**中国地质大学课程实习报告**

**智能地学虚拟仪器设计**



班 级： 231192

学 号： 20191000663

姓 名： 张宇一

组 员： 张宇一 周汶昊 舒俊 王珺璟

二○二二年一月二十

目录

[一、绪论 3](#_Toc8878)

[1. 实习目的 3](#_Toc10229)

[2. 实习环境 3](#_Toc12990)

[2.1 硬件环境 3](#_Toc32377)

[2.2 软件环境 3](#_Toc25398)

[二、 概要设计 4](#_Toc30879)

[2.1 系统流程图 4](#_Toc23351)

[2.2 功能模块设计概述 4](#_Toc16875)

[2.2.1 波形发射模块 5](#_Toc15753)

[2.2.2 数据采集模块 5](#_Toc27123)

[三、 LAB9000平台搭建 6](#_Toc15252)

[3.1 发射部分 6](#_Toc25603)

[3.2 接收部分 6](#_Toc27892)

[3.3 实物连线图展示 7](#_Toc31960)

[四、 LabVIEW设计部分 8](#_Toc30902)

[4.1 LabVIEW部分系统整体设计框图 8](#_Toc16236)

[4.2 LabVIEW各模块设计描述 8](#_Toc17802)

[4.2.1 发射模块设计 8](#_Toc18660)

[4.2.2 采集模块设计 11](#_Toc20582)

[4.2.3 一体化模块设计 12](#_Toc19740)

[五、 单片机程序设计 14](#_Toc1398)

[5.1 发射端程序 14](#_Toc22631)

[5.1.1 波形产生原理 14](#_Toc5447)

[5.1.2 发射端程序解析 14](#_Toc125)

[5.2 采集部分程序 19](#_Toc20623)

[5.2.1 采集原理 19](#_Toc1452)

[5.2.2 采集部分程序解析 19](#_Toc9992)

# 一、绪论

## 1. 实习目的

①通过本次实习提高学生的动手能力；

②通过本次实习加深学生对虚拟仪器课程知识点的理解；

③通过本次实习使学生掌握LabVIEW的使用；

④掌握简单的51单片机编程等。

## 实习环境

### 硬件环境

本次实习需要自带安装用Keil、LabVIEW等软件的电脑、LAB9000伟福实验箱、自制的全桥驱动。

### 软件环境

#### 2.2.1 LabVIEW软件

LabVIEW是一种程序开发环境，由[美国](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD/125486" \t "https://baike.baidu.com/item/LabVIEW/_blank)国家仪器（NI）公司研制开发，类似于C和BASIC开发环境，但是LabVIEW与其他计算机语言的显著区别是：其他计算机语言都是采用基于文本的语言产生代码，而LabVIEW使用的是图形化编辑语言G编写程序，产生的程序是框图的形式。

LabVIEW软件是NI设计平台的核心，也是开发测量或控制系统的理想选择。 LabVIEW开发环境集成了工程师和科学家快速构建各种应用所需的所有工具，旨在帮助工程师和科学家解决问题、提高生产力和不断创新。

#### 2.2.2 Keil软件

Keil C51是美国Keil Software公司出品的51系列兼容[单片机](https://baike.baidu.com/item/%E5%8D%95%E7%89%87%E6%9C%BA/102396" \t "https://baike.baidu.com/item/_blank)C语言软件开发系统，与[汇编](https://baike.baidu.com/item/%E6%B1%87%E7%BC%96/627224" \t "https://baike.baidu.com/item/_blank)相比，[C语言](https://baike.baidu.com/item/C%E8%AF%AD%E8%A8%80/105958" \t "https://baike.baidu.com/item/_blank)在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil提供了包括C[编译器](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8/8853067" \t "https://baike.baidu.com/item/_blank)、宏汇编、链接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个[集成开发环境](https://baike.baidu.com/item/%E9%9B%86%E6%88%90%E5%BC%80%E5%8F%91%E7%8E%AF%E5%A2%83/298524" \t "https://baike.baidu.com/item/_blank)（μVision）将这些部分组合在一起。

# 概要设计

## 2.1 系统流程图

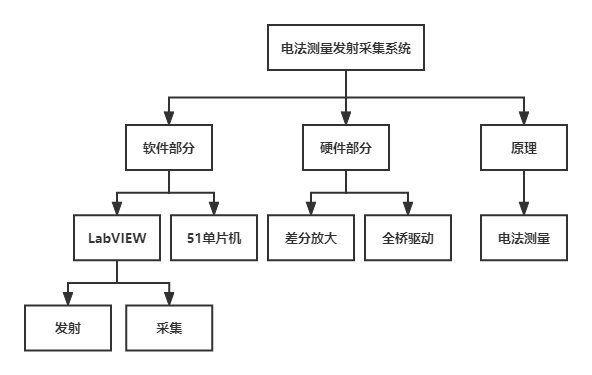


图 2.1 系统设计流程图

## 2.2 功能模块设计概述

本次实习所需要设计的功能模块有：

①波形发射模块：通过定时器产生频率可调的互补或非互补PWM波；

②功率驱动模块：由于单片机发出的PWM波最大电压只有5V故需要功率放大驱动模块将5V的PWM波提升至12V输入到土地中；

③数据采集模块：LAB9000上自带AD转换模块；

④串口通信。

### 波形发射模块

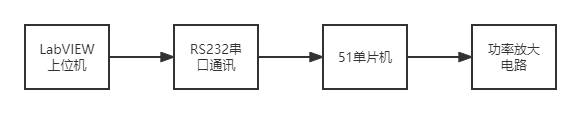


图 2.2 波形发射流程图

波形发射流程图如图2.2所示。在上位机LabVIEW的界面中选择发射频率，通过RS232串口通讯将指令发送至51单片机，在51单片机中对数据进行解析，并按照指令从单片机的P1口进行PWM输出，再经过功率放大电路将PWM波输入待测量的土地中。

### 数据采集模块

在地学虚拟仪器设计中，数据采集系统是计算机与外部物理世界的桥梁，将 被测对象（外部世界、现场）的各种参量通过各种传感器和变送器做适当转换后，在经过信号调理、采样、量化、编码、传输等步骤，最后送到处理器进行数据处理，存储记录和显示打印过程。

地学探测信号一般都是微弱信号，在时间域探测信号中，有效信号持续时间 非常短，衰减很快。所以，在地学探测仪器设计中，对数据采集精度和速度都有 很高的要求，限于实验条件，本设计主要以原理性设计为主。



图 2.3 数据采集流程图

ADC 数据上传原理具体如图2.3所示，首先将外部变换得到后的信号输入 到ADC0809 模数转化芯片中，ADC0809 是美国国家半导体公司生产的 CMOS工艺 8 通道，8 位逐次逼近式 A/D 模数转换器。其内部有一个 8 通道多路开关，它可以根据地址码锁存译码后的信号，只选通 8 路模拟输入信号中的一个进行 A/D 转换，而且输入电压为 0-5V。ADC0809 通过逐次逼近 A/D 转化的方法进行 ADC 值采集，然后将数据存储在 51 单片机系统中，然后通过 RS232通信将数据实时 发送到上位机 Labview 中显示。

# LAB9000平台搭建

## 3.1 发射部分

表 3.1 发射部分连线

|  |  |  |
| --- | --- | --- |
| 序号 | 连线孔1 | 连线孔2 |
| 1 | P1.0 | 全桥驱动DRIVE1、3 |
| 2 | P1.1 | 全桥驱动DRIVE2、4 |
| 3 | 电脑USB | RS232串口输入 |
| 4 | P3.0 | RS232的RXD |
| 5 | P3.1 | RS232的TXD |

LAB9000发射部分如表3.1所示。电脑通过RS232与单片机进行通信，由此可将用户的指令通过LabVIEW界面发送给单片机。注意P3.0、P3.1分别为单片机的TXD、RXD复用口，不可接反（常常会认为RXD接RXD、TXD接TXD）。单片机通过P1.0和P1.1输出波形，并连接到全桥驱动进行放大。

## 3.2 接收部分

|  |  |  |
| --- | --- | --- |
| 序号 | 连线孔1 | 连线孔2 |
| 1 | P3.0 | RS232的RXD |
| 2 | P3.1 | RS232的TXD |
| 3 | 电脑USB | RS232串口输入 |
| 4 | ADC0809 IN0 | 外部输入（滑动变阻器） |
| 5 | CS0 | 模数转换片选 |
| 6 | CS1 | 8255片选 |
| 7 | PA0~PA7 | L0~L7 |

ADC数据上传实验平台的接线方式如下表3.2所示。电脑通过RS232串口线连接到试验箱的RS232接口下发指令，RS232处的 RXD 与 P3.0、RS232处的TXD与 P3.1 相连，将 51 单片机系统与 232 电平转换部分连接。在模数转换部分将外部信号输入来连接到 IN0，将 模数转换片选连接到片选地址 CS0。在扩展引脚部分，将 8255片选连接到片选地址 CS1，最后将PA依此连接到 LED灯。

## 3.3 实物连线图展示

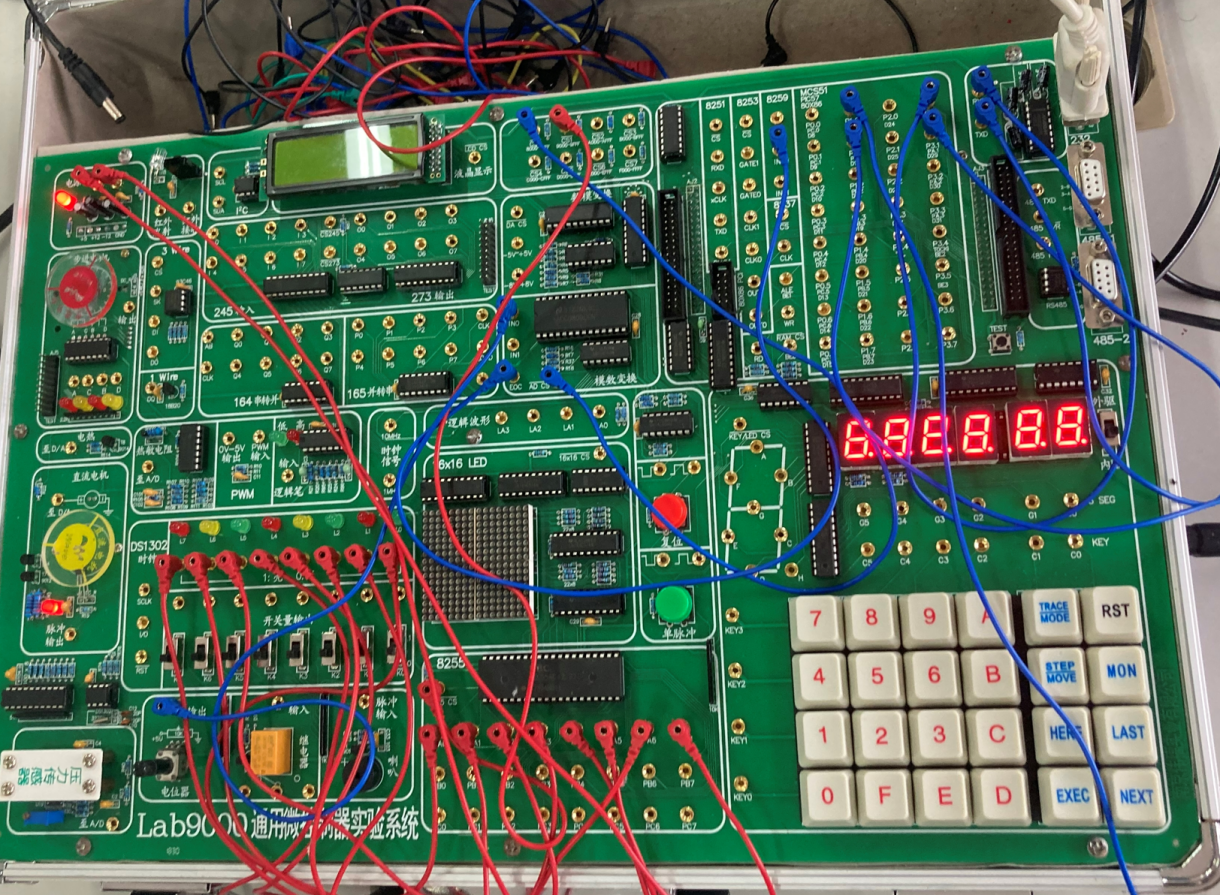


图 3.1 实物连线图（整体）

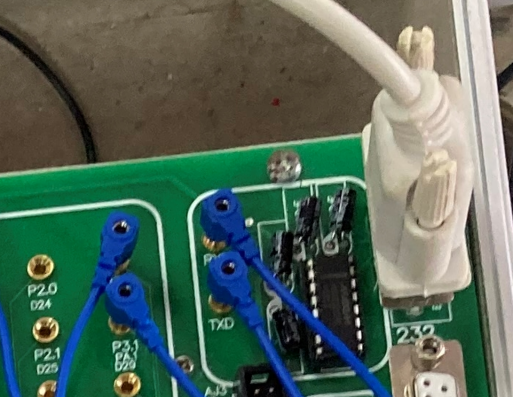
 

图 3.2 RS232部分连接图 图 3.3 模数转换部分连接图

# LabVIEW设计部分

## 4.1 LabVIEW部分系统整体设计框图

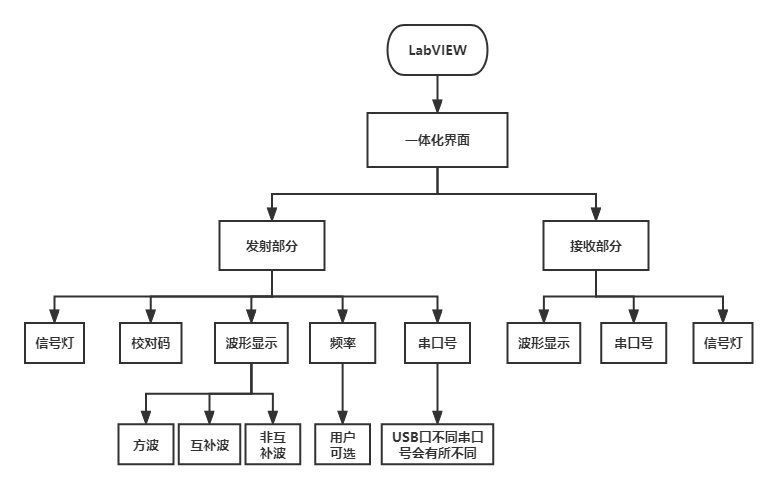


图 4.1 LabVIEW部分系统总流程图

## 4.2 LabVIEW各模块设计描述

根据系统功能要求可以将系统分解成几个模块来分别设计应用程序界面。

这里需要用到的知识有：(1) LabVIEW 循环结构； (2) LabVIEW 事件结构； (3) LabVIEW 顺序程序结构； (4) LabVIEW 串口通讯； (5) LabVIEW 基本数据类型（数值、 字符串和枚举等）及其操作函数； (6) LabVIEW 复杂数据类型（数组、簇等）及其操作函 数； (7) LabVIEW 图形显示（波形图、波形图表等）； (8) LabVIEW 文件 I/O 操作； (9) LabVIEW 信号生成； (10) LabVIEW 波形调理（滤波器）。

### 4.2.1 发射模块设计

**（1）程序框图**

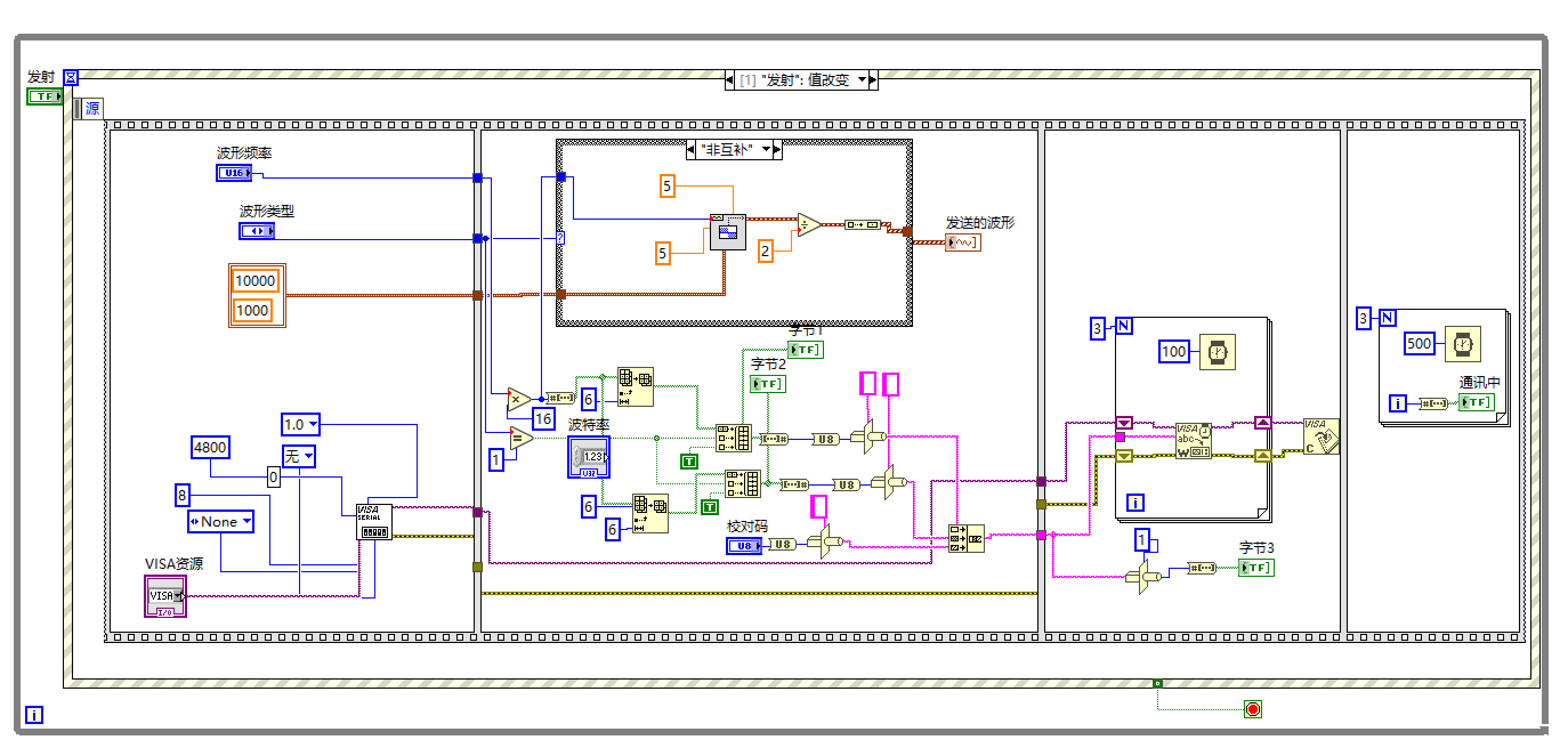


图 4.2 发射模块中的发射部分

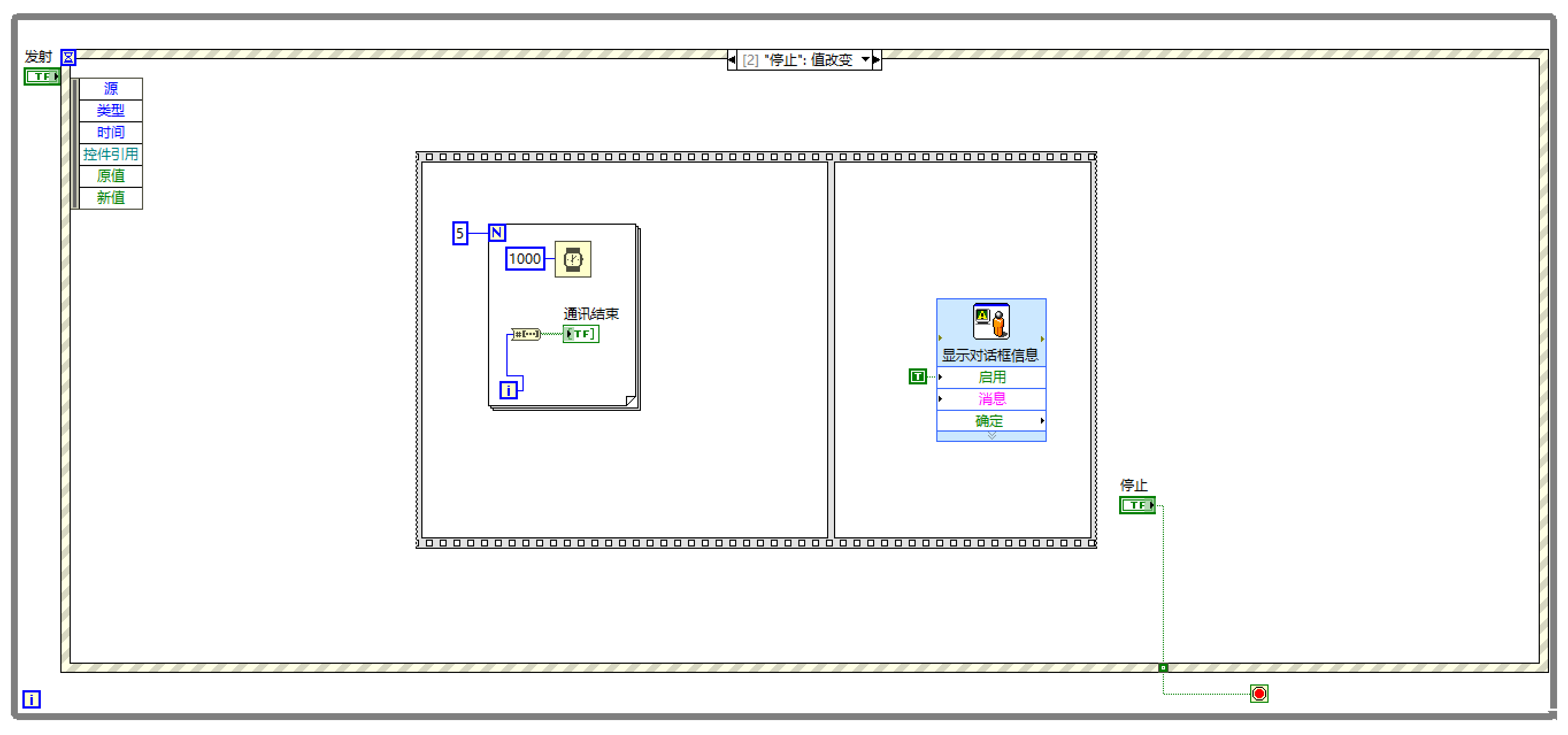


图 4.3 发射模块中的停止部分

1. **设计原理**

发送模块为 While循环嵌套事件结构嵌套顺序结构。在 while 循环内VISA配置串口进行读取，将 while 循环内部自带的“循环条件”连接到“关闭”控件上，使用该按钮来控制系统的关闭。在事件结构中发射按钮值改变事件，停止按钮值改变事件。按下停止出现如下弹框：

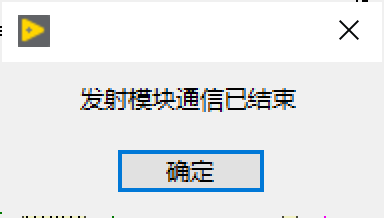


图 4.4 发射结束后的弹窗界面

按下发射按钮后在顺序程序结构中首先执行 VISA 配置串口；枚举类型数据（设置激励类型）、数值输入控件（设置激励频率），这里需要注意信号界限。其次在条件结构中输出波形进行激励信号显示，波形有互补波和非互补波，并将带有ASCII码的整数转换为相应的ASCII字符。之后在For循环中执行VISA写入与VISA关闭。

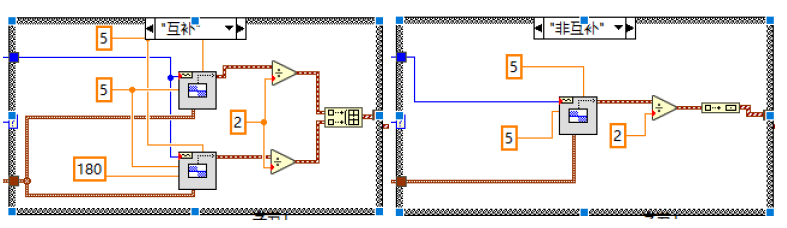


图 4.5 两种输出波形的方式

在进行数据转换时需要将带有 ASCII 码的整数转换为相应的 ASCII 字符。将ASCII代码连接到Type Cast VI的左侧输入并确保其格式为 U8（单个ASCII 字符或代码仅占用一个字节的信息）。Type Cast VI 的输出将是所需的 ASCII 字符。在此程序中用到三次数据转换，分别为前1-6位，6-12位校对码，剩下的位数为标志位。

1. **界面设计**



图 4.6 发射模块界面设计图

### 4.2.2 采集模块设计

（1）**程序框图**

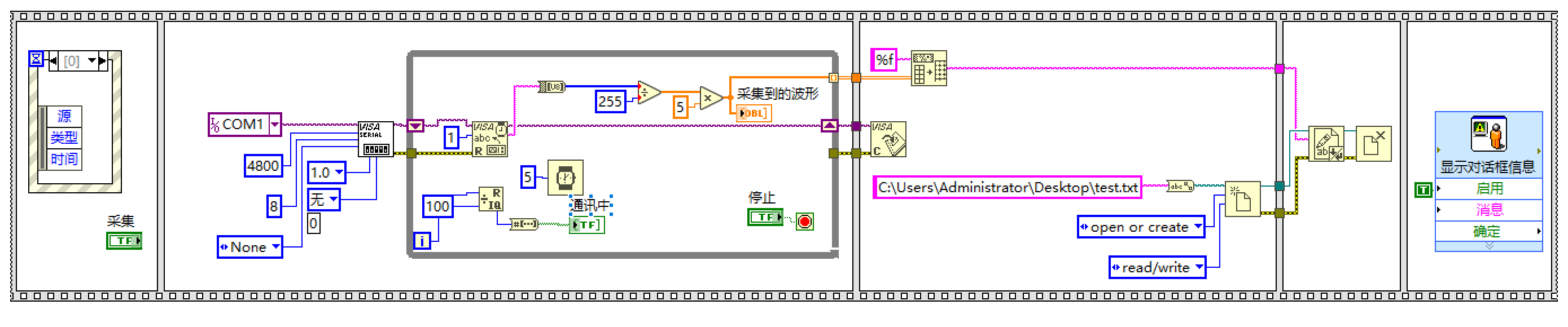


图 4.7 采集模块程序框图

1. **设计原理**

采集模块执行顺序程序结构，在采集部分：首先执行事件结构（按下采集按 钮值改变事件），其次在while循环内VISA配置串口进行读取，将while循环内部自带的“循环条件”连接到“关闭”控件上，使用该按钮来控制系统的关闭，之后进行信号数据转换与显示（波形图表），VISA关闭。在数据处理部分：首先打开或创建文件，再读取文件从而创建波形数据，经过滤波处理后关闭文件。

1. **界面设计**



图 4.8 数据采集模块界面设计

### 4.2.3 一体化模块设计

**（1）程序框图**

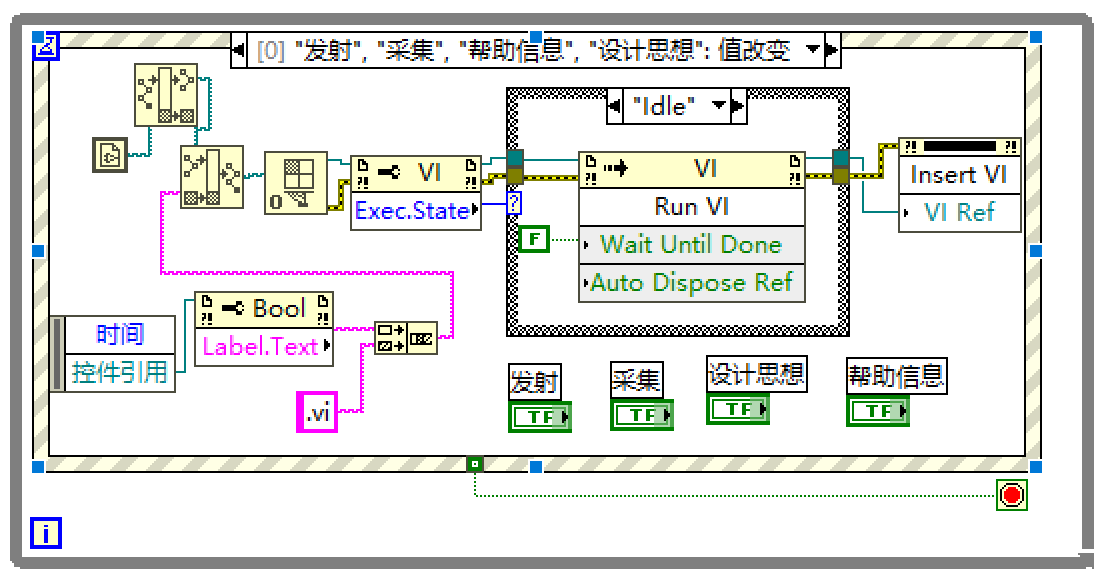


图 4.9 一体化模块主界面程序框图

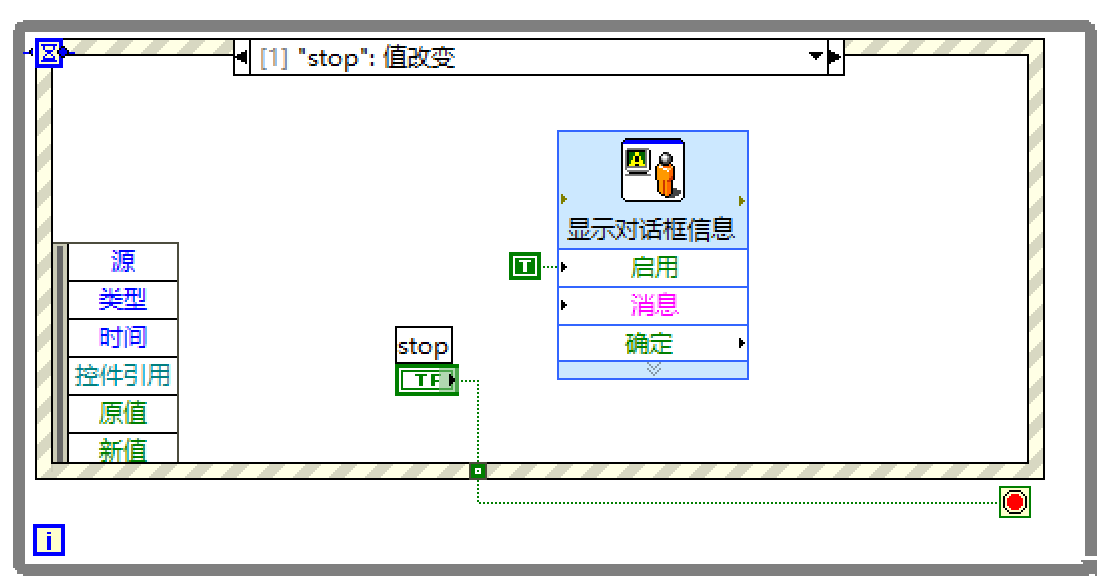


图 4.10 停止后的弹窗界面程序框图

程序运行后默认进入图4.9所示的主界面，在用户按下“STOP”按钮后，弹窗表示使用结束。

1. **设计原理**

将 while 循环内部自带的“循环条件”连接到“关闭”控件上，使用该按钮来控制系统的关闭。While 循环嵌套事件结构。

当事件发生时，首先放置两个按钮分别代表发射和采集用文件名命名，其次 在前面板中拖入一个子面板容器，然后在程序面板中我们可以看到一个调用节点， 右上角打开 vi 引用，将按键代表的名字加上.vi 传入，从而完成界面多面板切换。

当按下停止时表示该系统使用完毕出现如下弹窗：

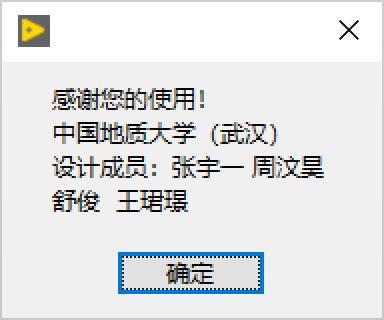


图 4.11 使用结束后的弹窗界面

1. **界面设计**

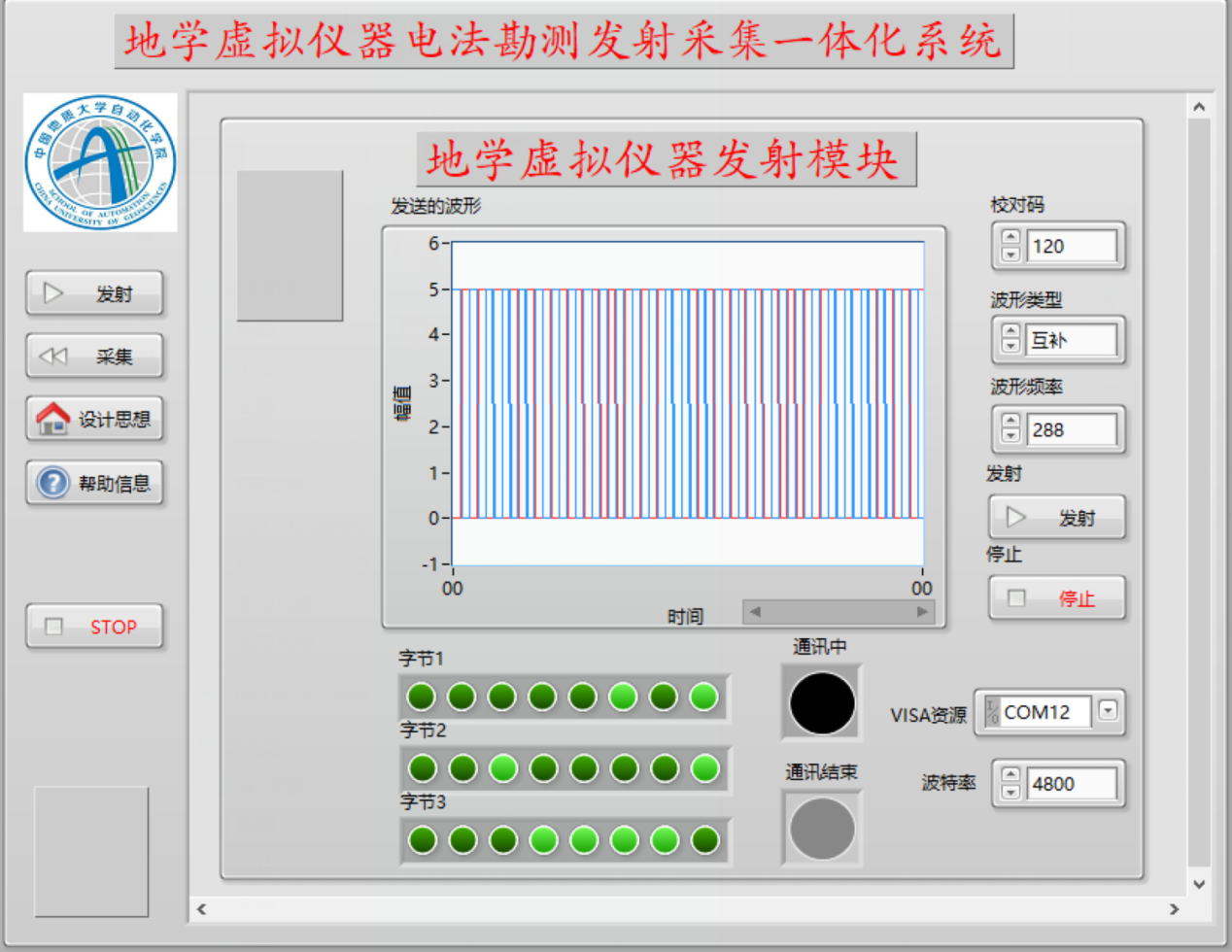


图 4.12 一体化界面

# 单片机程序设计

## 5.1 发射端程序

### 5.1.1 波形产生原理

（1）下位机通信方式

通信方式：异步通信

波特率：4800bps

数据位：8bits

停止位：1bits

校验位：无

1. 上位机通信方式

与上位机之间通过RS232连接通讯

1. 数据

发送数据arry[1]

1. 数据格式

发送数据格式如表5.1所示。

表 5.1 数据发送格式

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 启停 | 波形选择 | 频率（周期）T=（arry[0]&0x3F)\*4\*1000us(arry[0]&0x3F)为0时，T=1000us，同时T最大为0.2s | | | | | |
| 0：启动  1：停止 | 0：单方波输出  1：互补输出 | 0或1 | 0或1 | 0或1 | 0或1 | 0或1 | 0或1 |

### 5.1.2 发射端程序解析

当上位机下发指令时，串口接收数据产生中断，将数据存储并将接收数据完 成标志位置 1，然后返回主函数。将数据信息进行解析，包括启停、输出模式、频率（N 次定时器 T0 中断执行一次电平翻转），打开定时器T0后，定时器T0就可以每 1000us 进行一次定时器 T0 中断，在中断里循环计数，当计数 num等于N时就让输出的电平翻转一次。

1. **串口工作方式、定时器初始化**
2. void init\_T0() *//定时器 T0 初始化*
3. {
4. TMOD|=0x01; *//工作方式 1*
5. TH0=0xff; *//1000us 定时 (65536-1000)/256*
6. TL0=0xdb; *// (65536-1000)%256*
7. TR0=0; *//开启 T0 定时器*
8. ET0=0; *//允许 T0 定时器中断*
9. EA=1; *//开启总中断允许*
10. }
11. void init\_sci() *//串口初始化*
12. {
13. SCON=0x50; *//串口工作方式 1*
14. TMOD|=0x20; *//定时器 T1 工作方式 2*
15. PCON=0x80; *//电源寄存器 SMOD=1*
16. TL1=243; *//bps=(2^SMOD/32)\*fosc/[12\*(256-TH1)]*
17. TH1=243; *//bps=4800 fosc=12M*
18. TR1=1; *//启动定时器 1*
19. EA=1; *//打开总中断*
20. ES=1; *//打开串口中断*
21. }
22. **串口接收中断函数**

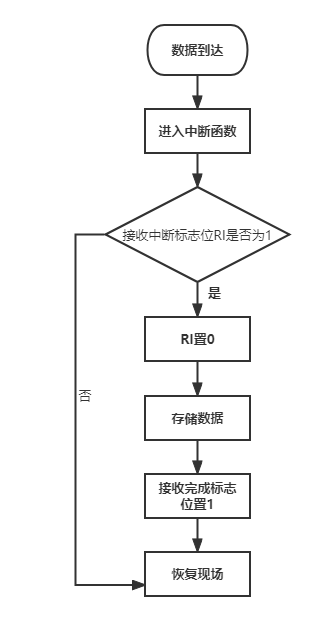


图 5.1 串口接收中断流程图

串口接收中断流程图如图5.1所示。

对标志位进行判断：

1. void sci\_rcv(void) interrupt 4 using 1 *//串口接收中断*
2. {
3. if(RI==1) *//接收中断标志位置 1*

清零接收中断标志位、储存数据并检查校验码是否正确：

1. RI = 0; *//接收中断标志位清0*
2. arry[2]=arry[1];
3. arry[1]=arry[0];
4. arry[0]= SBUF; *//接收数据*
5. if(arry[0]==120)

将接收完成标志位置1，返回主函数：

1. rcv\_flag=1*; //接收完数据*
2. **定时器T0中断函数**

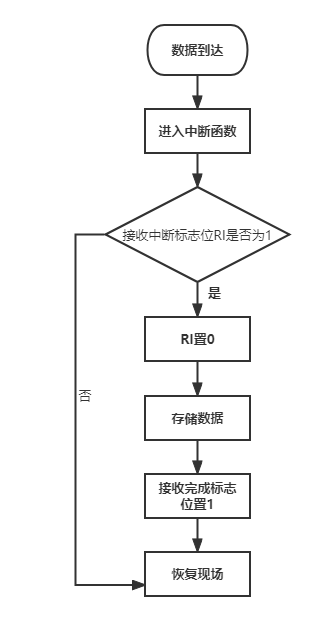


图 5.2 定时器中断函数流程图

num自增计数：

1. void time0(void) interrupt 1 *//定时器 T0 中断*
2. {
3. TH0=0xff; *//1000us 定时*
4. TL0=0xdb;
5. if(timeflag==0)
6. {
7. TH0=0xfe;
8. TL0=0x1a;
9. }
10. num++; *//计数*

通过判断num是否比N大确定是否电平翻转，通过改变N的值可以调节输出波形的频率。

1. if(num>=N) *//经过 N 执行一次*
2. {
3. if(out\_mode==0) *//输出模式 0 时*
4. {
5. if(Output0==1)
6. {
7. Output0=0; *//电平翻转*
8. }
9. else
10. Output0=1; *//电平翻转*
11. }
12. if(out\_mode==1) *//输出模式 0 时*
13. {
14. unsigned char i=0; *//延时计数*
15. if(Output0==0)
16. {
17. Output1=0;
18. for (i=0; i<5; i++); *//延时*
19. Output0=1; *//电平翻转*
20. }
21. else
22. {
23. Output0=0; *//电平翻转*
24. for (i=0; i<5; i++); *//延时*
25. Output1=1;
26. }
27. }

将计数器num清零等待下一次中断。

1. num=0*; //计数清零*
2. **主程序**

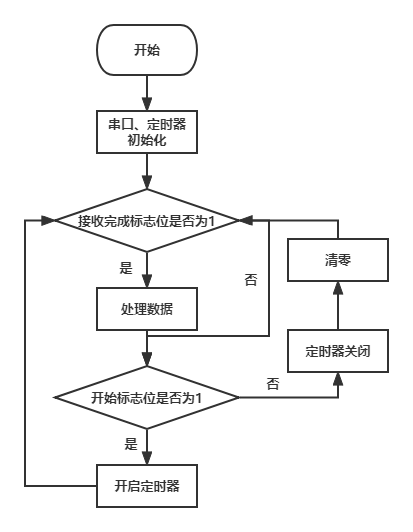


图 5.3 发射端主函数流程图

在 while（1）循环中对数据进行修改，接受标志位判断为 1 则处理数据，处理完成将该标志位置零。

开始标志位为 1 则打开定时器，为 0 则关闭并清零计数值：

1. if(start==1) *//开启时*
2. {
3. TR0=1; *//开启 T0 定时器*
4. ET0=1; *//允许 T0 定时器中断*
5. }
6. if(start==0)
7. {
8. TR0=0; *//关闭 T0 定时器*
9. ET0=0; *//关闭 T0 定时器中断*
10. rcv\_flag=0; *//标志位清零*
11. N=0; *//计数清零*
12. }

## 5.2 采集部分程序

### 5.2.1 采集原理

（1）下位机通信方式

通信方式：异步通信

波特率：4800bps

数据位：8bits

停止位：1bits

校验位：无

（2）上位机通信方式

与上位机之间通过RS232连接通讯

（3）数据

接收数据date[]

（4）数据格式

date[]为采样得到的ADC值（0x00~0xFF分别对应0~5V）

例如ADC值为0x02，则有电压U=2/256\*5=0.039V

### 5.2.2 采集部分程序解析

**（1）串口、定时器初始化**

1. void init\_T0()
2. {
3. TMOD|=0x01;
4. TH0=0xD8; //1000us定时   (65536-1000)/256
5. TL0=0xEF;    //   (65536-1000)%256
6. TR0=1;        //开启T0定时器
7. ET0=1;         //允许T0定时器中断
8. EA=1;           //开启总中断允许
9. }
10. void init\_sci()
11. {
12. CTL = mode;
13. SCON=0x50;
14. TMOD|=0x20;
15. PCON=0x80;
16. TL1=243;
17. TH1=243;
18. TR1=1;
19. EA=1;
20. ES=1;
21. }
22. **定时器T0中断**

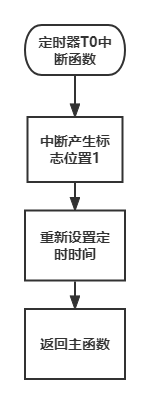


图 5.4 接收端定时器流程图

定时器T0中断函数中将中断标志位置1，并重新设置定时时间。

1. void time0(void) interrupt 1
2. {
3. it\_flag=1;
4. TH0=0xD8; *//150us定时   (65536-1000)/256*
5. TL0=0xEF;    *//   (65536-1000)%256*
6. }
7. **主函数**

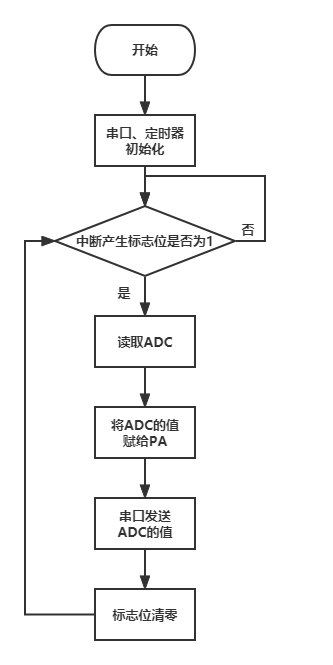


图 5.5 接收端主函数流程图

在while循环中等待定时器T0产生中断并将标志位置1，当标志位置1时就将ADC数据进行采集上传。