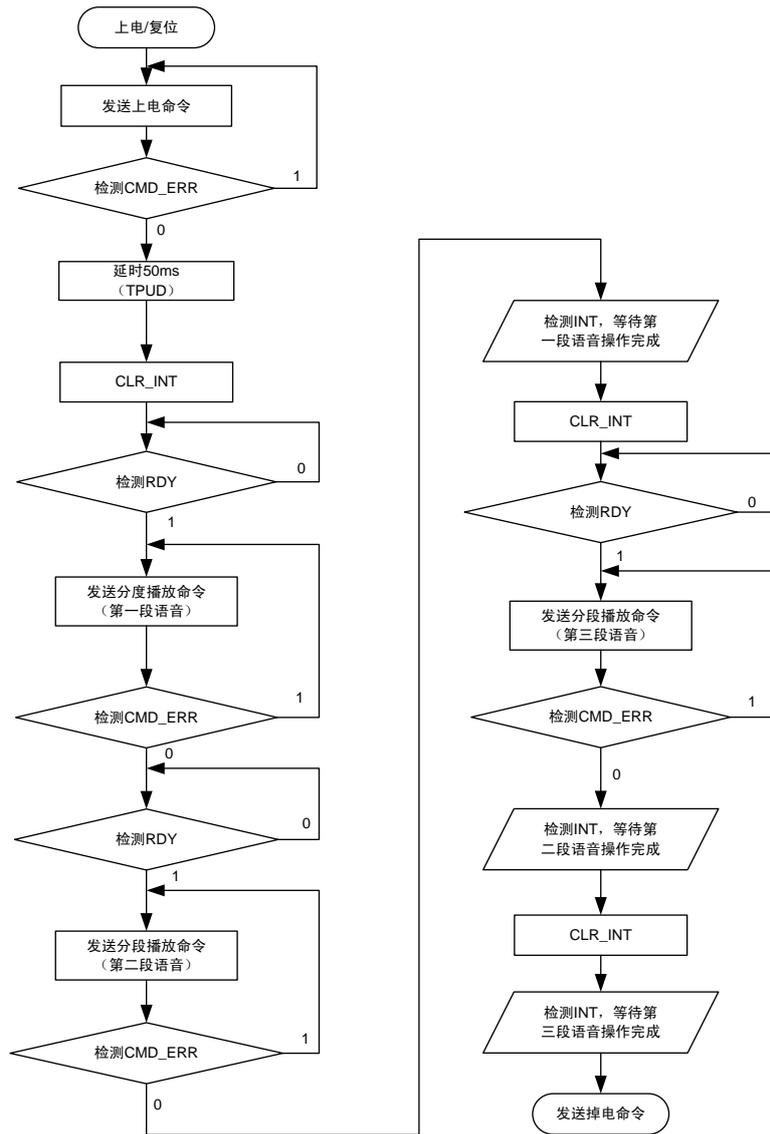


图 1.9 三段放音流程图

3. 系统程序清单

系统通过模拟 SPI 通信协议实现单片机与语音芯片之间的通信，灵活的对语音芯片进行控制，实现公交车报站。程序清单见文档附件。

1.6 总结

本系统使用 ZY1730 语音芯片，较好的实现了报站功能。该语音芯片性能稳定可靠，没有常见的背景噪音，且电路断电后语音内容不会丢失，满足了系统的设计要求。目前本系统使用手动按键的报站方式，将来会向智能化发展，即采用无线通信技术实现。而无论使用哪种报站器，ZY1730 语音芯片均可满足要求。

A.1 程序清单

```
/******  
** 公 司： 广州致远电子有限公司  
** 单片机： P89LPC922  
** 晶 振： 6MHz  
** 功 能： 使用 ZY1730 语音芯片实现公交车报站  
*****/  
  
#include <reg922.h>  
#define uchar unsigned char  
#define uint unsigned int  
  
//管脚定义  
  
sbit SS      = P0^4;  
sbit SCLK    = P0^3;  
sbit MOSI    = P0^2;  
sbit MISO    = P0^1;  
sbit KEY1    = P1^4;  
  
//全局变量的声明  
  
char  number;  
uchar num[2];  
uchar addr[28] = {0x00,0x00,0x25,0x29,0x6A,0x71,0x2A,  
                  0x2D,0x2E,0x31,0x32,0x36,0x37,0x3E,  
                  0x3F,0x43,0x44,0x49,0x4A,0x4F,0x50,  
                  0x53,0x54,0x5A,0x5B,0x61,0x62,0x69};  
/******  
** 函数名称： delay_us  
** 函数功能： 延时，运行于 LPC900 单片机 6MHz 大约为 u*1uS  
** 入口参数： u 延时时间位 u 微秒  
** 出口参数： 无  
*****/  
void Delay_us(uchar u)  
{  
    uchar i;  
    for(i = u;i > 0;i--);  
}  
  
/******  
** 函数名称： Delay_5ms  
** 函数功能： 延时，运行于 LPC900 单片机 6MHz 大约为 5mS 延时函数  
** 入口参数： 无  
*****
```

```

** 出口参数: 无
**/
void Delay_5ms(void)
{
    uint j;
    for(j = 2000;j > 0;j--);
}

**/
** 函数名称: delay_50ms
** 函数功能: 延时,运行于 LPC900 单片机 6MHz 大约 50mS
** 入口参数: 无
** 出口参数: 无
**/
void Delay_50ms()
{
    uint j;
    for(j = 20000;j > 0;j--);
}

**/
** 函数名称: RW_Byte
** 函数功能: 对 ZY1730 芯片的 SPI 口进行读写一个字节操作
** 入口参数: com    向语音芯片发送的数据
** 出口参数: revdat 语音芯片向主机返回的数据
**/
uchar RW_Byte(uchar com)
{
    uchar i;
    uchar revdat = 0,senddat = 0;
    for(i = 0;i < 8;i++)
    {
        SCLK = 0;                                /* 在时钟信号上升沿将数据发送到 */
                                                /* 总线上                          */
        Delay_us(8);
        senddat = com & 0x01;                       /* 首先发送最低位                  */
        com >>= 1;
        if(senddat == 0x01)
        {
            MOSI = 1;
        }
        else
        {
            MOSI = 0;
        }
    }
}

```

```

    }
    if(MISO)
    {
        revdat = revdat | (0x01 << i);           /* 从最低位开始接收数据 */
    }
    SCLK = 1;
    Delay_us(8);
}
MOSI = 1;
return(revdat);
}

/*****
** 函数名称: Two_Byte_Com
** 函数功能: 对 ZY1730 芯片的 SPI 口进行读写两个字节操作
** 入口参数: com 向语音芯片发送的数据
** 出口参数: 无
*****/

void Two_Byte_Com(uchar com)
{
    uchar temp;
    do{
        SS = 0;                               /* SPI 启动 */
        RW_Byte(com);
        RW_Byte(0x00);
        SS = 1;

        SS = 0;
        temp=RW_Byte(0x15);                    /* 读状态寄存器 */
        RW_Byte(0x00);
        RW_Byte(0x00);
        SS = 1;
    }while(temp & 0x01);                       /* 判断命令是否被成功接收 */
}

/*****
** 函数名称: ZY1730_PU
** 函数功能: 对 ZY1730 芯片进行上电操作
** 入口参数: 无
** 出口参数: 无
*****/

void ZY1730_PU(void)
{
    Two_Byte_Com(0x01);
}

```

```
}

/*****
** 函数名称: ZY1730_PU
** 函数功能: 对 ZY1730 芯片进行掉电操作
** 入口参数: 无
** 出口参数: 无
*****/

void ZY1730_PD(void)
{
    Two_Byte_Com(0x07);
}

/*****
** 函数名称: ZY1730_CLR_INT
** 函数功能: 对 ZY1730 芯片发送清中断命令
** 入口参数: 无
** 出口参数: 无
*****/

void ZY1730_CLR_INT(void)
{
    Two_Byte_Com(0x04);
}

/*****
** 函数名称: Wait_RDY
** 函数功能: 等待芯片准备接收下一条命令
** 入口参数: 无
** 出口参数: 无
*****/

void Wait_RDY(void)
{
    uchar temp;
    do{
        SS = 0;
        RW_Byte(0x15);          /* 读状态寄存器 */
        RW_Byte(0x00);
        temp = RW_Byte(0x00);
        SS = 1;
    }while(!(temp & 0x01));    /* 判断芯片是否准备好 */
}

/*****
** 函数名称: Wait_Finish
```

```

** 函数功能：等待当前操作完成
** 入口参数：无
** 出口参数：无
*****/
void Wait_Finish(void)
{
    uchar temp;
    do{
        SS = 0;
        temp = RW_Byte(0x15);          /* 读状态寄存器          */
        RW_Byte(0x00);
        RW_Byte(0x00);
        SS = 1;
    }while(!(temp & 0x10));          /* 判断是否产生中断      */
    ZY1730_CLR_INT();              /* 发送清中断命令        */
    Wait_RDY();                    /* 等待芯片准备好接收命令 */
}

*****/
** 函数名称：ZY1730_Set_PLAY
** 函数功能：对 ZY1730 芯片写分段放音命令
** 入口参数：s_addr 起始地址
               e_addr 结束地址
** 出口参数：无
*****/
void ZY1730_Set_PLAY(uchar s_addr,uchar e_addr)
{
    uchar temp;
    do{
        SS = 0;
        RW_Byte(0x90);              /* 分段播放命令          */
        RW_Byte(0x00);
        RW_Byte(s_addr);            /* 发送起始地址          */
        RW_Byte(0x00);
        RW_Byte(e_addr);           /* 发送结束地址          */
        RW_Byte(0x00);
        RW_Byte(0x00);
        SS=1;

        SS=0;
        temp=RW_Byte(0x15);         /* 读状态寄存器          */
        RW_Byte(0x00);
        RW_Byte(0x00);
        SS = 1;
    }
}

```

```

    }while(temp & 0x01);
}

/*****
** 函数名称: ZY1730_Wr_APC
** 函数功能: 对 ZY1730 芯片的 APC 寄存器进行写操作
** 入口参数: 无
** 出口参数: 无
*****/
void ZY1730_Wr_APC(void)
{
    uchar temp;
    do{
        SS = 0;
        RW_Byte(0x65);          /* 写 APC 寄存器命令          */
        RW_Byte(0x40);          /* 写 APC 寄存器 D0~D7 为 0x40 */
        RW_Byte(0x04);          /* 写 APC 寄存器 D8~D11 为 0x04 */
        SS=1;

        SS=0;
        temp=RW_Byte(0x15);      /* 读状态寄存器              */
        RW_Byte(0x00);
        RW_Byte(0x00);
        SS = 1;
    }while(temp & 0x01);
}

/*****
** 函数名称: Play_Three
** 函数功能: 控制 Y1730 语音芯片连续播放三段语音
** 入口参数: address[28] 存放语音的起始和终止地址
** 出口参数: 无
*****/
void Play_Three(uchar address[28])
{
    ZY1730_PU();                /* 发送上电命令              */
    Delay_50ms();
    ZY1730_CLR_INT();           /* 发送清中断命令            */
    Wait_RDY();

    ZY1730_Wr_APC();            /* 设置 APC 寄存器            */
    Wait_RDY();                 /* 等待芯片准备好接收下一条命令 */
    ZY1730_Set_PLAY(address[2 * number],address[2 * number + 1]);
    Wait_RDY();                 /* 等待芯片准备好            */
}

```

```

    Wait_Finish();                                /* 等待第一段语音放音完成 */

    ZY1730_Set_PLAY(0x10,0x12);                  /* 发送分段放音命令 */
    Wait_Finish();                                /* 等待第三段语音放音完成 */

    ZY1730_Set_PLAY(0x13,0x24);
    Wait_Finish();                                /* 等待第二段语音放音完成 */

    ZY1730_PD();
}

/*****
** 函数名称: main
** 函数功能: 系统主函数
** 入口参数: 无
** 出口参数: 无
*****/
void main(void)
{
    number = 0;
    P0M1 = 0x00;                                  /* IO 口初始化 */
    P0M2 = 0x00;
    P1M1 = 0x00;
    P1M2 = 0x00;
    while(1)
    {
        if(KEY1 == 0)
        {
            Delay_5ms();
            if(KEY1 == 0)
            {
                Play_Three(addr);                /* 执行放音功能 */
                while(!KEY1);
            }
        }
    }
}

/*****
END FILE
*****/

```

A.2 版本信息

修订版本	修订日期	描述
Rev 1.0	2009 年 05 月 15 日	原始版本

A.3 版权声明

广州周立功单片机发展有限公司随附提供的软件或文档资料旨在提供给您（本公司的客户）使用，仅限于且只能在本公司制造或销售的产品上使用。

该软件或文档资料为本公司和/或其供应商所有，并受适用的版权法保护。版权所有。如有违反，将面临相关适用法律的刑事制裁，并承担违背此许可的条款和条件的民事责任。

本公司保留在不通知读者的情况下，修改文档或软件相关内容的权利，对于使用中所出现的任何效果，本公司不承担任何责任。

该软件或文档资料“按现状”提供。不提供保证，无论是明示的、暗示的还是法定的保证。这些保证包括（但不限于）对出于某一特定目的应用此软件的适销性和适用性默示的保证。在任何情况下，公司不会对任何原因造成的特别的、偶然的或间接的损害负责。