

**通信系统开发综合实训报告**

**学 院： 信息与控制学院**

**专业班级： 通信工程2班**

**姓名学号：**

**指导老师： 贾婷 、安璐**

**成 绩：**

**2021年7月9日**

摘 要

本次实训通过对单片机上位机和下位机串行通信设计的分析与研究，进行本次实训设计。本实训分析了测控系统的结构和特点,结合现代控制技术、微机技术以及单片机STM32处理器为代表的微电子技术设计了智能时钟系统,阐述了系统设计要求、设计方案和实现方法。考虑到串口通信精度高、操作简单等特点，所以采用有线串口通信为执行器。并且加入了人机交互功能,使得系统操作更具有人性化。

系统是以单片机STM32芯片为最小系统作为控制系统,系统设计先是下位机按键控制，蜂鸣器响及播放音乐及音乐切换，单片机将必理完成的数据通过串口通信模块传输到PC机上，此时Python上位机界面上将会实时的显示当前数据，达到本次实训设计要求。通过最后的操作表明,系统完全达到了设计要求。

系统的软硬件设计采用了模块化的设计思想。硬件方面详细介绍了各个电路功能模块及系统扩展所需要的其他一些外围电路的设计,软件方面以硬件电路为基础分别进行了模块化设计。为使系统控制更准确、操作更简化,在数据传输的过程中采用了有线串口通信的数据传输方法。同时设计中考虑到可能存在的各种温度干扰,采用了软硬件结合的控制技术,提高系统的稳定性。最后对系统进行了模拟性能测试,测试结果表明,系统性能良好。系统的个别环节还有待进一步优化和完善。

关键词：STM32最小系统；Python上位机；串口通信

**Abstract**

In this paper, through the analysis and research of the serial communication design between the upper computer and the lower computer of the single chip microcomputer, the training design is carried out. This design analyzes the structure and characteristics of the measurement and control system, designs the () system combined with modern control technology, microcomputer technology and micro electronic technology represented by STM32 processor, and expounds the design requirements, design scheme and implementation method of the system. Considering the high precision and simple operation of serial communication, the wired serial communication is used as the actuator. And the human-computer interaction function is added to make the system operation more humanized.

The system is based on STM32 chip as the minimum system as the control system, the system design is the lower computer key control, buzzer sound and play music and music switching, MCU will complete the data through the serial communication module to the PC, at this time, python host computer interface will display the current data in real time, to meet the training design requirements. The final operation shows that the system fully meets the design requirements.

The software and hardware design of the system adopts the idea of modularization. In terms of hardware, the design of each circuit function module and some other peripheral circuits needed for system expansion are introduced in detail. In terms of software, the modular design is carried out based on the hardware circuit. In order to make the system control more accurate and the operation more simplified, the data transmission method of wired serial communication is adopted in the process of data transmission.

**Keywords:** STM32 minimum system; Python host computer; Serial communication

目 录

**[摘 要](#_Toc17481)** [I](#_Toc17481)

**[绪 论](#_Toc11845)** [1](#_Toc11845)

**[1 硬件部分设计](#_Toc1464)** [2](#_Toc1464)

[1.1硬件结构设计 2](#_Toc7641)

[1.2硬件模块选择 2](#_Toc7677)

[1.2.1蜂鸣器模块设计 2](#_Toc6186)

[1.2.2单片机模块设计 3](#_Toc32068)

[1.2.3 LCD显示模块 4](#_Toc18626)

[1.2.4 NRF24L01无线模块 6](#_Toc21720)

[1.2.4 TTP226八路触摸模块 7](#_Toc2071)

**[2 软件部分设计](#_Toc15940)** [8](#_Toc15940)

[2.1开发环境 8](#_Toc26806)

[2.2下位机程序设计 8](#_Toc3170)

[2.2.1主程序设计 8](#_Toc23175)

[2.2.2 程序流程图 8](#_Toc7697)

[2.2.3 LCD1602程序设计 9](#_Toc32054)

[2.2.3蜂鸣器程序设计 9](#_Toc1896)

[2.2.3无线通信程序设计 9](#_Toc17179)

[2.3上位机程序设计 9](#_Toc19866)

[2.3.1主程序设计 9](#_Toc20489)

[2.3.2 程序流程图 9](#_Toc4842)

[2.3.3 串口程序设计 10](#_Toc7149)

[2.3.4 接收数据程序设计 10](#_Toc15203)

[2.3.5 定时发送数据程序设计 10](#_Toc24366)

**[3 系统测试](#_Toc6196)** [11](#_Toc6196)

[3.1硬件测试 11](#_Toc16162)

[3.2软件测试 11](#_Toc22222)

**[结 论](#_Toc8981)** [12](#_Toc8981)

**[致 谢](#_Toc6730)** [13](#_Toc6730)

**[附 录](#_Toc7870)** [15](#_Toc7870)

绪 论

通过对硬件及软件的设计，实现上位机对下位机的控制，完成以PC机为上位机，单片机为下位机的双向通信。掌握PC机座位上我记以及单片机作为下位机的工作原理图，通过学习PC机与单片机的工作原理及编程方法，提高编程能力及创新能力，掌握Python、单片机语言中的设计和学会分析程序，进而能够根据自己的需要编写代码。

本次设计通过keil5单片机为下位机，Python为上位机实现PC机与单片机之间的串口通讯功能，串行端口的本质功能是作为CPU和串行设备间的编码转换器。当数据从CPU经过串口发送时，数据由字节转换为串行的位，CPU接收从串口发来的数据是，数据将串行的位转换为字节。以单片机为核心的数字时钟是很有社会意义和社会价值的。钟表原先的报时功能已经原不能满足人们日益增长的要求，现代的电子时钟多带有类似自动报警、按时自动打铃、时间程序自动控制、定时广播、自动起闭路灯、通断动力设备、甚至各种定时电气的自动启用等功能

数字闹钟通过数字电路实现时、分、秒。数字显示的计时装置,广泛用于个人家庭、车站、码头办公室等公共场所成为人们日常生活中不可少的必需品。由于数字集成电路的发展和石英晶体振荡器的广泛应用,使得数字钟的精度远远超过老式钟表。

多功能数字钟的应用非常普遍。由单片机作为数字钟的核心控制器，通过它的时钟信号进行实现计时功能，将其时间数据经单片机输出，利用显示器显示出来。通过键盘可以进行校时、定时等功能。输出设备显示器可以用液晶显示技术和数码管来显示技术。

本系统利用单片机实现具有计时、校时等功能的数字时钟，是以单片机STM32为核心元件同时采用LED数码管显示器动态显示“时”、“分”、“秒”使其具有校时功能，秒表功能，和定时器功能，利用单片机实现的数字时钟具有编程灵活，便于功能的扩充等优点。

串行通信采用异步通信，异步通信在发送字符时，所发送的字符之间的时间间隔可以是任意的。当然，接收端必须时刻做好接收的准备。发送端可以在任意时刻发送字符，因此必须在每一个字符的开始和结束的地方加上标志，即加上开始位和停止，以便接收端能够正确的将每一个字符接收下来。位当运行时，上位机界面显示预定时间、运行时间、串口号、按键及LCD显示数据，下位机LCD显示预定时间和运行时间，当运行时间达到预定时间时，蜂鸣器响起音乐，按键按下可切换音乐。

1 硬件部分设计

1.1硬件结构设计

本次设计硬件部分通过单片机、蜂鸣器等模块完成时钟的响应。通过对keil5单片机进行编程，连接蜂鸣器，控制蜂鸣器响应，通过按键控制蜂鸣器切换音乐，LCD屏幕显示数据，八路触摸模块（TTP226）控制按键操作。在单片机上设定一个预定时间，当程序启动，LCD屏幕上显示运行时间和预定时间，当运行时间达到预定时间时，蜂鸣器响起音乐，按下按键可以切换音乐。硬件总体结构框图如图1.1所示。

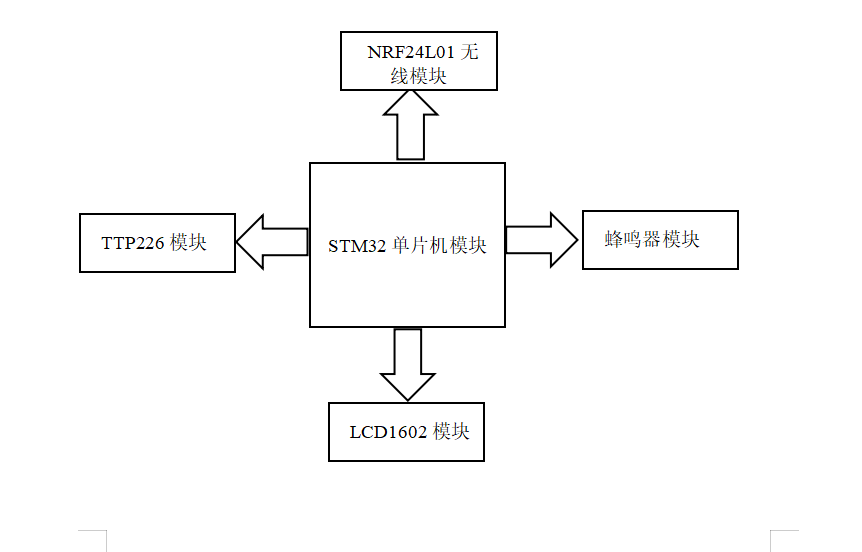


图1.1硬件总体结构模块

1.2硬件模块选择

1.2.1蜂鸣器模块设计

蜂鸣器是一种一体化结构的电子讯响器，采用直流电压供电，广泛应用于计算机、打印机、复印机、[报警器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=558213&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。蜂鸣器本质上是一个感性元件，其电流不能瞬变，因此必须有一个续流二极管提供续流。否则，在蜂鸣器两端会产生几十伏的尖峰电压，可能损坏驱动三极管，并干扰整个电路系统的其它部分。发声元件，在其两端施加直流电压（有源蜂鸣器）或者方波（无源蜂鸣器）就可以发声，其主要参数是外形尺寸、发声方向、工作电压、工作频率、工作电流、驱动方式（直流/方波）等。这些都可以根据需要来选择。蜂鸣器的发声原理由振动装置和谐振装置组成，而蜂鸣器又分为无源他激型与有源自激型。无源蜂鸣器则需要连接在音频输出电路中连接，才会周期性地振动发声。为了使负载获得良好的阻抗匹配，使负载获得最大功率。可以把把变压器的初级即220V的那个绕组，接到原来接高阻耳机的那个地方，把次级绕组接在原来接普通耳机的那个地方就好了。

蜂鸣器模块原理图如图1.2所示。

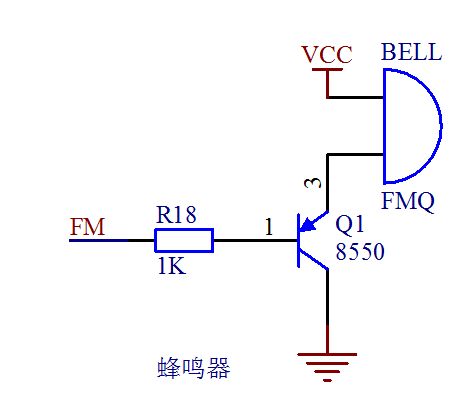


图1.2蜂鸣器模块原理图

1.2.2单片机模块设计

STM32F103ZET6是一种嵌入式微控制器的集成电路，芯体尺寸是32位，速度是72MHz，程序存储器容量是256KB，程序存储器类型是FLASH，RAM容量是48K；32位高性能ARM Cortex-M3处理器；时钟高达72M，实际还可以超频一点点；单周期乘法和硬件除法；IO口有144个引脚，112个IO，大部分IO口都耐5V(模拟通道除外)；存储器容量：512K FLASH，64K SRAM；时钟和复位和电源管理为2.0~3.6V电源和IO电压，上电复位，掉电复位和可编程的电压监控，强大的时钟系统；4~16M的外部高速晶振，内部8MHz的高速RC振荡器，40KHz低速RC振荡器；外部低速32.768K的晶振，主要做RTC时钟源；具备睡眠，停止和待机三种低功耗模式，可用电池为RTC和备份寄存器供电；3个12位AD（多达21个外部测量通道），转换范围为0~3.6V内部通道可以用于内部温度测量，内置参考电压；DA：2个12位DA；DMA有12个DMA通道支持外设：定时器，ADC，DAC，SDIO，I2S，SPI，I2C，和USART；包含多达11个定时器。4个通用定时器，2个基本定时器，2个高级定时器，1个系统定时器，2个看门狗定时器；含多达13个通信接口，2个I2C接口，5个串口，3个SPI接口，1个CAN2.0，1个USB FS，1个SDIO；STM32最小系统为供电，复位，时钟。

STM32原理图如图1.2所示。

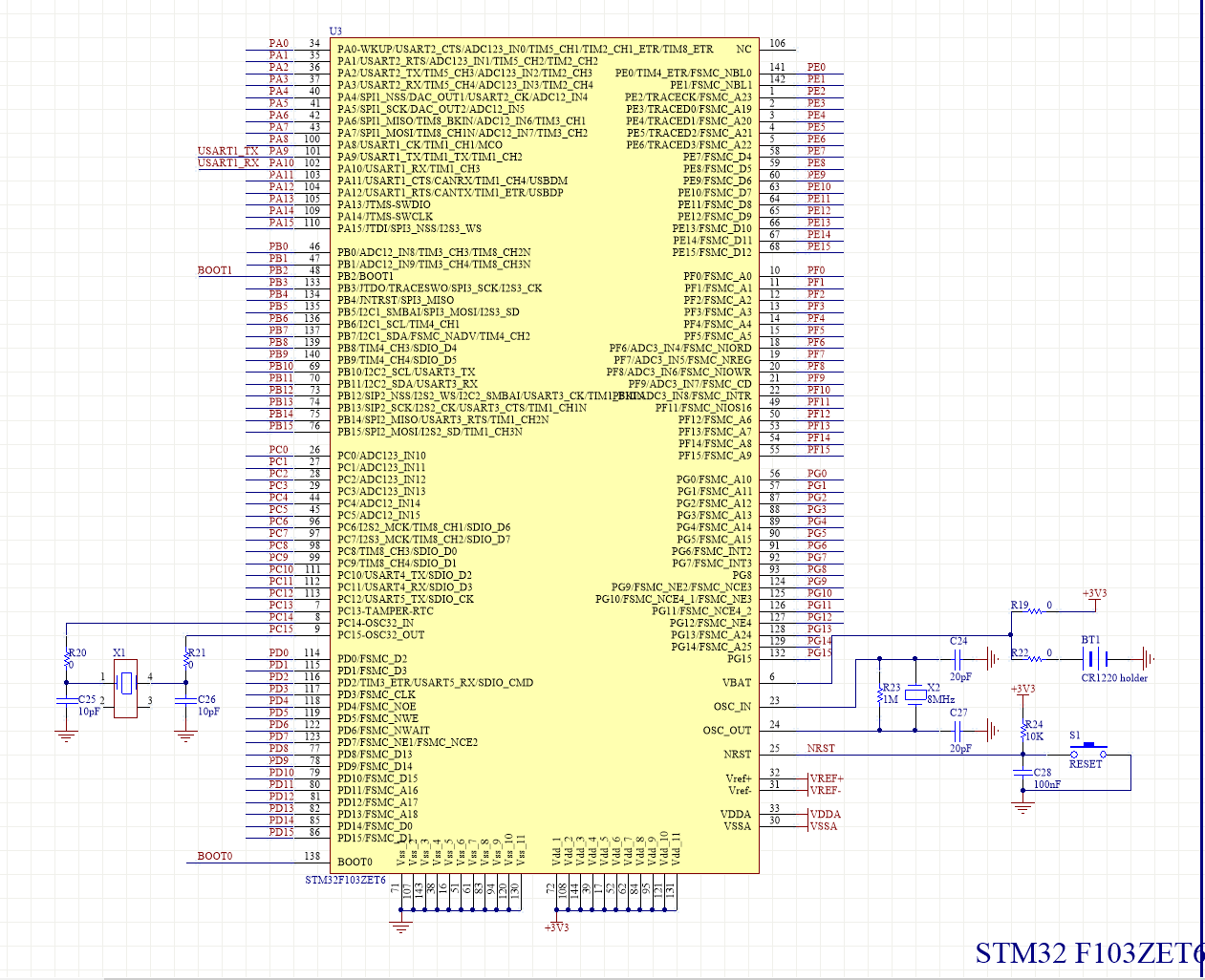


图1.2 单片机模块原理图

引脚说明：

浮空输入\_IN\_FLOATING

带上拉输入\_IPU

带下拉输入\_IPD4

模拟输入\_AIN

开漏输出\_OUT\_OD

推挽输出\_OUT\_PP

复用功能的推挽输出\_AF\_PP

复用功能的开漏输出\_AF\_OD

1.2.3 LCD显示模块

LCD1602液晶也叫1602字符型液晶，它是一种专门用来显示字母、数字、符号等的点阵型液晶模块。它由若干个5X7或者5X11等点阵字符位组成，每个点阵字符位都可以显示一个字符，每位之间有一个点距的间隔，每行之间也有间隔，起到了字符间距和行间距的作用，正因为如此所以它不能很好地显示图形（用自定义CGRAM，显示效果也不好）。它具有微功耗、体积小、显示内容丰富、超薄轻巧的特点，常用在袖珍式仪表和低功耗应用系统中。

技术参数：

显示容量：16&TImes;2个字符

芯片工作电压：4.5—5.5V

工作电流：2.0mA（5.0V）

模块最佳工作电压：5.0V

字符尺寸：2.95&TImes;4.35（W&TImes;H）mm

LCD1602模块原理图如图1.4所示。

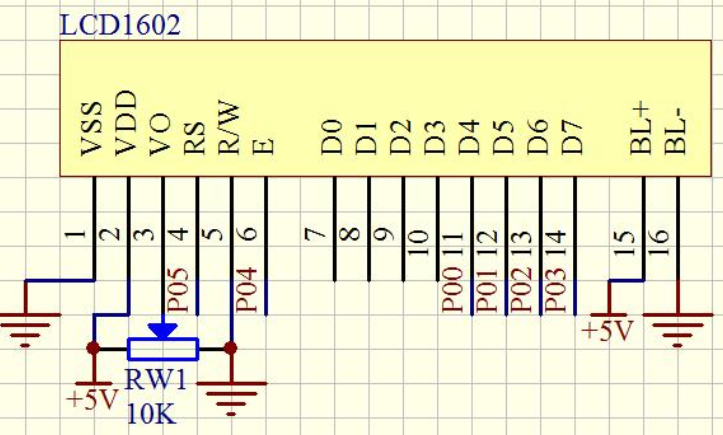


图1.4 LCD1602模块原理图

引脚说明：

第1脚：VSS为电源地

第2脚：VCC接5V电源正极

第3脚：V0为液晶显示器对比度调整端，接正电源时对比度最弱，接地电源时对比度最高（对比度过高时会 产生“鬼影”，使用时可以通过一个10K的[电位器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=3147400&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)调整对比度）。

第4脚：RS为寄存器选择，高电平1时选择[数据寄存器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=55934744&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)、低电平0时选择[指令寄存器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=2425481&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)。

第5脚：RW为读写信号线，高电平(1)时进行读操作，低电平(0)时进行写操作。

第6脚：E(或EN)端为使能(enable)端,高电平（1）时读取信息，负跳变时执行指令。

第7～14脚：D0～D7为8位双向数据端。

第15～16脚：空脚或背灯电源。15脚背光正极，16脚背光负极。

1.2.4 NRF24L01无线模块

[NRF24L01](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=63532287&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)是一款新型单片射频收发器件，工作于2.4 GHz～2.5 GHz ISM频段。内置[频率合成器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=8438874&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)、[功率放大器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=4923693&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)、[晶体振荡器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=509059&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)、[调制器](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7918418&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)等功能模块，并融合了增强型ShockBurst技术，其中输出功率和通信频道可通过程序进行配置。

其原理图如图1.5所示。

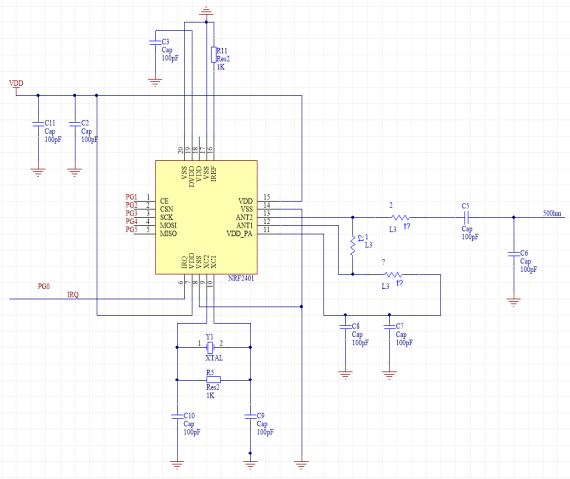


图1.5 NRF24L01无线模块原理图

性能参数：

（1）小体积，QFN20 4x4mm封装

（2）宽电压工作范围，1.9V~3.6V，输入引脚可承受5V电压输入

（3）工作温度范围，-40℃~+80℃

（4）工作频率范围，2.400GHz~2.525GHz

（5）发射功率可选择为0dBm、-6dBm、-12dBm和-18dBm

（6）数据传输速率支持1Mbps、2Mbps

（7）低功耗设计，接收时工作电流12.3mA，0dBm功率发射时11.3mA，掉电模式时仅为900nA

（8）126个通讯通道，6个数据通道，满足多点通讯和调频需要

（9）增强型"ShockBurst"工作模式，硬件的CRC校验和点对多点的地址控制

（10）数据包每次可传输1~32Byte的数据

（11）4线SPI通讯端口，通讯速率最高可达8Mbps，适合与各种MCU连接，编程简单

（12）可通过软件设置工作频率、通讯地址、传输速率和数据包长度

（13）MCU可通过IRQ引脚块判断是否完成数据接收和数据发送

1.2.4 TTP226八路触摸模块

八路触摸模块TTP226 TonTouch"是一款接触板检测IC，提供8个接触键。接触检测IC有意取代传统固定pad尺寸的直接按钮键。低功耗和宽工作电压是接触键在DC或AC应用中的特点。其原理图如图1.6所示。

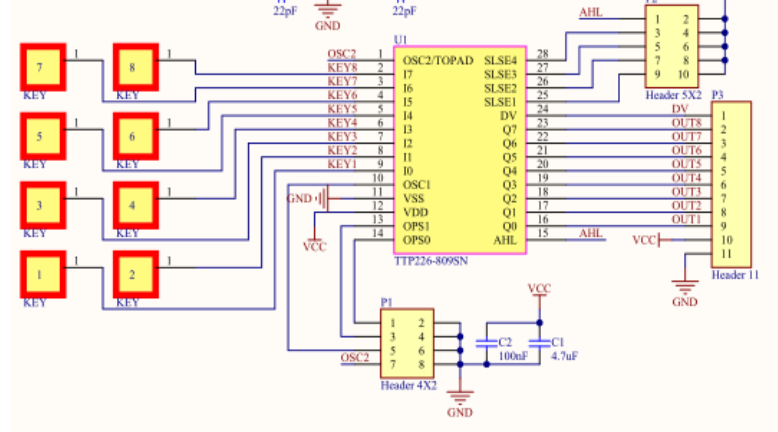


图1.6 TTP226模块原理图

工作特点：

（1）工作电压2.0V~5.5。

（2）工作电流在VDD=3V时典型值8OuA，最大值160uA输出刷新率在VDD=3V时约55Hz。

（3）64 阶可选灵敏度(SLSEO~5管脚选项)。

（4）另外提供2种基阶(base-step)(OPST管脚选项)。

（5）稳定的人体接触检测，以取代传统直接切换的键(direct switch key)。

（6）提供直接(direct)模式、矩阵(matrix)模式和串行(serial)模式,由pad选项选择。

（7）固定的2\*4和3\*3矩阵类型提供最多8个输入pads.输出可由pad选项选择为高电平有效或低电平有效。

（8）在上电之后有一段稳定时间，在此期间不要触摸键区(key-pad)，且功能无效，TTP226的是0.8~1.0秒

（9）始终进行自校准，当所有键没被触摸时，重校准周期TTP226的是0.8~1.0秒

2 软件部分设计

2.1开发环境

Keil uVision5是一款集编辑，编译和项目管理于一身的基于窗口的软件开发环境。uVision5具有如下特性:功能齐全的源代码编辑器，用于配置开发工具的设备库，用于创建工程和维护工程的项目管理器，所有的工具配置都采用对话框进行，集成了源码级的仿真调试器，包括高速CPU和外设模拟器,用于往Flash ROM下载应用程序的Flash编程工具，完备的开发工具帮助文档，设备数据表和用户使用向导。

2.2下位机程序设计

2.2.1主程序设计

整个系统要完成的任务是从发射端接收发来的数据，在LCD模块上显示预定时间和运行时间，当预定时间和运行时间相同时，蜂鸣器响音乐，按按键控制音乐切换。

2.2.2 程序流程图

下位机程序流程图如图2.1所示。

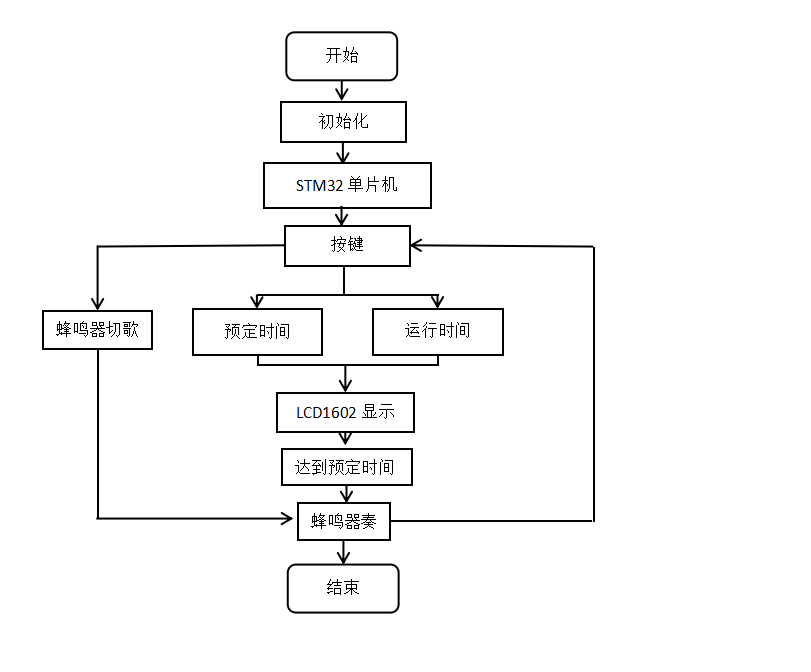


图2.1 下位机程序流程图

2.2.3 LCD1602程序设计

LCD1602模块屏幕上显示预定时间和运行时间

for(i=0;i<100;i++)

GPIO\_ResetBits(GPIOB,GPIO\_Pin\_1);

delay\_ms(50);

GPIO\_SetBits(GPIOB,GPIO\_Pin\_1);

delay\_ms(50);

2.2.3蜂鸣器程序设计

达到预定时间，蜂鸣器奏乐，按键切换音乐

if(music==0)

play\_music();  
if(KEY3==0)  
GPIO\_SetBits(GPIOB,GPIO\_Pin\_1);

2.2.3无线通信程序设计

通过无线通信实现设备间的通信。

if(NRF\_IRQ==0)

if(NRF24L01\_RxPacket(tmp\_buf)==0)

if( tmp\_buf[1]=='1')

printf("sd");

if( tmp\_buf[1]=='2')

2.3上位机程序设计

2.3.1主程序设计

上位机显示串口助手界面，上位机下位机连接后，上位机首先对硬件进行检测，查找是否有可用的COM端口，并对该端口进行简要判断，包括这些端口是否是串口，是否正在使用。通过计算机对串口的自检后，可以对串口参数进行简单的配置。

2.3.2 程序流程图

上位机程序流程图如图2.2所示

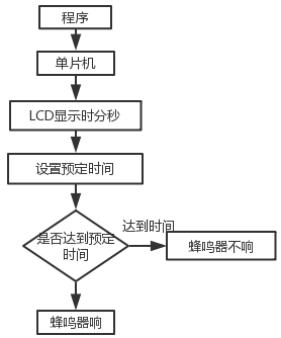


图2.2 上位机程序流程图

2.3.3 串口程序设计

def port\_open(self):

self.ser.port = self.s1\_\_box\_2.currentText()

self.ser.baudrate = int(self.s1\_\_box\_3.currentText())

self.ser.bytesize = int(self.s1\_\_box\_4.currentText())

self.ser.stopbits = int(self.s1\_\_box\_6.currentText())

self.ser.parity = self.s1\_\_box\_5.currentText()

2.3.4 接收数据程序设计

def data\_receive(self):

try: num = self.ser.inWaiting()

except: self.port\_close() return None

if num > 0:

data = self.ser.read(num) num = len(data)

2.3.5 定时发送数据程序设计

def data\_send\_timer(self):

if self.timer\_send\_cb.isChecked():

self.timer\_send.start(int(self.lineEdit\_3.text()))

self.lineEdit\_3.setEnabled(False)

else:

self.timer\_send.stop()

self.lineEdit\_3.setEnabled(True)

3 系统测试

3.1硬件测试

上电后，LCD屏幕第一行显示预定时间，第二行显示运行时间并开始计时，当运行时间达到预定时间时，蜂鸣器奏乐，通过按键可进行音乐的切换。LCD屏幕显示如图3.1所示。

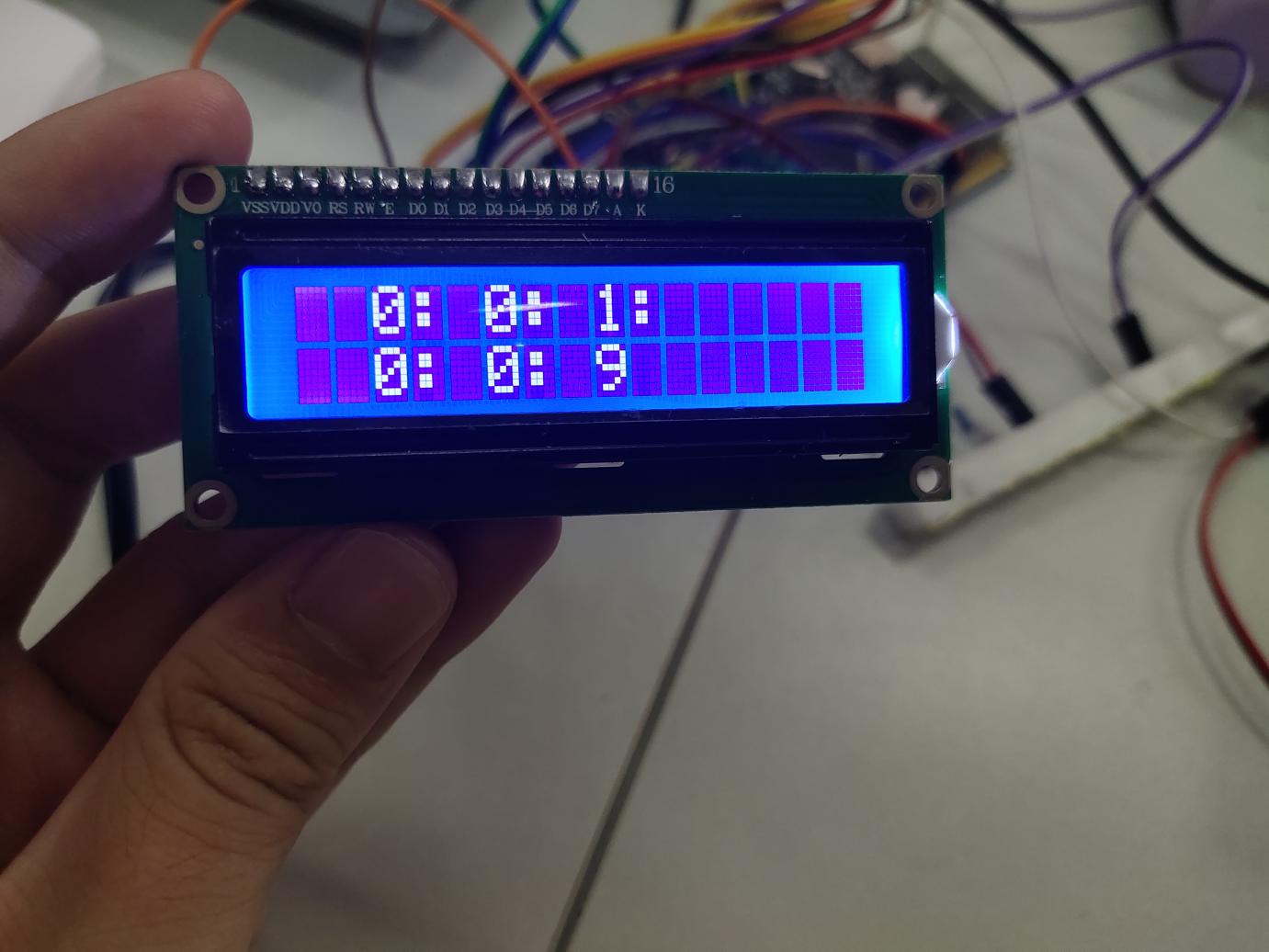


图3.1 LCD屏幕显示图

3.2软件测试

上位机程序运行后，出现界面，首先进行硬件检测，查找是否有可用的COM端口，连接好硬件电路，在上位机界面发送测试消息，下位机收到后向上位机发送消息，显示在上位机界面接收区中，上位机界面如图3.2所示。

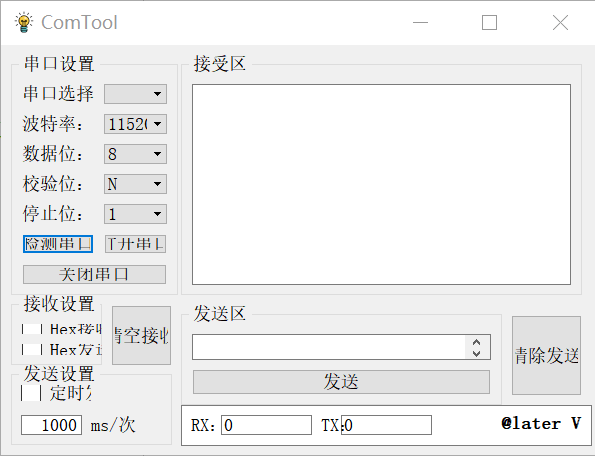


图3.2上位机界面

结 论

本次实训的目的通过对上位机下位机的编程，实现了上位机下位机之间的通信，用此制作了无线遥控电子时钟，通常上位机指的是PC，下位机指的是单片机或者带微处理器的系统。下位机一般是将模拟信号经过AD采集将模拟量转换为数字量，下位机再经过数字信号处理以后将数字信号通过串口发送到上位机，相反上位机可以给下位机发送一些指令或者信息。常见的通信串口包括RS232、RS485、RS422等。这些串口只是在电平特性有所不同，在上位机与下位机进行数据通信时可以不考虑电平特性，而且现在在硬件上有各种转接接口，使用起来也很方便。串口是计算机上一种非常通用的设备通信协议。大多数计算机包含两个基于RS-232的串口。串口同时也是仪器仪表设备通用的通信协议；很多GPIB兼容的设备也带有RS-232口。同时，串口通信协议也可以用于获取远程采集设备的数据。

通过上下位机连通后，下位机蜂鸣器奏乐，通过按键控制切换音乐，上位机界面显示实时数据。

致 谢

通过这次实训，真正了解并掌握了单片机的开发应用和编程控制。为以后从事单片机软硬件产品的设计开发、打下了良好的基础，树立独立从事产品研发的信心，并在这种能力上得到了比较充分的锻炼。本学期实训的实验内容是制作无线遥控智能电子时钟，过程非常考验思考能力且有趣生动。在动手操作的同时又学习了单片机的相关知识，对于专业课程很有帮助，做到了理论课程学习+实际应用操作。通过此次的实训制作，更加认识到了动手能力和理论知识的重要性，而理论与实践的结合更是重中之重。做学问搞研究，真的不是一件容易的事情,但我们必须学会在苦中找乐，把它变成-件快乐的事。从论文选题到今天顺利完成，我经历了人生的一次考验，学到的远远超过写论文本身，这样的经历也让我更加充满自信去面对未来的生活。在这个过程中，还有许多可敬的师长、同学、朋友给了我无言的帮助，在这里无法提及，感谢你们。

能在这几年的学习中汲取大量专业知识，迅速提高自身能力。同时，也要感谢你们在论文选题、开题过程中对我提出的宝贵和中肯的意见，正是你们的建议和指导才得以让我的论文更加完善，感谢你们!能在这几年的学习中汲取大量专业知识，迅速提高自身能力。同时，也要感谢你们在论文选题、开题过程中对我提出的宝贵和中肯的意见，正是你们的建议和指导才得以让我的论文更加完善，感谢您们!求学之路，因为有了同窗好友的陪伴，才会显得格外的精彩和让人难忘。

参考文献

[1]刘正扬.基于DP801单片机技术的智能限高和涵洞积水预警系统的研究[J].价值工程,2019(20):230-232.

1. 金琦淳,李俊斌,任俊,袁明新,王琪.基于IAP15F2K61S2的多驱动控制救援服务小车[J].机械与电子,2019(07):43-48+53.
2. 杨晓燕,徐广振,王琰琰.基于stm32单片机的表面贴装自动质检装置的设计[J].内燃机与配件,2019(13):240-241.
3. 曲鸣飞,陈楠.基于单片机的机械设备显示器触摸屏控制系统设计[J].内燃机与配件,2019(13):277-279.
4. 杨欢,杜少华,袁国锋,陈晓.基于Proteus的单片机A/D转换仿真实验[J].教育教学论坛,2019(30):278-280.
5. 周丽荣.物联网电子产品中单片机技术的应用研究[J].电子测试, 2018(02):79-80.
6. 牛景乐, 鲍权鑫,鲍丙豪.基于单片机和蓝牙技术的智能风扇设计[J].仪表技术, 2018(03) :23-26+46.

附 录

上位机：

def port\_check(self):

# 检测所有存在的串口，将信息存储在字典中

self.Com\_Dict = {}

port\_list = list(serial.tools.list\_ports.comports())

self.s1\_\_box\_2.clear()

for port in port\_list:

self.Com\_Dict["%s" % port[0]] = "%s" % port[1]

self.s1\_\_box\_2.addItem(port[0])

if len(self.Com\_Dict) == 0:

self.state\_label.setText(" 无串口")

# 串口信息

def port\_imf(self):

# 显示选定的串口的详细信息

imf\_s = self.s1\_\_box\_2.currentText()

if imf\_s != "":

self.state\_label.setText(self.Com\_Dict[self.s1\_\_box\_2.currentText()])

# 打开串口

def port\_open(self):

self.ser.port = self.s1\_\_box\_2.currentText()

self.ser.baudrate = int(self.s1\_\_box\_3.currentText())

self.ser.bytesize = int(self.s1\_\_box\_4.currentText())

self.ser.stopbits = int(self.s1\_\_box\_6.currentText())

self.ser.parity = self.s1\_\_box\_5.currentText()

try:

self.ser.open()

except:

QMessageBox.critical(self, "Port Error", "此串口不能被打开！")

return None

# 打开串口接收定时器，周期为2ms

self.timer.start(2)

if self.ser.isOpen():

self.open\_button.setEnabled(False)

self.close\_button.setEnabled(True)

self.formGroupBox1.setTitle("串口状态（已开启）")

# 关闭串口

def port\_close(self):

self.timer.stop()

self.timer\_send.stop()

try:

self.ser.close()

except:

pass

self.open\_button.setEnabled(True)

self.close\_button.setEnabled(False)

self.lineEdit\_3.setEnabled(True)

# 接收数据和发送数据数目置零

self.data\_num\_received = 0

self.lineEdit.setText(str(self.data\_num\_received))

self.data\_num\_sended = 0

self.lineEdit\_2.setText(str(self.data\_num\_sended))

self.formGroupBox1.setTitle("串口状态（已关闭）")

# 发送数据

def data\_send(self):

if self.ser.isOpen():

input\_s = self.s3\_\_send\_text.toPlainText()

if input\_s != "":

# 非空字符串

if self.hex\_send.isChecked():

# hex发送

input\_s = input\_s.strip()

send\_list = []

while input\_s != '':

try:

num = int(input\_s[0:2], 16)

except ValueError:

QMessageBox.critical(self, 'wrong data', '请输入十六进制数据，以空格分开!')

return None

input\_s = input\_s[2:].strip()

send\_list.append(num)

input\_s = bytes(send\_list)

else:

# ascii发送

input\_s = (input\_s + '\r\n').encode('utf-8')

num = self.ser.write(input\_s)

self.data\_num\_sended += num

self.lineEdit\_2.setText(str(self.data\_num\_sended))

else:

pass

# 接收数据

def data\_receive(self):

try:

num = self.ser.inWaiting()

except:

self.port\_close()

return None

if num > 0:

data = self.ser.read(num)

num = len(data)

# hex显示

if self.hex\_receive.checkState():

out\_s = ''

for i in range(0, len(data)):

out\_s = out\_s + '{:02X}'.format(data[i]) + ' '

self.s2\_\_receive\_text.insertPlainText(out\_s)

else

self.s2\_\_receive\_text.insertPlainText(data.decode('iso-8859-1'))

# 统计接收字符的数量

self.data\_num\_received += num

self.lineEdit.setText(str(self.data\_num\_received))

# 获取到text光标

textCursor = self.s2\_\_receive\_text.textCursor()

# 滚动到底部

textCursor.movePosition(textCursor.End)

# 设置光标到text中去

self.s2\_\_receive\_text.setTextCursor(textCursor)

else:

pass

# 定时发送数据

def data\_send\_timer(self):

if self.timer\_send\_cb.isChecked():

self.timer\_send.start(int(self.lineEdit\_3.text()))

self.lineEdit\_3.setEnabled(False)

else:

self.timer\_send.stop()

self.lineEdit\_3.setEnabled(True)

# 清除显示

def send\_data\_clear(self):

self.s3\_\_send\_text.setText("")

def receive\_data\_clear(self):

self.s2\_\_receive\_text.setText("")