

# 目录

一 实习目的与分工.....	1
1.1 实习目的.....	1
1.2 分工说明.....	1
二 地学探测原理.....	2
2.1 地学虚拟仪器方法.....	2
三 系统设计.....	7
四 硬件设计（负责 90%）.....	9
4.1 设计概述.....	9
4.2 嵌入式部分主控制器.....	9
4.3 信号发射部分设计.....	11
4.4 差分放大电路信号采集部分.....	13
五 软件设计（负责 80%）.....	15
5.1 设计概述.....	15
5.2 运行流程.....	16
5.3 发射信号部分.....	16
5.4 信号采集部分.....	18
5.5 通讯部分.....	19
5.6 显示屏部分.....	20
六 LabVIEW 设计.....	21
6.1 通讯协议.....	21
6.2 Laview 人机界面设计.....	23
6.3 发射部分 Labview 设计.....	24
6.4 接收部分 Labview 设计.....	28
6.5 Labview 部分测试.....	31
七 整体调试.....	34
八 实习心得与体会.....	35
九 附录.....	36
9.1 单片机功能板硬件设计.....	36
9.2 硬件电路板（发射与接收）.....	37
9.3 软件部分核心代码.....	39

## 一 实习目的与分工

### 1.1 实习目的

本次的实习，是依据本学期的课程-虚拟仪器原理基础上进行的实践，包含了各个环节，最终是一个完整的地学勘探器件，旨在提高学生的动手实践能力，巩固和加强课堂教学效果，帮助学生了解嵌入式系统的应用知识，培养初步的应用能力分析了嵌入式系统和利用 LabVIEW 应用能力的嵌入式系统，本次实习用到的一个软件 Labview，由美国 NI 公司研发，与原有的代码编程风格不同，Labview 采用的编程方法是一种更为简单的图形编辑语言 G 来进行程序的编写，他的逻辑就会更加的清晰，利用程序框图来进行程序的表示，但是同样的，因为这种语言本身比较繁杂，对于接口和复杂功能的实现编程就会变得比较麻烦。

本次实习的目标是设计出一个地学探测的器件，主要包含上位机发送与接收信号，单片机产生波形与数据处理 A/D 转换，以及全桥驱动和放大电路进行信号处理，最后的实地测试，根据本次实习包含的几个部分，我们可以看到，通过本课程的实习，我们应学会配置 LabVIEW 实验平台和使用硬件开发设备，掌握嵌入式系统采样的工作原理和工程开发过程；学习简单的单片机编程；学习使用 LabVIEW 实验平台形成虚拟现实仪器课程设计的主要目的是使学生通过实践，深刻理解和掌握课堂教学内容的重点、难点和基本概念，并解决项目实施过程中遇到的难题学习操作 LabVIEW 实验平台，使学生对嵌入式系统课程和地学虚拟仪器软硬件设计有一个全面的概念，熟悉常用的嵌入式系统和地质虚拟仪器设备的设计方法和使用方法，掌握嵌入式系统和研究虚拟仪器软硬件设计原理和开发的基本技能，具有初步的系统设计和实现能力。

### 1.2 分工说明

本次实习以分工形式进行，主要由硬件部分，软件部分，LabVIEW 部分三部分组成。分工如下：

- 硬件部分：郑一楠（90%）殷帆（10%）
- 软件部分：郑一楠（80%）殷帆（20%）
- LabVIEW：闫浩鹏（80%）殷帆（20%），LabVIEW 整体工作量相对较大

## 二 地学探测原理

### 2.1 地学虚拟仪器方法

#### 2.1.1 电法勘探

##### (1) 电法勘探原理

以岩石、矿石的电学性质（如导电性、电化学反应（激发极化特性）、介电性和导磁性）的差异为基础，使用专用的仪器设备，观测和研究与这些电性差异有关的（天然或人工）电场或电磁场的变化和分布规律，进而查明地下地质构造及有用矿产的一类地球物理勘探方法。

电法勘探具有效率高、成本低、适用性广等优点，各种电法勘探方法是适应不同地质任务的需要而发展起来的。它们广泛地应用于各种地质工作中，不仅可以寻找金属及非金属矿产，还可以进行地质填图，查明地下地质构造、寻找油气田、煤田和地下水等。此外，电法勘探还用于地壳及上地幔的研究之中。近年来，一些建立在电法勘探基本原理基础之上的新方法如管线探测、探地雷达等广泛用于城市工程勘查，它们在管线勘查、路基、高层建筑地基及大型水电站、水库坝基勘查方面发挥了重要作用。

##### (2) 电法测量基本结构

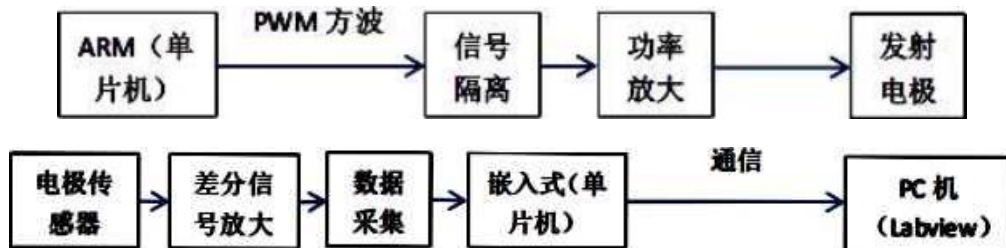


图 2.1 电法测量基本结构

##### (3) 电法测量基本原理

在实际工作中，通过两个供电电极向地下供电，另外两个测量电极测量电位。如图 2.1 所示，在 AB 两点插入信号源（一定频率或直流）产生激励信号，假设电流为  $I$ ，给电流在 AB 两点之间会产生电场，并在地下产生磁场。地下介质在磁场作用下会发生极化现象。如果地下介质属于高电阻率，极化不明显，AB 两点间的电场几乎是半圆形的，相应的磁场垂直与电场均匀分布。当介质中出现低阻时，低阻介质在磁场作用下就会发生极化产生涡流，涡流反过来影响磁场的分布。

在表明插入电极 MN, MN 两点之间的电源随着介质电阻率的不同而发生变化。通过反演探测的数据，就可以获取介质的电阻率。

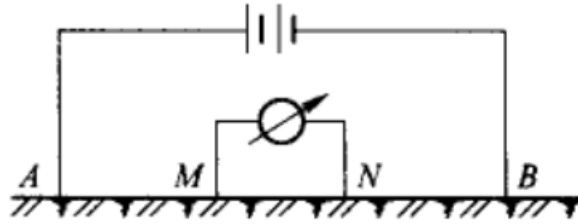


图 2.2 电法勘测原理图

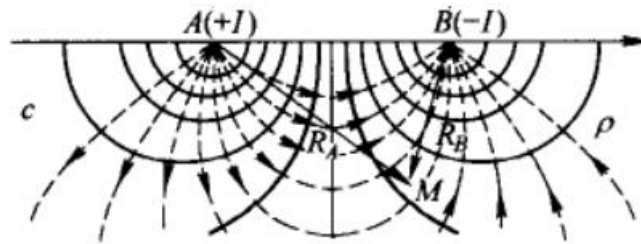


图 2.3 电法电场和电流分布

根据稳定电流场的基本性质，在地下岩石导电性分布不均匀的情况下，自供电电极 A 供入地中的电流总是趋向于沿着电阻较小、路程较近的路径流向电极 B；电流线间彼此互相排斥，因此电流线不会全部集中于良导体内，也不会所有电流线都选择同一条最短路径，而是呈现分布状态。

视电阻率的定性分析公式：

$$\rho_s = \frac{J_{MN}}{J_0} \rho_{MN}$$

MN 距离相对于岩石很小，可近似认为 MN 范围内电流密度  $J_{MN}$  和电阻率  $\rho_{MN}$  不变， $J_0$  为岩石均匀时 MN 的电流密度。该式是视电阻率表达式的微分形式，它将视电阻率与电流密度的分布联系起来，可用于对视电阻率数据的定性分析。

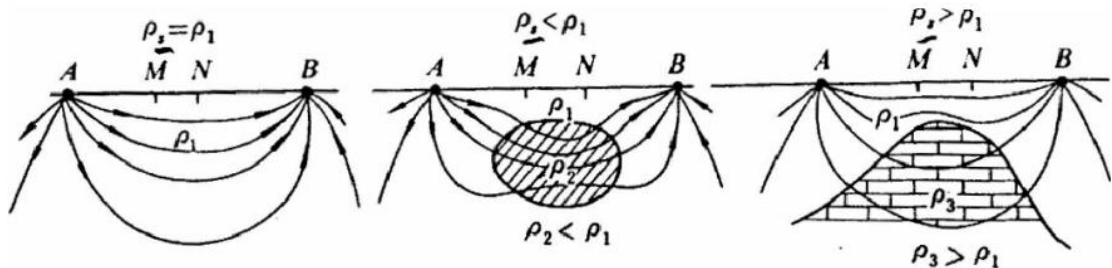


图 2.4 不同电阻率时的电流分布示意图

沿地表剖面逐点观测视电阻率，所获得的  $\rho_s$  剖面曲线的变化能清楚地反映出地下导电性不均匀体的位置及电阻率的相对高低。

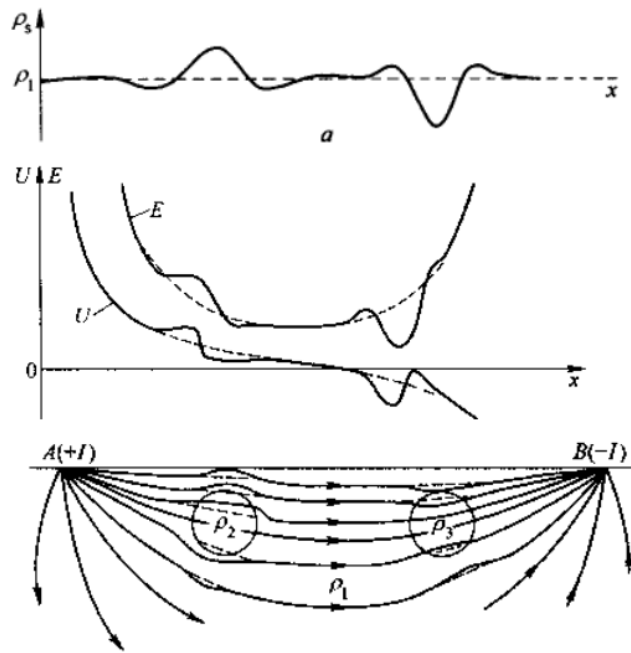


图 2.5 局部不均匀体对正常电流场的扰动及其产生的视电阻率异常示意图

#### (4) 基本要求

- 供电电流要求：地面电阻率法仪器一般是集供电和测量功能为一体的。就供电来说，要求供电电流稳定，供电电流大小可以在一定范围内调节；
- 分辨率好，灵敏度高。在待测的视电阻率和电极装置大小一定时，仪器分辨率越好、灵敏度越高，可以准确测量的电位差就越小，需要的供电电流就越小。这有利于减轻电源、供电电极和供电导线等装备的重量，提高工作效率；
- 稳定性好。要求仪器能够适应潮湿和温度大幅度变化等比较恶劣的野外工作条件；
- 抗干扰能力强。野外存在各种天然和人文的干扰信号，要求仪器对各种干扰信号有很强的压制能力。在我国，特别要求能够很好地压制 50Hz 的工频干扰信号；
- 输入阻抗高。要保证仪器在测量电极接地条件不太好、接地电阻比较大时仍然能够获得准确的测量结果。

### 2.1.2 瞬变电磁法

#### (1) 瞬变电磁法原理

瞬变电磁法 (Transient electromagnetic methods, TEM) 又称时间域电磁法，是一种建立在电磁感应原理基础上的时间域人工源电磁探测方法。它利用不接地回线 (磁源) 或接地线源 (电偶源) 向地下发送一次脉冲磁场 (通常称为一次场)，在其激发下，地下地质体中激励起的感应涡流将产生随时间变化的感应电磁场

（通常称为二次场）。由于二次场包含有地下地质体丰富的地电信息，在一次脉冲磁场的间歇期间，利用线圈或接地电极观测二次场（或称响应场），通过对这些信息的提取和分析，从而达到探测地下地质体的目的。

### （2）瞬变电磁法基本结构

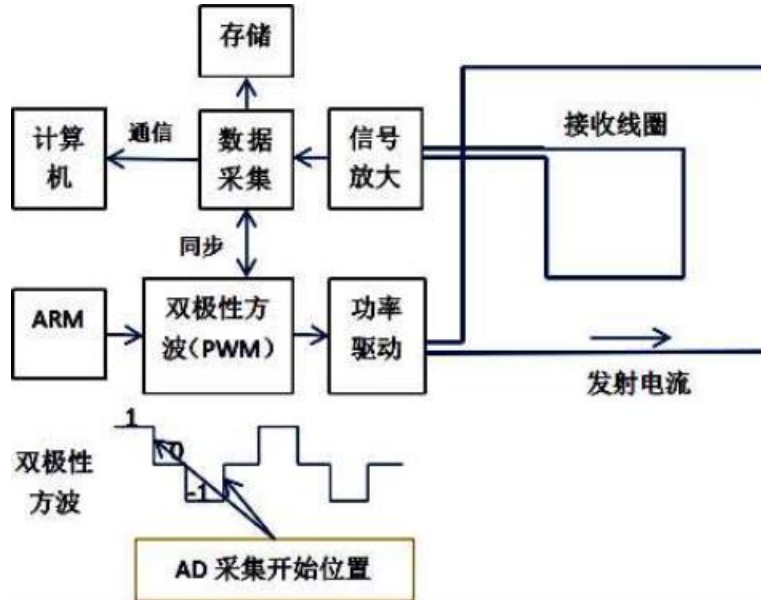


图 2.6 瞬变电磁法基本结构

### （3）瞬变电磁法基本原理

频率域电磁法（FEM）也是建立在电磁感应原理基础上观测研究响应场的方法，它研究的响应场与频率的关系。TEM 和 FEM 的机理没有本质的不同，两者都是通过傅里叶变换关系相互关联，在某些条件下，一种方法的数据可以转换为另一种方法的数据。然而，就一次场对观测结果的影响而言，两者却截然不同，FEM 研究在一次场背景上的二次场，地形、工作装置参数改变、地表层电阻率不均匀及一次场背景值等都会对观测结果带来不可忽略的影响；TEM 则研究一次脉冲磁场的间歇期间的二次场，具有更高的探测和分辨能力。瞬变电磁法是利用不接地回线向下发射由阶跃波形电磁脉冲激发的一次场，在一次脉冲电磁场的间歇期间（断电），会产生一个二次场的涡流场，这个涡流场会随着时间的变化，并且有着一定的规律，包含着与地质体有关的信息，在实际观测中，常通过接收线圈观测一次场脉冲间歇期间的二次场随时间的衰减变化，如下图 2.8 所示。

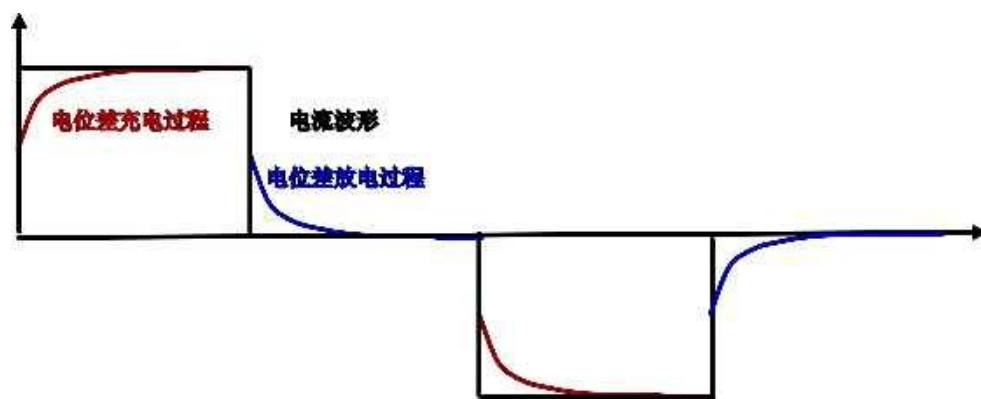


图 2.7 瞬变电磁探测双极性波形

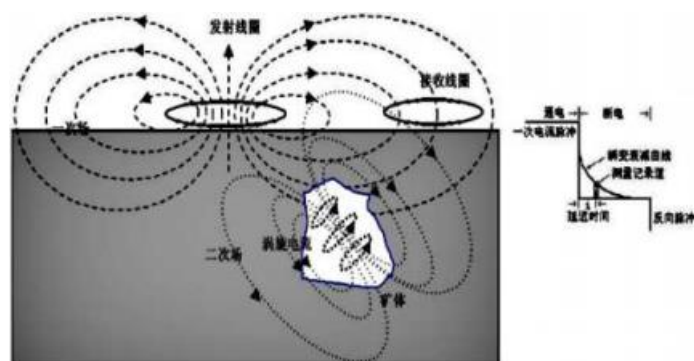


图 2.8 瞬变电磁法勘探原理图

当发射回线中的稳定电流被切断后，根据电磁感应理论，发射回线周围会因为中断电流的突然变化而产生磁场，通常称之为一次场。在一次场的激发下，地下地质体将激励产生大小依赖于地质目标体导电程度的感应涡流，并在空间产生随时间变化的感应涡流电磁场，通常称为二次场。感应涡流在高导率的媒介中扩散速度慢，在低导率媒介中扩散速度快，所以根据这一原理，发射回线的二次场必然会含有地下地质体的相关信息。在实际勘探中，在地表对二次磁场的相关数据进行采集，然后对其进行分析与资料解释，从而获取被测大地的电性信息。

### 三 系统设计

根据教学大纲要求,《地学虚拟仪器实习》内容主要分四部分:

- 基于 LabVIEW 的软件功能模块练习;
- 仪器功能模块(硬件);
- 仪器功能模块设计(软件);
- 地学虚拟仪器构建。

在这四个部分中,首先完成 LabVIEW 功能模块的设计,然后完成仪器模块的设计,包括 ARM(或 MCU 编程)、硬件设计和制作,为构建地理虚拟仪器提供条件;在此基础上,完成了地学虚拟仪器系统的构建。在此次的实习中,我负责的部分主要为仪器功能模块的硬件部分的设计,由于决定使用自行开发的印刷电路板完成实验,伴随着实习的开始,硬件部分的设计与制作也同步的第一时间开始,作为嵌入式平台以及各项功能的载体,硬件的搭建是其他部分得以实物测试以及正式测量的前提,因此硬件部分的制作时间较前,整体周期为实习的第一周,后两周则是各功能模块的调试与不断的小调整与完善。

地学探测信号一般是微弱信号。在时域检测信号中,有效信号持续时间很短,衰减很快。因此,在地测仪器的设计中,对数据采集的精度和速度提出了很高的要求。

在 LabVIEW 设计和数据采集的基础上,根据地学探测原理和相应的传感器,设计了基本的地学探测虚拟仪器设计的。因为目前的地学探测仪器几乎都采用有源探测,所以在本次设计中,除了数据采集外,还需要设计有源信号源和相应的功率驱动电路。

在虚拟仪器的设计研究中,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,场景中的各种参数)连接起来(可以是物理量,也可以是化学量,生物量,等)通过各种传感器和发射机进行适当的转换后,经过信号调理、采样、量化、编码、传输等步骤,最后到处理器进行数据处理、存储记录和显示打印过程。它是监测过程中获取原始数据的主要手段,控制和管理一个系统。用于数据采集的一整套设备称为数据采集系统。所以我们要设计三个部分:

- 可以控制发射信号和计算采样数据的 MCU;
- 可以发射和接收信号的硬件电路;
- 可以控制单片机并且对单片机的采集进行读取的 LabVIEW 上位机。

整体设计模式图如下:



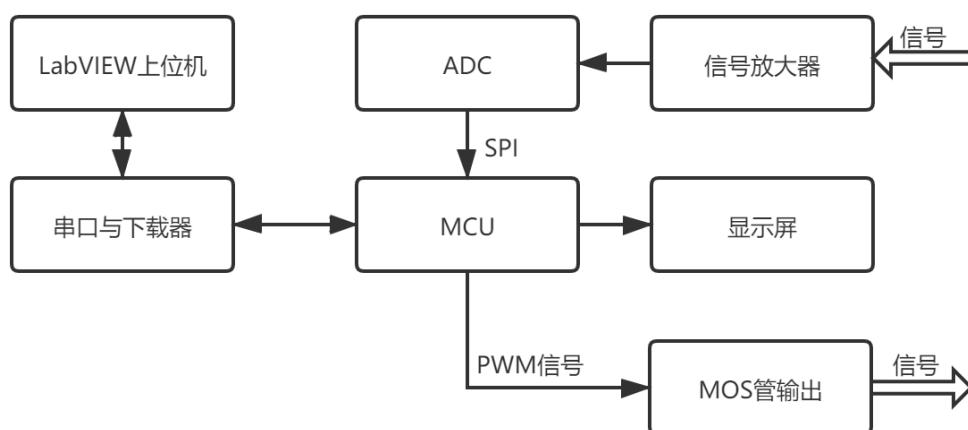


图 3.1 总设计模式图

我们可以看出，MCU 是我们整体设计的核心，此次实习我们采用的主控为 STM32F103C8T6，采集系统通过信号放大器进入 ADC 后传给 MCU，MCU 将数据通过串口通信至上位机软件之中，构成采集系统。在发射系统方面上位机向 MCU 发射需要的信号类型，MCU 处理后利用 PWM 信号驱动 MOS 管输出信号。而显示屏作为发射系统和采集系统的重要信息的显示作用。

## 四 硬件设计（负责 90%）

### 4.1 设计概述

硬件平台部分的搭建主要包括全桥驱动信号功率放大电路，差分输入的采集信号放大部分以及基于 STM32F103C8T6 的数据传送以及通讯部分。推荐使用基于 IR2110S 全桥栅极驱动芯片以及 LM324N 四通道运放完成实验的各硬件搭建，使用 STM32F103C8T6 自带的 ADC 将模拟信号转换为数字信号，最终由串口发送至电脑由 LabVIEW 控件显示出来。在实验的初期，由于我们期望能够在系统中增加更多的可以实现的功能，同时使系统更加稳定，外观更加美观。因此我们未使用推荐的试验箱完成实验，我们组自行设计并绘制了印刷电路板并且修改了部分模块组成，最终完成实验，具体硬件部分设计如下。完整硬件设计电路图见附录部分。

### 4.2 嵌入式部分主控制器

伟福 LAB9000 试验箱使用的为基于 MCS51 指令集的 51 单片机，由于 51 系列单片机问世时间较早，虽使用依然十分广泛，但由于其较低的主频、较少的外设资源，且总线宽度仅有八位，在对资源需求较多的场景下使用，会略显捉襟见肘。在此次的设计中，我们希望能够达到更高的采样率与通讯速度，因此，使用基于 MCS51 指令集的 51 单片机会使设计较为困难。为了能够达到较高的采样率同时为其他的功能的实现提供可能，同时结合目前单片机行业产品价格上涨的现状，我们决定使用意法半导体公司生产的 STM32F103C8T6 单片机作为嵌入式部分的主 MCU，该单片机基于 ARM Cortex-M3 内核设计，主控实物图如图 4.1 所示：

从图中可以看出，针对单片机部分，我们将其大部分功能的接口引出，设计了一款功能板，能够提供 LED 点亮，蜂鸣器，两路串口输出，四路 PWM 输出，八个多余 IO 引脚（可以提供 ADC 功能），三路输入捕获通道，以及一个控制按键，可以完全覆盖此次实习的所有功能应用，原理图如图 4.2 所示。

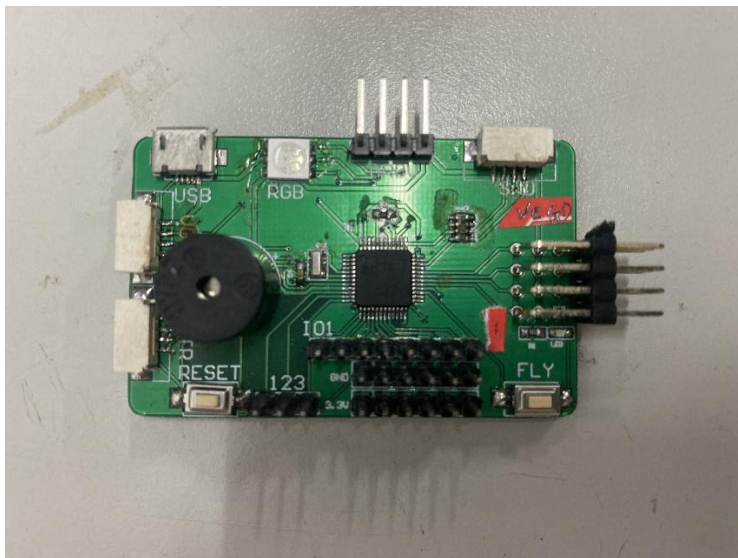


图 4.1 嵌入式部分功能板实物图

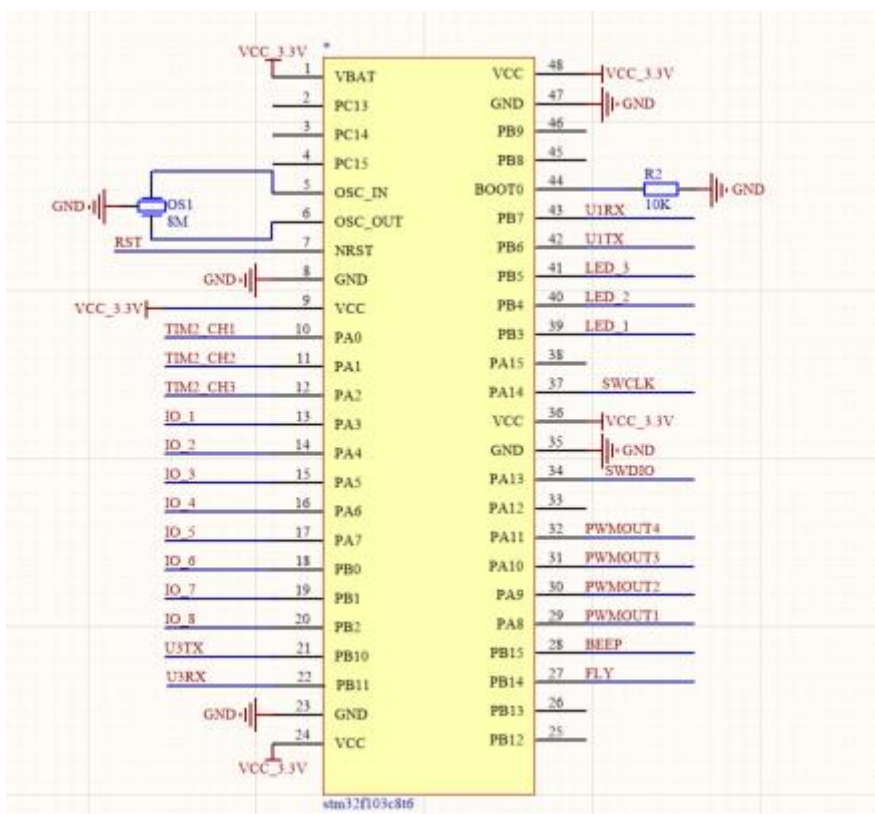


图 4.2 单片机原理图

### 4.3 信号发射部分设计

在本地学虚拟仪器中，采集的信号为自行发射信号在土壤中的变化情况，因此，采集信号的前提是由控制部分产生具有一定功率的信号，并将信号发射到土壤中，该信号通常为直流信号和低频交流信号，由于单片机直接产生的信号通常功率较小且不为交流，此处使用基于 MOSFET 的桥式全桥驱动电路来实现信号功率的放大以及产生交流信号。此部分的发射能量源为 12V 电源适配器，控制部分能量源为 5V 电源适配器。单片机通过信号驱动该部分的栅极驱动器，再由栅极驱动器驱动 MOSFET。MOSFET 在该电路中起的是开关的作用，由于 N 沟道的 MOSFET 在导通时往往具有极低的内阻且价格较低，栅极驱动器通过驱动 MOSFET 改变输出端与 12V 电源以及公共点地的连接情况便可以实现较大的功率输出，且具有较低的成本。驱动部分整体原理图如图所示：

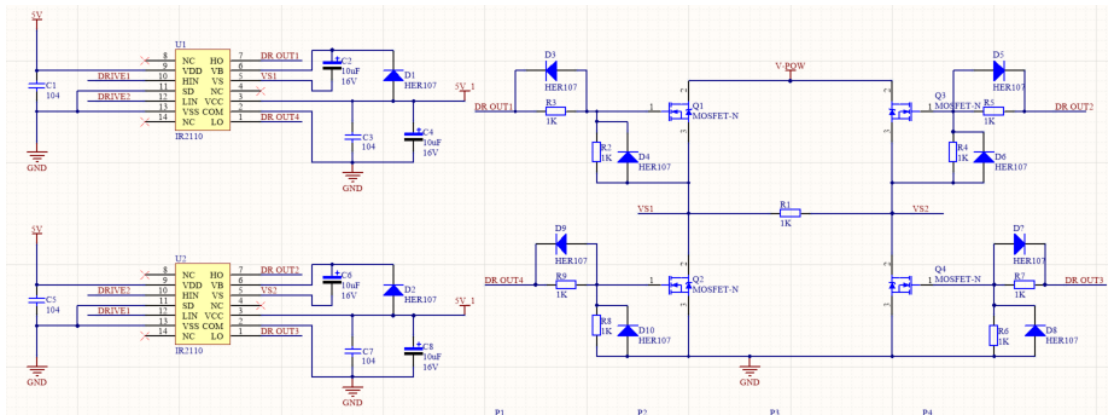


图 4.3 驱动部分整体原理图

此次的虚拟地学仪器中使用的栅极驱动器为英飞凌公司生产的 IR2110S 栅极驱动器，其具有最高可达 500V 的运行电压，用于 IGBT 和 MOSFET 的驱动，具有可达 2.5A 的拉电流和 2.5A 的灌电流，也具有专为自举操作设计的浮动通道，栅极的驱动能力在 10 至 20V，具有双通道的欠压锁定以及 3 至 20V 的逻辑供电范围，非常的适合此处虚拟地学仪器信号发射的应用场景。其通常为 16 脚的 SOIC 封装，但是为了方便焊接，此处使用的为 14 脚的 PDIP 封装，不同的封装之间引脚数不同，且引脚定义略有不同，但引脚功能完全相同，多余引脚均为空引脚。

如图 4.4 所示，IR2110S 在工作时与两片 MOSFET 共同构成一个驱动的半桥，两个驱动的半桥便可以构成一个完整的全桥驱动，其中 HIN 管脚控制 HO 输出管脚，控制高边的 MOSFET，LIN 管脚控制 LO 输出管脚，控制低边 MOSFET，即当 HIN 管脚为高电平时，高边 MOSFET 导通，LIN 管脚为高电平时，低边 MOSFET 导通，控制全桥不同方向的高边和低边的 MOSFET 同时导通，即可以实现电流方向的正反改变，实现交流的产生。值得注意的是，通向的高边和低边

MOSFET 一定不能同时导通，否则会引起电源的短路。

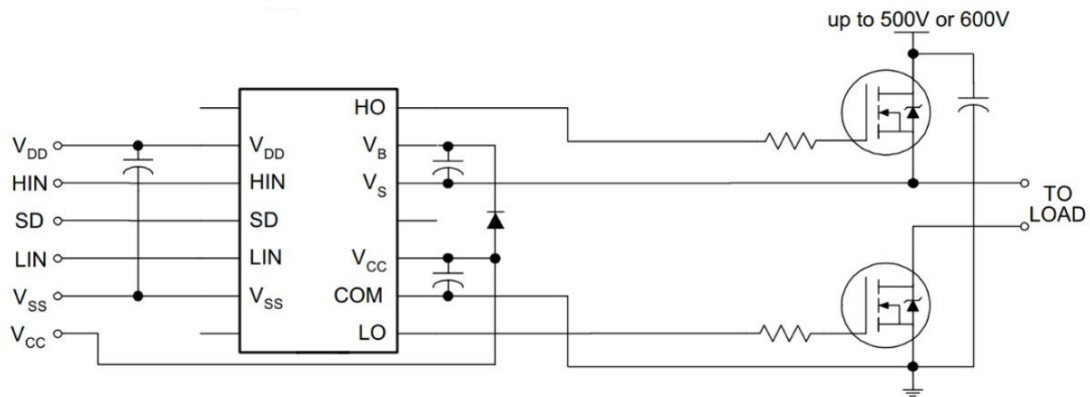


图 4.4 IR2110S 驱动原理图

在高边 MOSFET 导通的同时，会导致高边 MOSFET 的源极电压被抬升，使得该 MOSFET 的栅极与源极之间的电压降低，MOSFET 便无法再继续导通，为解决这个问题，IR2110S 具有浮动通道，在高边 MOSFET 驱动的供电极与高边 MOSFET 的源极之间连接一个电容，并通过二极管给电容以及供电极供电，当高边不导通时，通过二极管给电容充电，当电容充满时，电容两端电压等于电源电压。当高边导通时，由于电容两边的电压不能突变，当源极的电压升高时，电容另一端的电压也会随之升高，使电压差保持相对稳定，此时，便是使用电容所存储的能量来驱动栅极，使栅极与源极之间的电压始终保持大致为电源电压，使 MOSFET 保持导通。焊接完成的成品如下图所示：

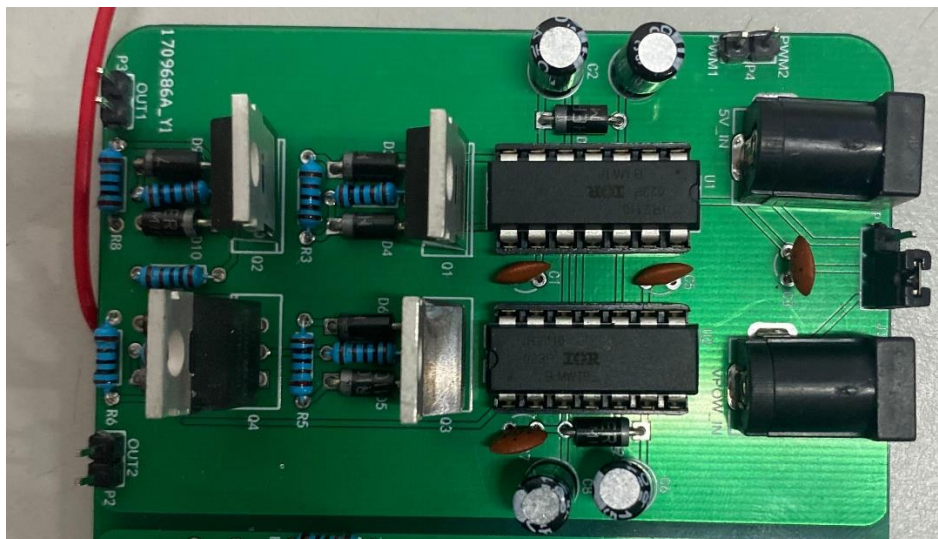


图 4.5 焊接实物图

在焊接过程中，为了保证焊接的质量，在设计好走向位置并且完成焊接后，需要使用万用表检测每根走线的导通情况，防止因为焊接失误造成的驱动不工作。同时，由于 MOSFET 自身的特性，需要先提供控制部分的供电再给 MOSFET 供



电，防止栅极悬空造成的 MOSFET 自开启从而造成短路。在设计时值得注意的是，由于 IR2110S 具有欠压保护的设计，因此在选择驱动供电时，需选择 10V 以上的电源供电，防止因为欠压保护造成驱动器的不工作。驱动部分测试图如下：

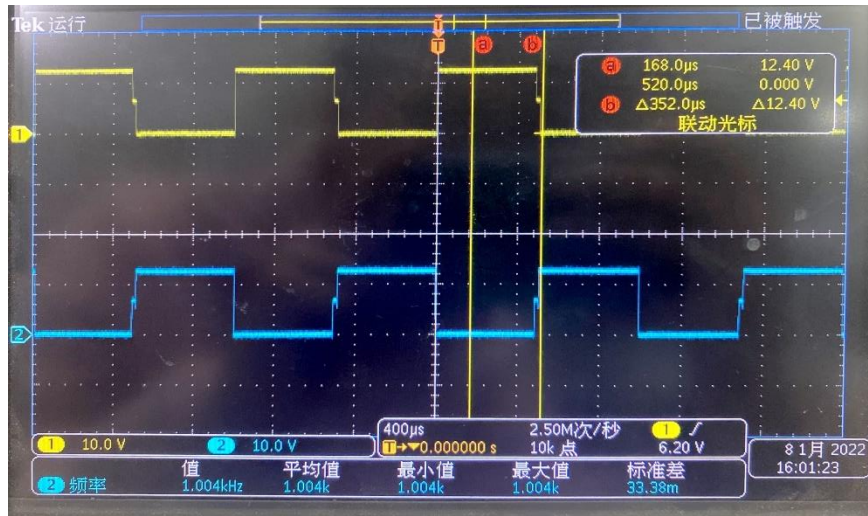


图 4.6 驱动部分测试图

#### 4.4 差分放大电路信号采集部分

此次的虚拟地学仪器，所使用的电法探测、磁法探测以及电磁探测等，在测量层面，都是对大地中微弱电信号的检测，单片机以及模数转换器无法直接检测如此微弱的电信号，同时，该微弱信号中还存在较强的共模干扰，因此，在信号的调理过程中需要使用差分输入的仪表运算放大器，放大差模微小信号，同时屏蔽共模干扰的影响，将放大后的差模信号再送入模数转换器，实现模拟信号到数字信号的转换。在本设计中，为降低成本，使用具有四通道的通运运算放大器 LM324N 构成差分输入的运算放大器。使用两个运算放大器构成信号的差分输入与屏蔽共模干扰，增益倍数可调，在该差分输入的后级再使用一个运算放大器搭建反向运算放大器，实现固定倍数的二级放大。值得注意的是，由于模数转换芯片的电压范围为 0 至 5V，因此需要对放大得到的幅值范围为正负 2.5V 的电压进行电压抬升，使其满足模数转换的电压输入范围，使用电阻分压，得到 2.5V 的基准参考电压，使用四通道运放的最后一个通道，构建电压跟随器，降低该参考电压的输出阻抗，使得其更加稳定，将之前的第二级运算放大器的参考电压点选择为该 2.5V，即可实现电压的抬升，满足模数转换器的输入要求。

在该电路中，由于使用的运算放大器需要使用双电源供电，因此需要在正电压的基础上产生一个负电压，为运算放大器供电。此设计中，使用的为 MAX660 电压转换芯片，该芯片使用电荷泵的原理，将原本为正 12V 的电压转化为负 12V，

使用该正负 12V 的电源，供给运算放大器使用。完整电路原理图如图所示：

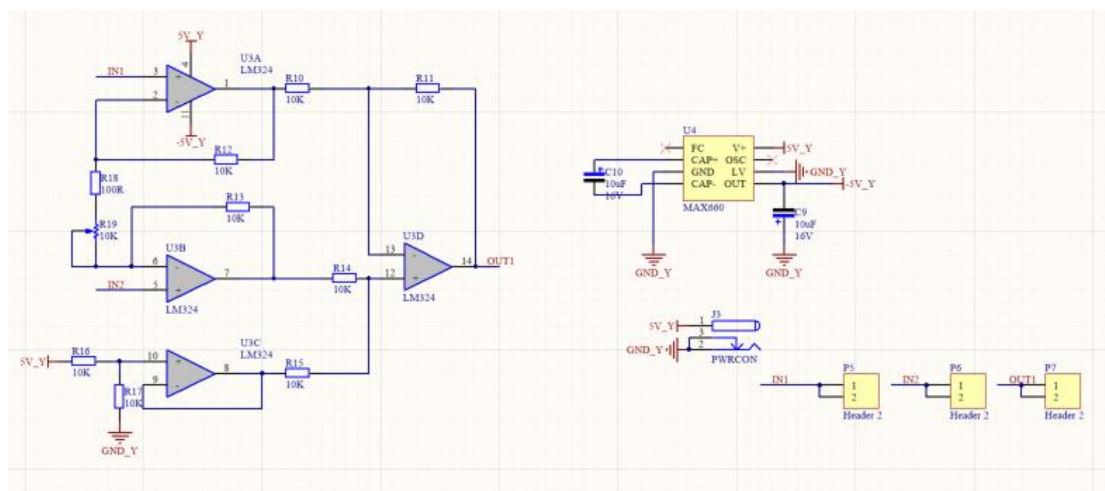


图 4.7 放大部分完整原理图

经过布局与设计，将其布局并焊接在洞洞板上，并留出相应的输入输出与电源接口，焊接实物图如图所示：



图 4.8 焊接实物图

测试图如图所示：

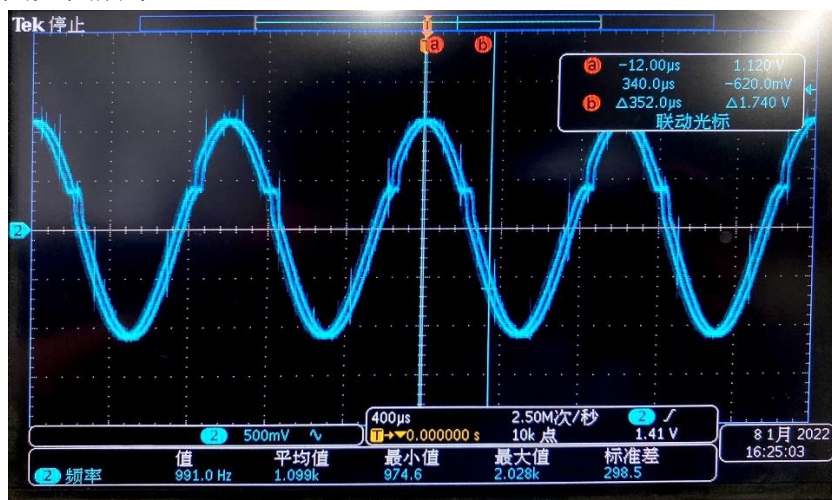


图 4.9 放大部分测试结果图

## 五 软件设计（负责 80%）

### 5.1 设计概述

在软件设计部分，主要是基于单片机功能板的相关开发，此次选用的单片机为 STM32F103C8T6。对单片机部分，我们将其大部分功能的接口引出，设计了一款功能板，能够提供 LED 点亮，蜂鸣器，两路串口输出，四路 PWM 输出，八个多余 IO 引脚（可以提供 ADC 功能），三路输入捕获通道，以及一个控制按键，可以完全覆盖此次实习的所有功能应用。同时采用 HAL 库对其进行初始化编程，其中引脚的设置如图 5.1 所示。

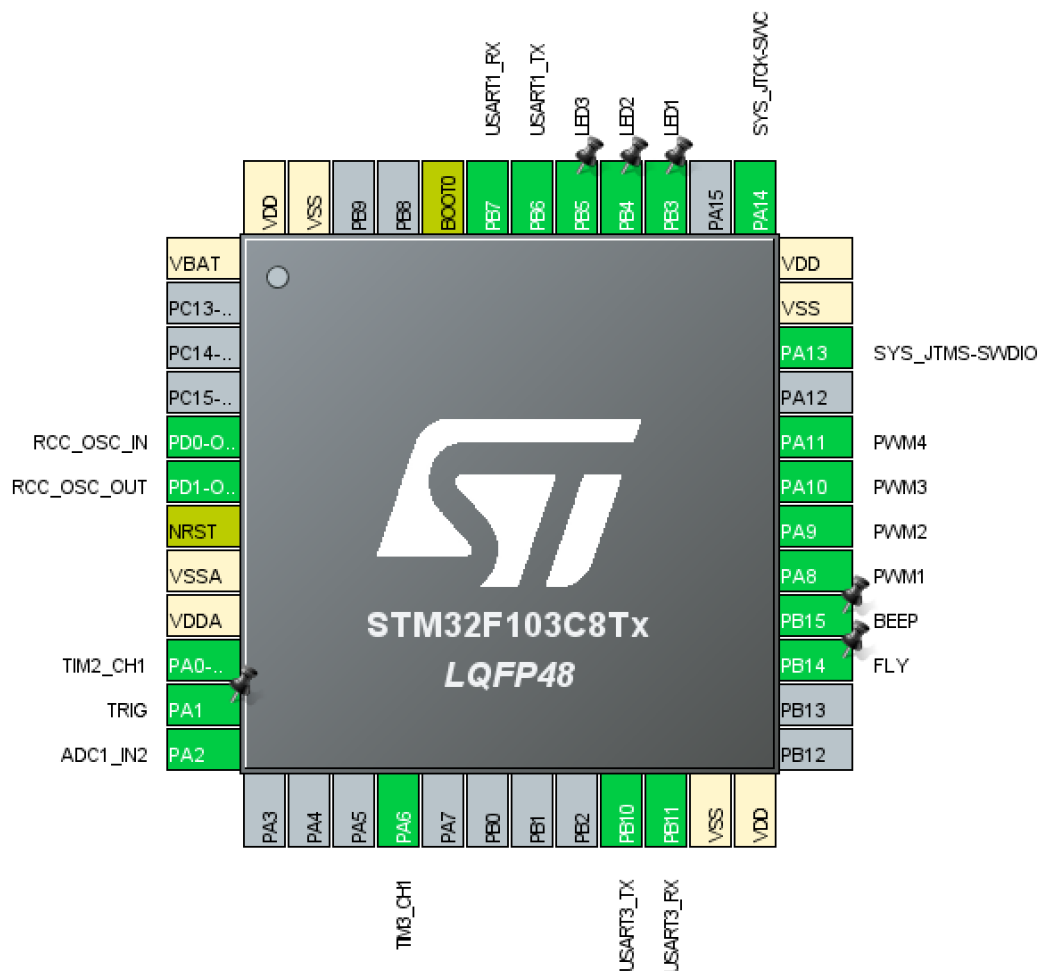


图 5.1 单片机资源配置图



## 5.2 运行流程

本装置的核心工作由主控 MCU 来完成，MCU 需要承担信号发射、信号采集、数据通讯及整体系统运行流程的管理和维护。为了完成地学探测装置的系统工作，我们将软件程序暂且分为如图 5.2 所示几个模块。

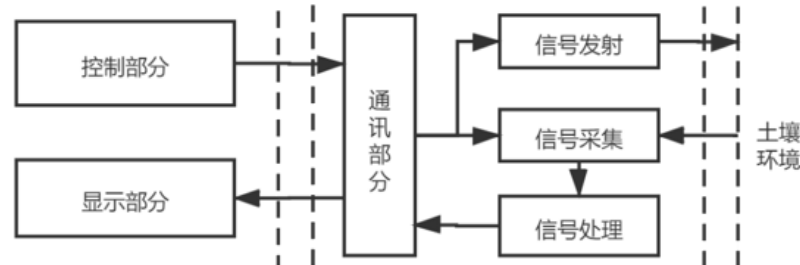


图 5.2 程序模块划分

在整体系统中，由上位机配置所需要的信号发射参数（如波形种类、频率等）以及采集条件（如采样率、采样时间等），下位机多次与上位机进行配对握手以及指令传输后，将开始执行整个地学探测信号发射采集工作。其中该装置能够发出 1-200KHz 频率的 PWM 脉冲信号，受通讯速率及单片机任务周期影响，其能够提供 300Hz 无失真的信号采集（实际采样速率为 1500-3000Hz，满足 5-10 倍工程采样需求）。

系统在运行过程中，上位机可以实时的更改设备所需参数，满足“边调试边检测”的实际使用需求，当信号处理完成后也会按批发送给上位机进行实时显示，方便观察当前采集现象。

## 5.3 发射信号部分

在系统中，信号发射是十分关键的，信号发射的稳定程度，可调节范围情况都将影响到装置对土壤探测的实际效果。考虑到实际使用情况以及设备的工作可靠程度，我们将设备的 PWM 信号发射频率限制在了 1-200KHz 之内，占空比的调节范围则为 0-100%之间，并支持单极性 PWM 信号和双极性 PWM 信号的输出。只需要由上位机配置相应的设定参数，并通过 uart 串口的方式发送给主控 MCU，主控 MCU 会对所发送的指令包进行实时解包并修改信号发送的重配置，信号发射流程图如图 5.3 所示。

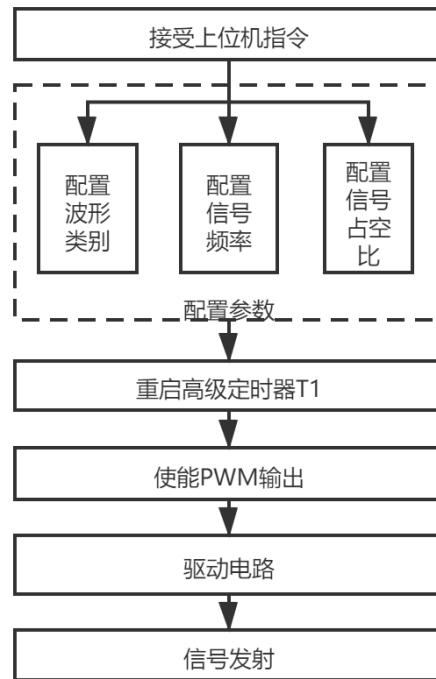


图 5.3 信号发射流程图

在信号配置的过程中，由于实际工作频率采用预分频的方式完成，并非为直接给定预期的工作频率，而占空比也是以比较器的方式来触发完成的，所以关于工作频率及占空比数值的配置存在公式进行换算。

定时器的频率由 MCU 的分频系数决定，由于系统运行时钟配置在了 96M，故当预分频系数设置为 98 时，即可获得 1M 的中断频率，而再根据所需的频率设置一个比较值，即可在特定周期后重置系数，使得系统根据我们所配置的频率而工作。

针对占空比也是采用了同样的方式进行，只是占空比的比较值设定范围则是我们频率所配置的系数范围内，当占空比的比较值等于最大系数时，及占空比为 0，相反则取 100%。

信号发射部分结果可以参考硬件部分。

## 5.4 信号采集部分

在信号采集部分，我们使用的是 STM32F103C8T6 自带的 ADC，首先需要对时钟进行相关配置，从而确定 ADC 的采样频率。时钟树配置如图 5.4 所示。

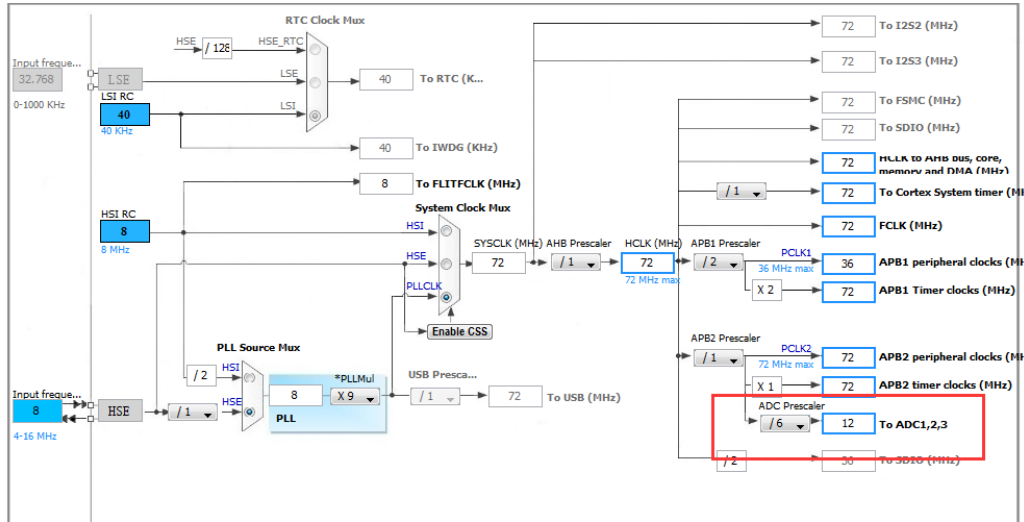


图 5.4 单片机时钟树配置

这个始终可以结合 ADC 设置里配置的采样时间结合算出 ADC 转换得时间，进而计算频率。因为 ADC 总的转换时间和 ADC 的输入时钟以及采样时间有关，所以其计算公式为：

$$T_{CONV} = \text{采样时间} + 12.5 \text{ 个周期}$$

其中 12.5 个周期是采集 12 位 AD 时间是固定的。当  $ADC\_CLK=12\text{MHz}$ ，ADC 为 1.5 个周期，那么总的转换时间为： $T_{CONV} = 1.5 + 12.5 = 14$  个周期  $= 1.167\mu\text{s}$ 。

出了单纯配置 ADC 读取，我们还使用 DMA 用于数据的传输。DMA(直接存储器访问)允许不同速度的硬件装置进行沟通，而不需要依赖于 CPU 的大量中断负载。由于本次实习过程中单片机本身已有许多中断需要进行相关处理，使用 DMA 可以很好的缓解 CPU 压力，相关代码如下：

```
void DMA1_Channel1_IRQHandler(void)
{
    HAL_ADC_Stop_DMA(&hadc1); //停止 DMA 的 ADC 转换，AD_DMA 0~3 对应 ADC 0~3
    HAL_TIM_Base_Stop_IT(&htim3); //关闭定时器
    printf("AD_DMA_0 = %d\r\n", AD_DMA[0]);
    HAL_TIM_Base_Start_IT(&htim3); //重新开启定时器

    /* USER CODE END DMA1_Channel1_IRQn 0 */
    HAL_DMA_IRQHandler(&hdma_adc1);
}
```

## 5.5 通讯部分

通讯部分大部分与 LabVIEW 有关相关流程不在此赘述，在实际设计当中，我们设计 LED 灯的显示用于判断当前的通信状态，有利于进行调试。结果如图 5.5-5.7 所示。

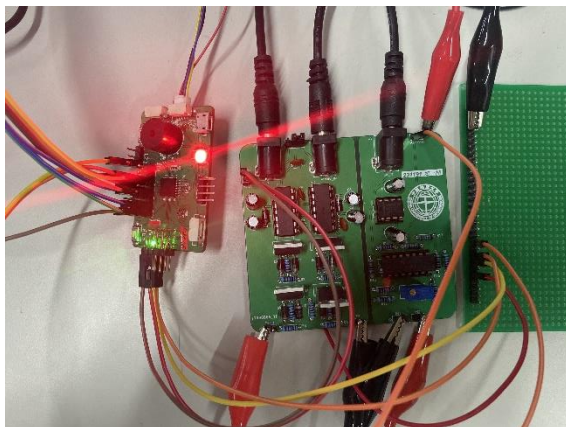


图 5.5 系统未连接显示红灯

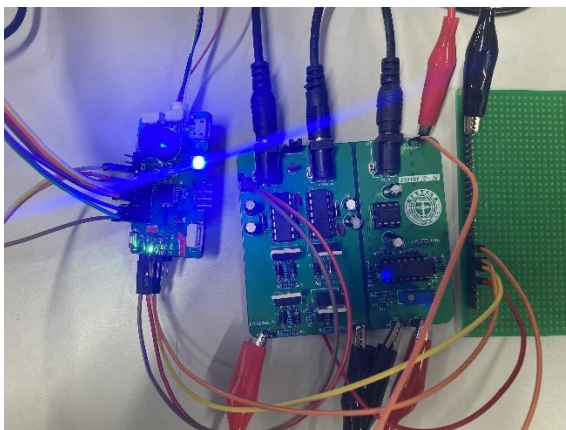


图 5.6 系统第一次握手显示蓝灯

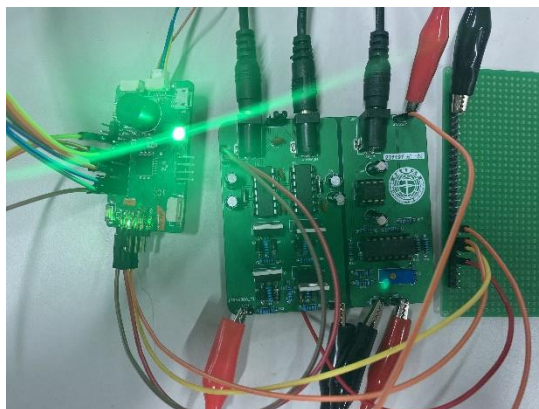


图 5.7 系统握手成功正常通讯，显示绿灯

## 5.6 显示屏部分

在本装置中，我们还额外增加了一个 SPI 显示屏，主要用于辅助显示当前设备运行状态，我们为了能够确认传送数据的正确与否，设立了一个小的 led 显示屏，可以显示我们的波形信息，包含占空比和频率信息，可以确保我们的数据发送没有错误，并且方便我们的调节。



图 5.8 OLED 显示屏实物图

## 六 LabVIEW 设计

### 6.1 通讯协议

在本次实习中，上位机与下位机的通讯协议一共分为三大部分，第一部分是上位机向下位机传输起始控制信号；第二部分是下位机与上位机传输数据信息，以显示结果数据；第三部分是上位机向下位机传输终止控制信号。以上所有的信号传输全部通过串口通信实现。

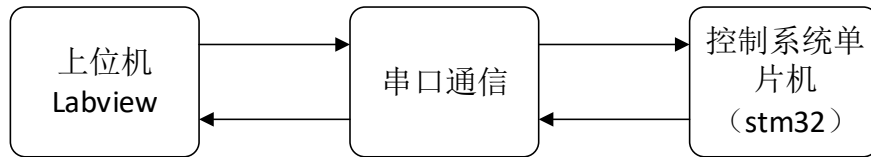


图 6.1 上位机下位机通讯流程图

#### 6.1.1 起始控制信号

我们的起始控制信号主要作用是为了给单片机发送有关波形信号的信息，我们输出的信号信息主要包含占空比、频率、信号类型、头部校验、尾部校验共六个字节，每一帧的包含信息如下图所示：

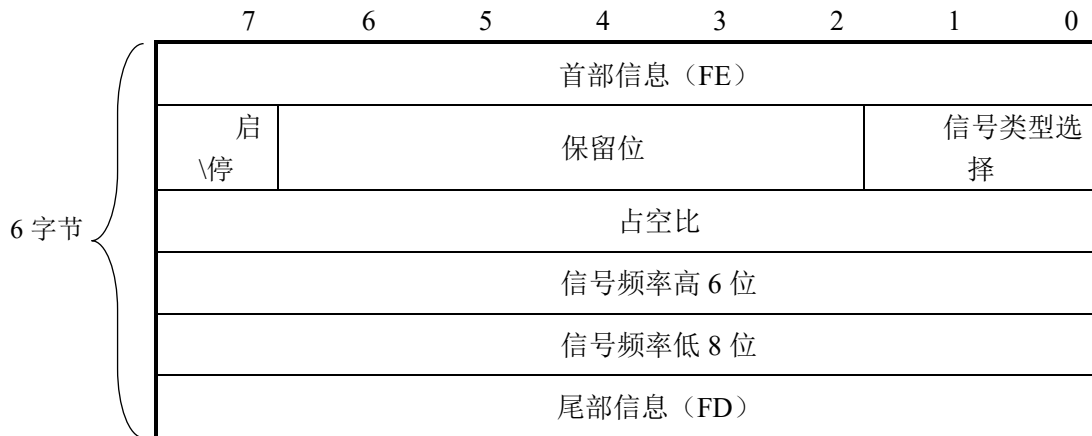


图 6.2 起始控制信号设计

如图所示，起始控制信号共由 6 个字节，42 位数据组成，第一字节是整个起始控制信号的首部，其值设定为 FE。第二字节决定了此时是否需要开始发送波形，与发送波形的类型。当第 7 位的启停标志位为“1”，代表可以开始发射目标波形，反之当启停标志位为“0”，代表终止发射目标波形。当第 0、1 位为“10”时代表所需的发射信号类型为互补信号，当第 0、1 位为“01”时代表所需的发射信号类型为非互补信号。第三个字节包含的信息为占空比，可以让单片机发送任意占空比的方波信号，第四字节和第五字节包含了我们的频率，使用 2 个字节

来传送频率信息，可以让我们频率的包含范围更大，可测量的范围更大，第八字节是整个控制信号的尾部，其值为 FD。

### 6.1.2 传输信号协议

当我们的下位机也就是单片机接收到上位机对信号的相关要求之后，就会控制硬件电路输出对应类型、对应频率、对应占空比的信号，同时单片机也会启动 A/D 转换，将采样的结果传输给上位机，由上位机负责将输出的结果显示在界面上。但是由于考虑到上位机与下位机均无法准确判断出对方目前的状态，即上位机无法确定下位机状态，因此在数据传输信号协议中采用了类似于计算机网络中 TCP 协议建立连接时的三次握手通讯连接确认方式，将其进行了简化并用于本系统。

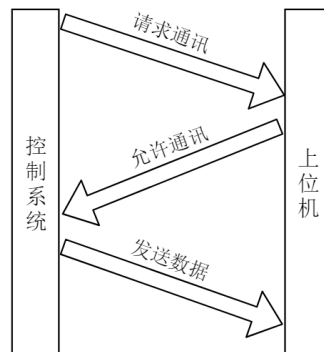
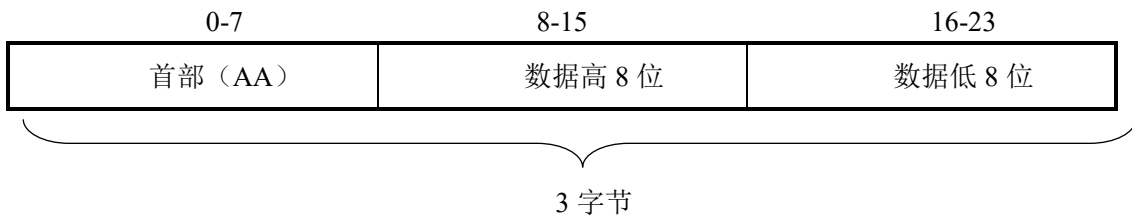


图 6.3 数据传输信号三次握手示意图

单片机在向上位机发送信息时，主要发送的数据包为：



可以看到我们对数据进行接收时，因为数据的连续性，我们对数据进行了一个处理，我们一个帧首之后会加入 50 组数据，总计 100 个字节，每两个字节代表一个数据，这样就可以提高我们的采样精度，让我们可以更仔细地观察测到信号的变化。

### 6.1.3 终止控制信号协议

我们在按下停止按钮时，Labview 就会关闭对数据的读取，此时单片机也会停止数据的发送，直到下一次新的发射信号到达，单片机重新读取信号的设置，然后输出新的 PWM 波。

## 6.2 Laveiw 人机界面设计

按照本次实习的要求，我们要在 Labview 设计出界面，于是我们小组在 Labview 中搭建了一个界面，按照各功能模块的划分，界面大致可以分为：发射部分波形显示（红色框）、接收部分波形显示（褐色框）、过程状态指示（蓝色框）、波形数据设定（橘色框）。



图 6.4 Labview 人机交互界面

发射部分波形显示可以显示出我们发射信号中包含的波形，在此可以看到我们发射信号的波形显示。

接收部分波形显示可以显示出我们接收单片机数据之后形成的波形，我们设立了两个表，一个是原有值，另一个是我们经过换算之后的实际信号，并且我们经过了滤波，每一个数据都是 50 个数据的采样和，这样波形的显示速率变慢，而且我们能够更加明显的看到检测过程中出现的信号波动。

过程状态指示包含了发射和停止按键，我们在发射之后任意时刻可以停止，然后修改波形数据进行再次发送，另外我们设立了几个状态灯，指示现在的连接状况等，并且设立频率显示的灯，可以显示我们当前的发射信号频率。

波形数据设定部分，主要包含了对波形的一些设定，我们在此处可以对波形的占空比、频率、波形类型、串口和数据位数进行设置，并且可以设置我们的接收数据的存储地址。



## 6.3 发射部分 Labview 设计

### 6.3.1 初始化发射信号数据

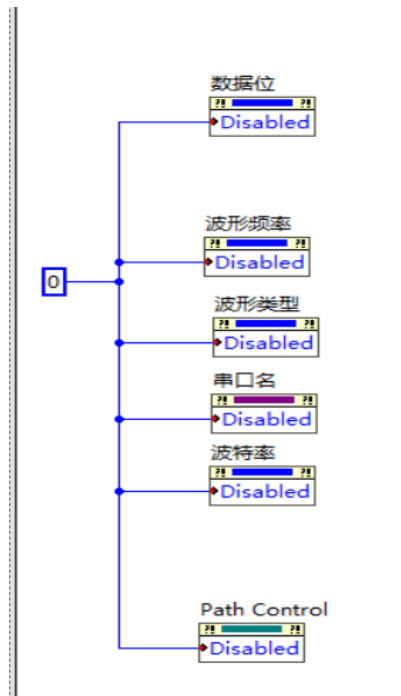


图 6.5 Labview 配置按钮激活

在发射之前，我们将按钮和配置都激活，此时可以对信号的设置进行调整，在此处，属性节点 Disabled 若参数为 0，则可以进行修改，若为 1，则隐藏并且不能修改，若为 2，则显示为半透明状态，不可修改。

### 6.3.2 信号发射部分

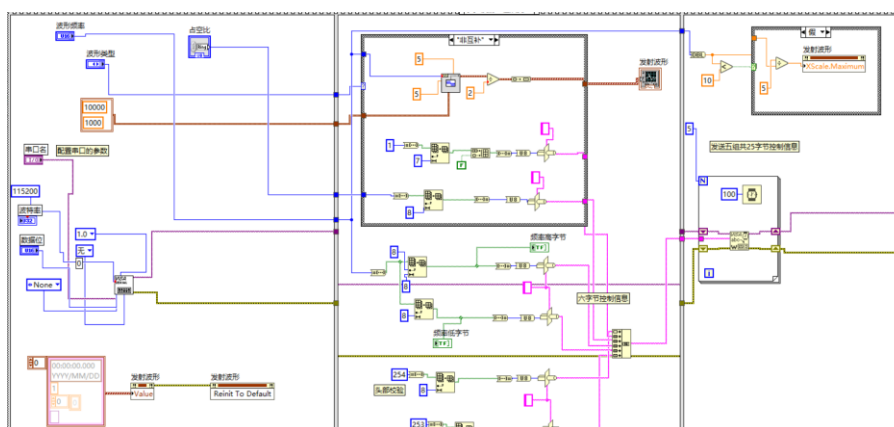


图 6.6 信号发射部分

上图是我们对发射部分的一个总览，可以看到，我们对波形的各项设置，通

过我们的互补和非互补框图，可以产生对应的波形数据，然后将上述提到的发射部分六个字节信息发送给下位机单片机，防止数据出现错误，我们发送五组，可以供单片机进行检测。

### 6.3.3 发射按下时对波形显示的处理

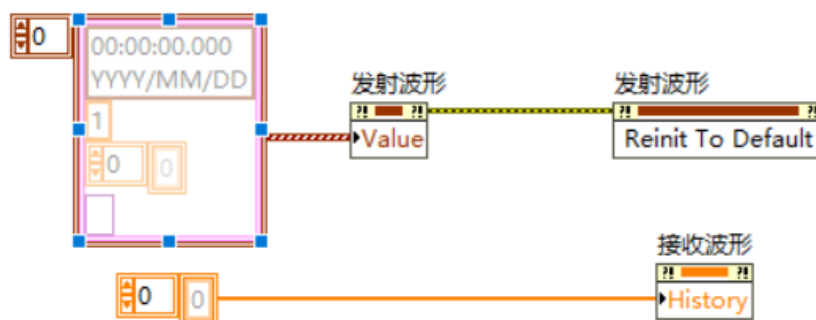


图 6.7 发射波形初始化和接受波形清零

因为在发射按钮按下时，我们的波形设置会发生改变，所以此时我们要将发射波形图中的图形进行新的初始化，使他显示为当前的波形，使用波形 Reinit To Default 可以重新显示当前的波形图。

而我们上次接收到的接收波形的数据，仍然显示在波形图中，所以我们要对接收波形的数据进行清零，使用属性节点 History 赋值为常量 0 即可实现清零。

### 6.3.4 互补波形产生

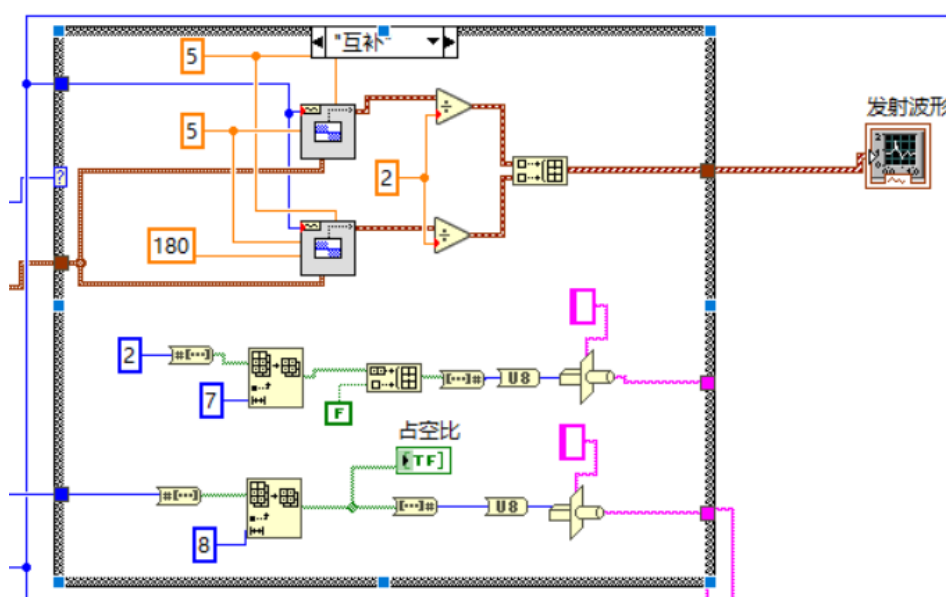


图 6.8 产生互补波形

根据方波波形的组件，我们可以看到我们产生了两组频率相同、相位相差 180 度的互补波形，并且我们可以看到，我们对于互补的波形，使用第一个字节的第一位和第二位去表示，若为 10 即表示此时为互补的波形。

### 6.3.5 非互补波形产生

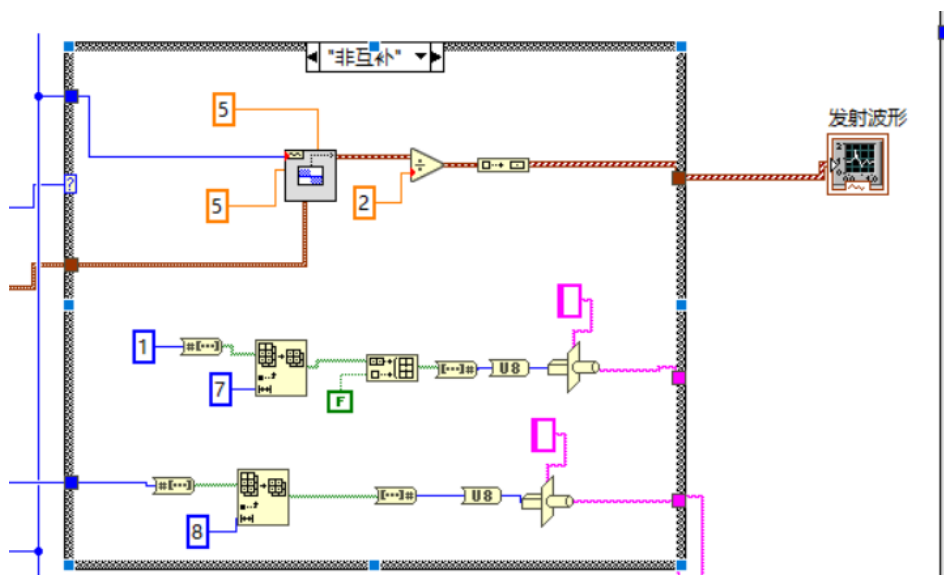


图 6.9 产生非互补波形

在此我们可以看出产生非互补波形时，我们只需要关闭一路方波信号即可，并且我们设定，发射信号第一个字节的第一位和第二位为 01 时，波形类型为非互补类型。

### 6.3.6 数据发送

我们通过 for 循环来对波形数据进行发送，共发送五次，共计 30 个字节。

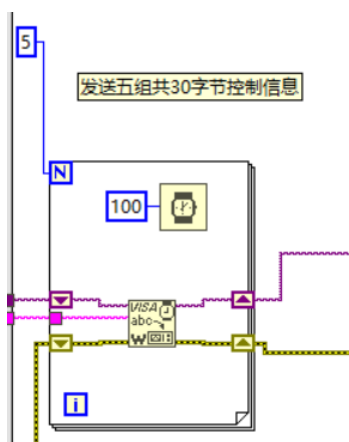


图 6.10 数据发送

### 6.3.7 对发射波形图表坐标轴调节

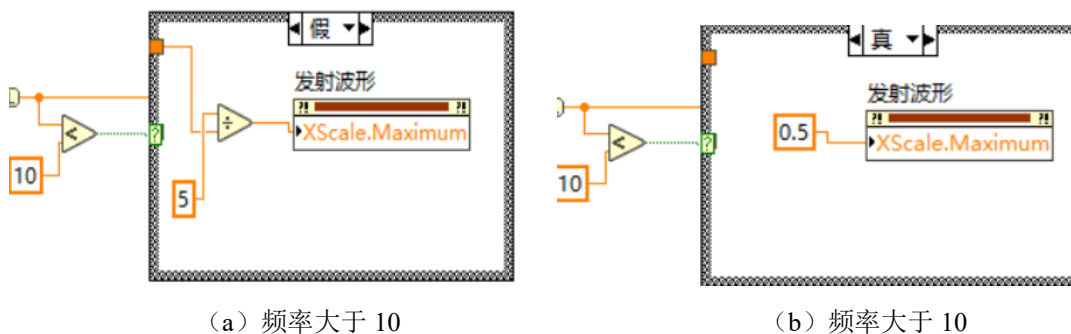


图 6.11 调节发射波形坐标轴

为了在高频与定频环境下都能在波形图显示处显示易于观察的实例波形，因此采用了在频率高于 10 时修改坐标轴大小的方式，以固定图像不变；而在频率小于 10 时固定坐标轴，以促使图像发生改变。其程序设计主要使用属性结点对波形图坐标轴大小进行控制。

### 6.3.8 通讯指示灯亮起



图 6.12 通讯指示灯亮

当我们在进行发送时，通讯指示灯会闪烁五次，表示我们的信号发射进行中。

## 6.4 接收部分 Labview 设计

数据信号接收部分的程序主要框架如图 6.14 所示。

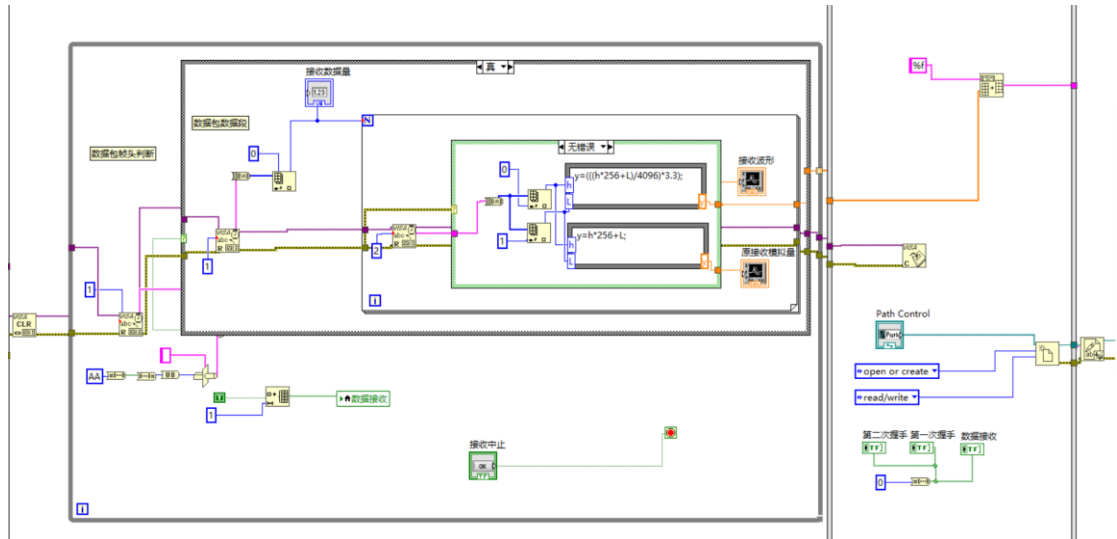


图 6.14 数据信号接收部分程序

整个程序由于需要执行 3 次挥手的通讯连接过程，因此采用了以平铺式顺序结构为基础的程序结构，依次进行了界面控制与通讯连接、第一次握手、第二次握手、第三次握手与数据读取和数据保存和终止控制信号传递这几个部分，其主要程序流程图如图 6.14 所示。

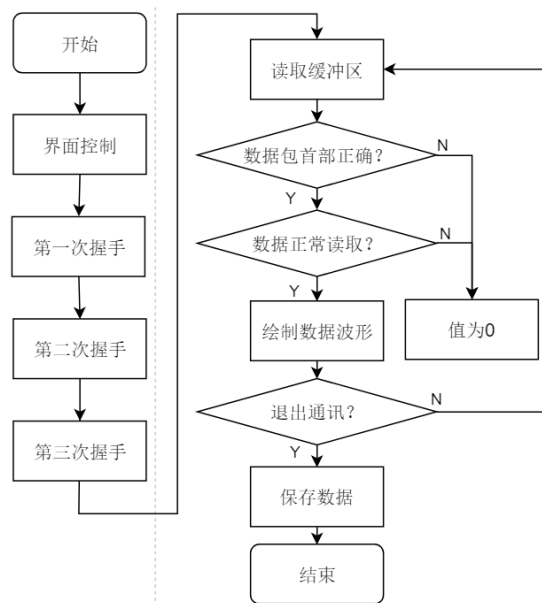


图 6.15 数据信号接收部分程序设计流程图



### 6.4.3 第三次握手以及数据接收

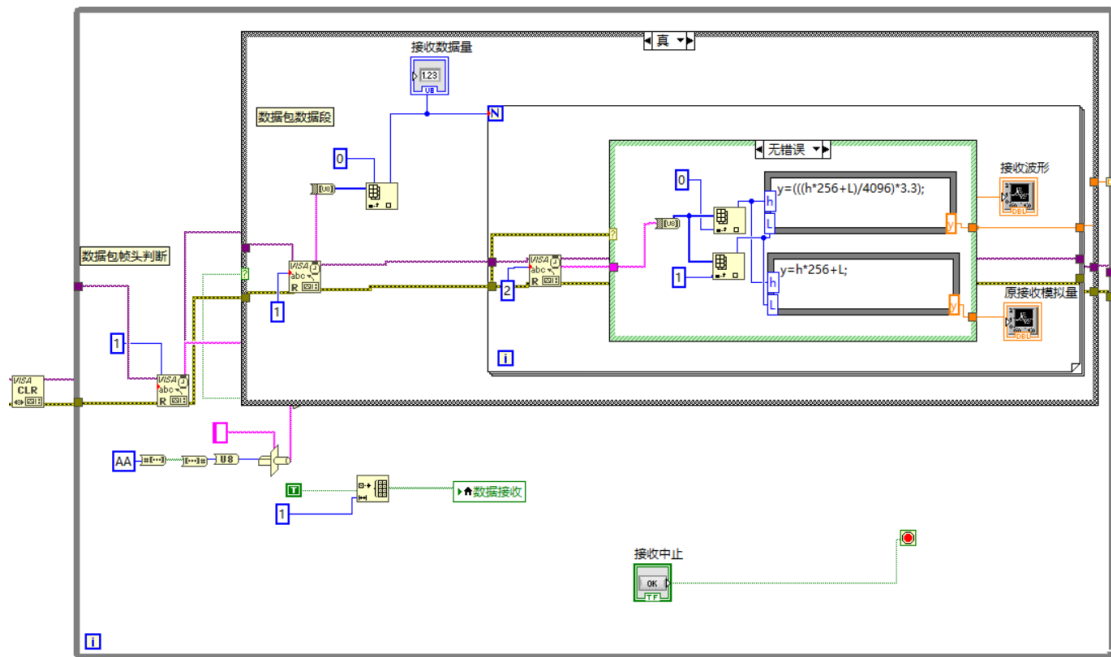


图 6.14 数据信号接收部分第三次握手及数据传送

经过前两次的握手，已经基本确定数据发送端和接收端都已经做好了通讯的准备，从第三次握手开始就会开始传递含有一字节首部信息以及两字节为一组数据的 50 组数据（共 100 个字节）信息的数据包。

我们将原数据和经过处理的后的数据显示在两张不同的表中。

### 6.4.4 数据保存

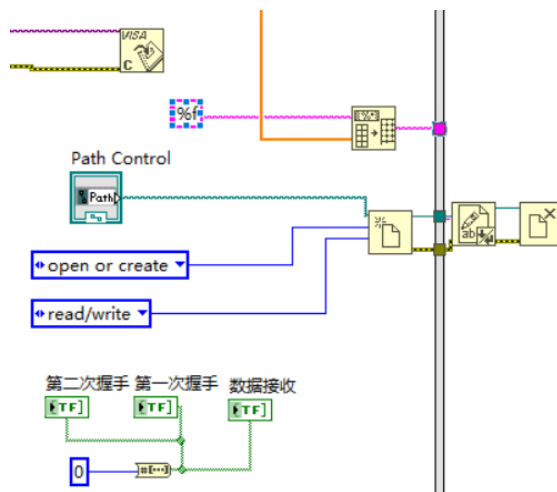


图 6.14 数据信号接收部分数据保存

当使用者按下“Pause”键后停止读取数据，就会进入到本阶段。

在数据保存方面，我们会将数据保存在文件中，方便我们后续的使用。

## 6.5 Labview 部分测试

我们对 Labview 部分进行设计完成之后，对它进行了测试。

### 6.5.1 发射测试

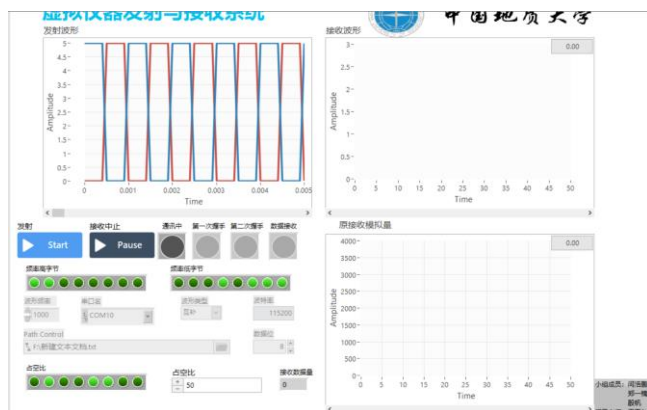


图 6.15 Labview 向串口发送信息

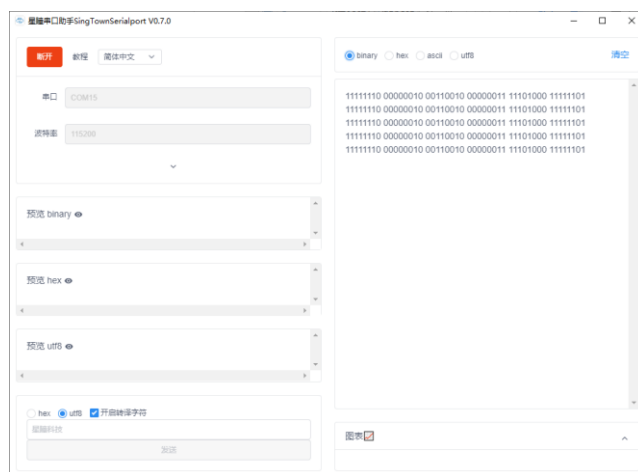


图 6.16 串口接收到 Labview 发送的波形信息

测试中我们发送了如图所示的波形信息，然后我们在串口助手中可以看到，串口成功地接收到了这部分信息，可以看到我们的发射部分并没有错误。

### 6.5.2 第一次握手及第二次握手

在下图中我们使用串口给上位机发送了 FF 的信息，这是单片机请求建立连接，上位机接收到之后第一次握手指示灯亮起，而后上位机自动发送 FF 的信息传回连接建立的信息，此时串口会收到 FF 的回复表明连接建立，接下来就会进入到数据接收阶段。



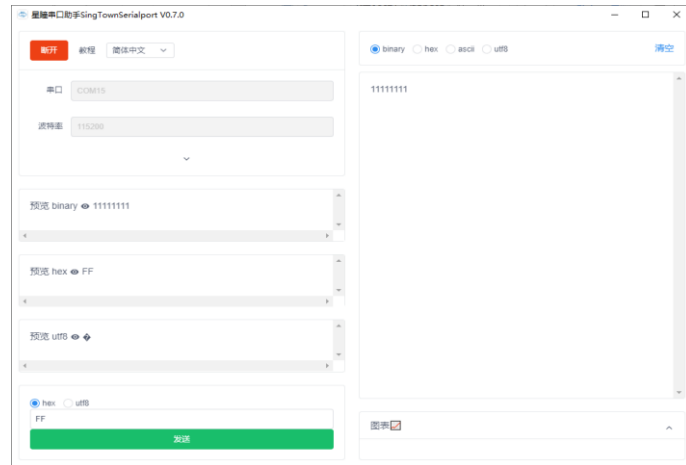


图 6.17 串口在第一次握手和第二次握手的接收与发送

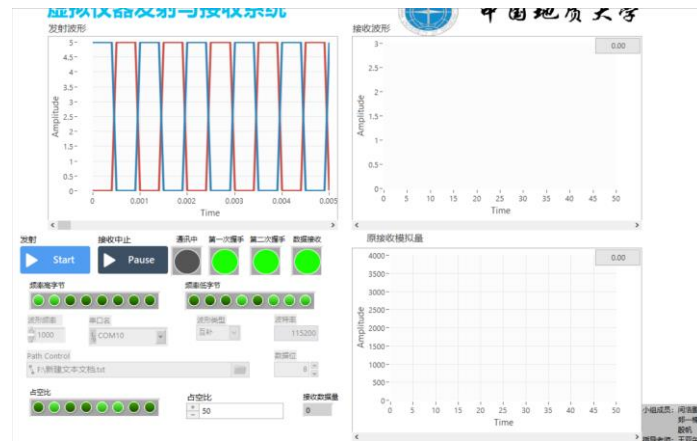


图 6.18 上位机在第一次握手和第二次握手的接收与发送

### 6.5.3 数据接收阶段

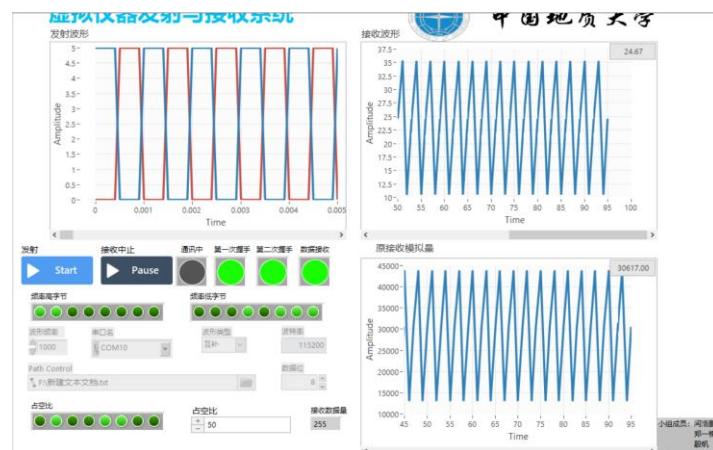


图 6.19 上位机第三次握手及数据接收波形显示

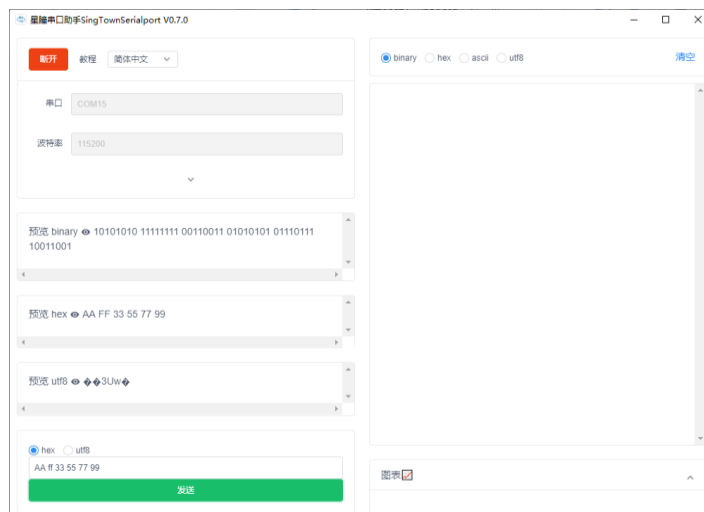


图 6.20 串口向上位机数据发送

在此时，我们使用串口助手发送帧头为 AA 的数据信息，于是我们发送信息之后在接受图表中显示，可以看到数据接收部分功能也完善，所有功能均正常运行。

## 七 整体调试

在仪器的调试中，我们使用左右金属探针发送 PWM 信号，使用中间探针接收采集信号，观测采集到的信号是否存在波动，根据波形的波动状态得出结论，观测是否可以探测到有金属的插入与拔出。

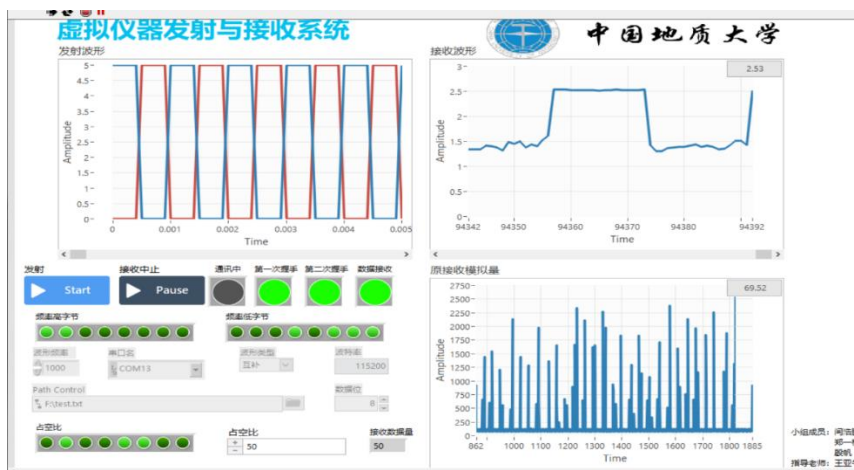


图 7.1 上位机 PWM 发射与接收显示

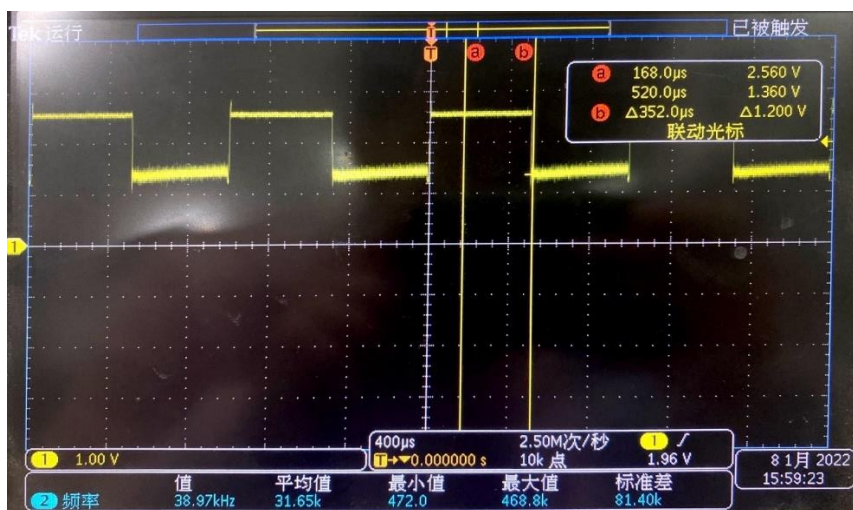
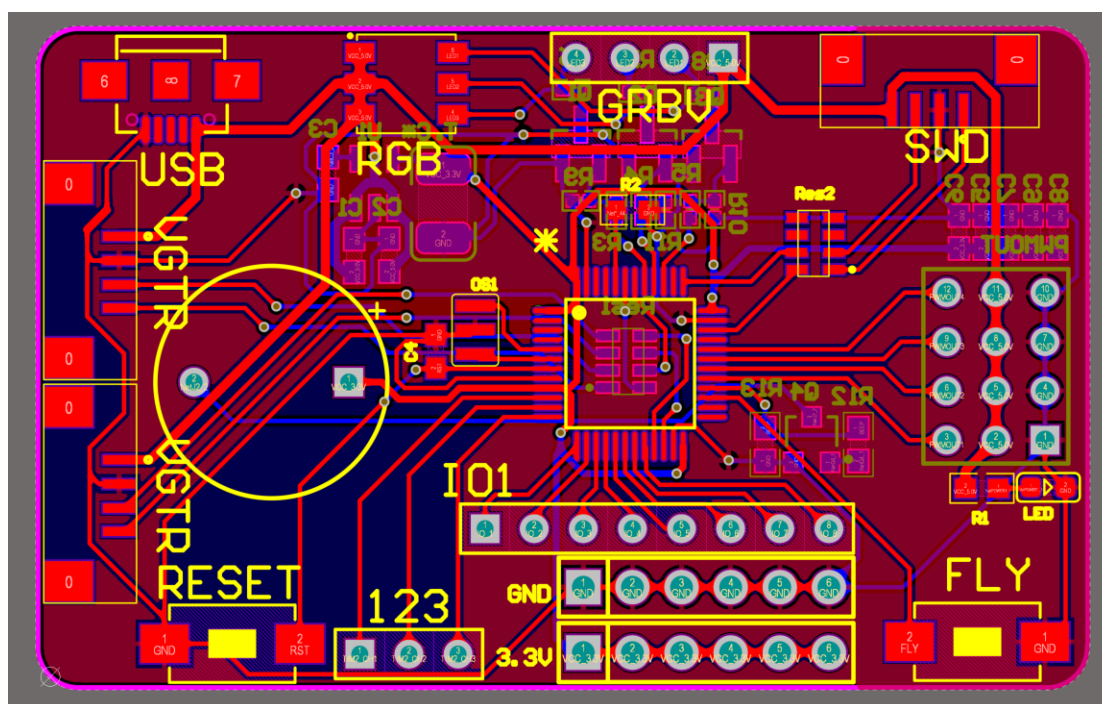
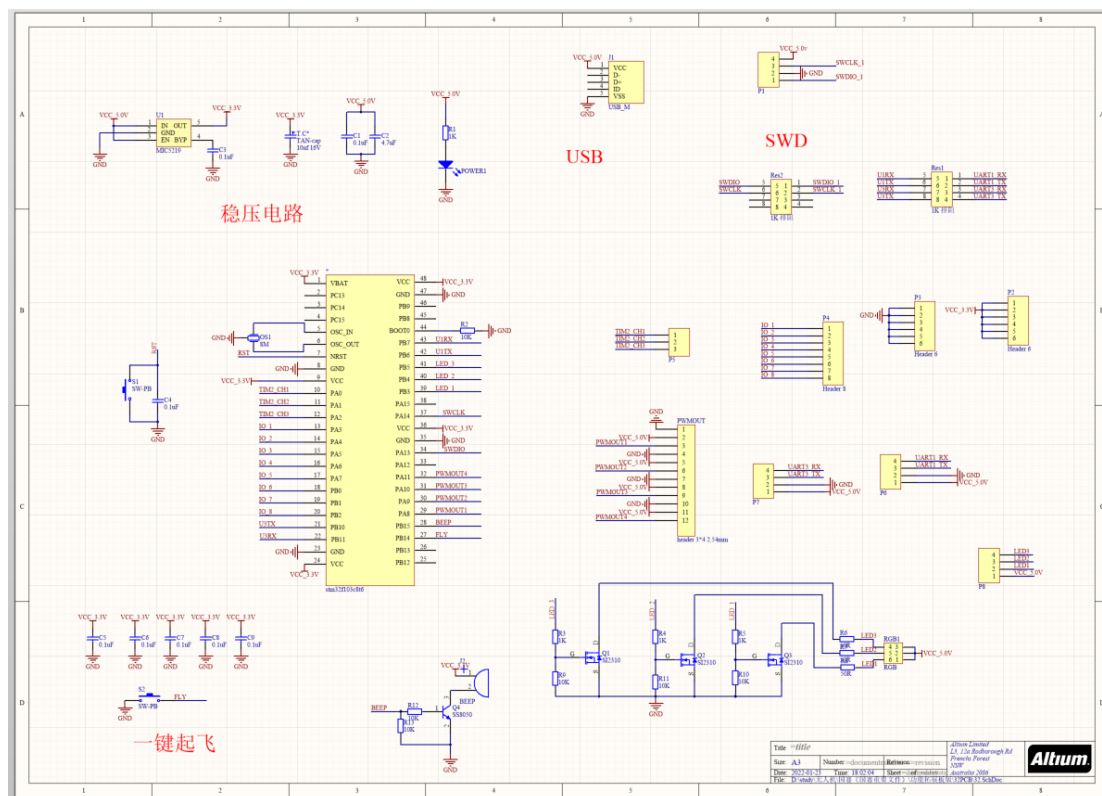


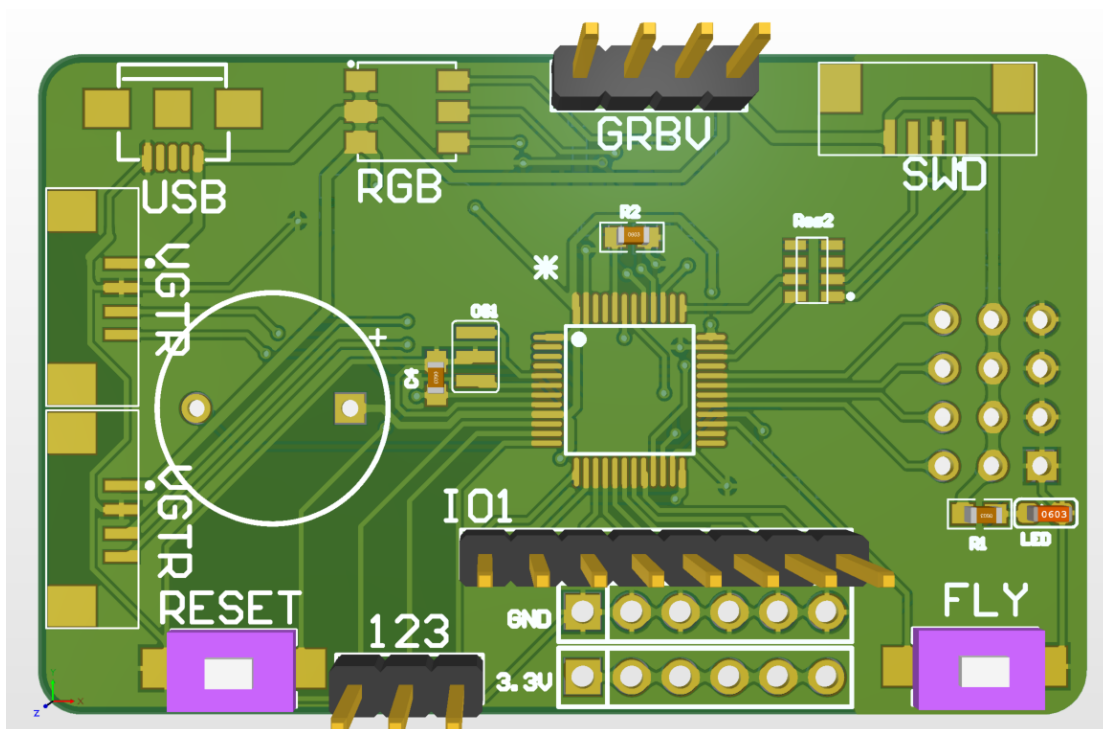
图 7.2 采集信号的示波器显示

我们在图中可以发现采集到的数据在金属插入时会出现波峰，因为我们是滤波之后的数据，所以我们在上位机进行观测时可以清晰地看到图像的波峰所在，通过上述，我们成功的完成了电法勘测的目的，获得了地下电阻率的信息。

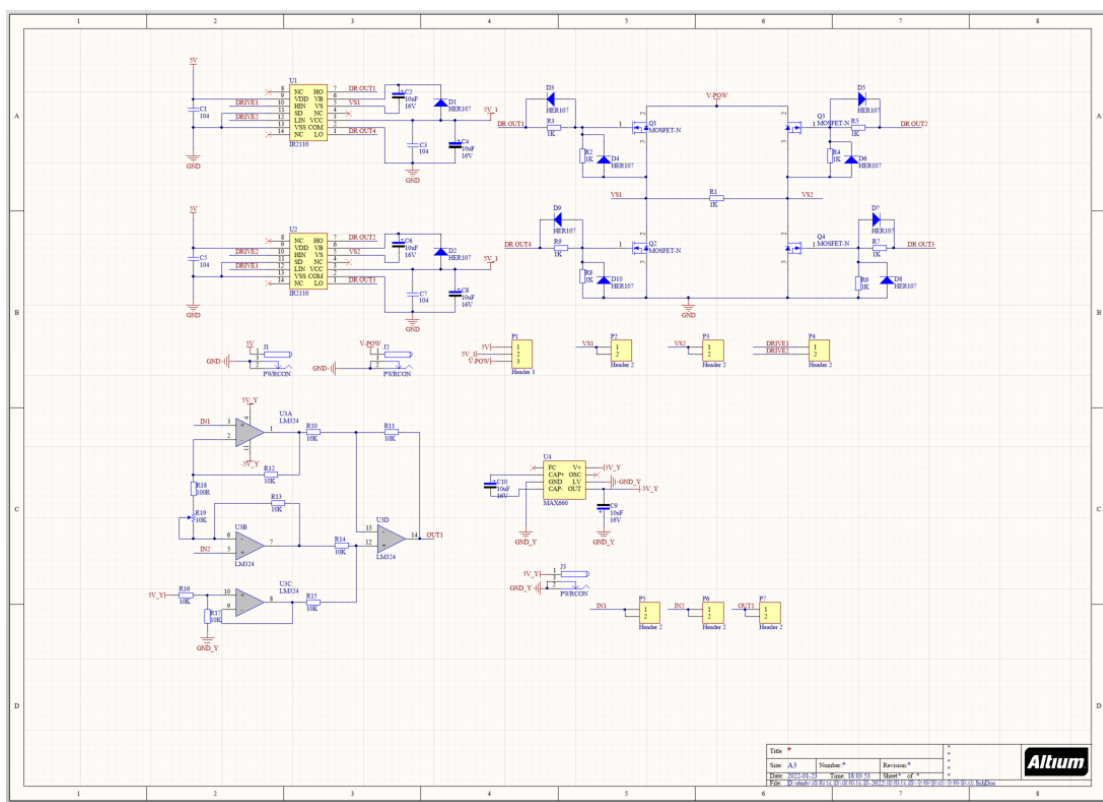
## 九 附录

### 9.1 单片机功能板硬件设计

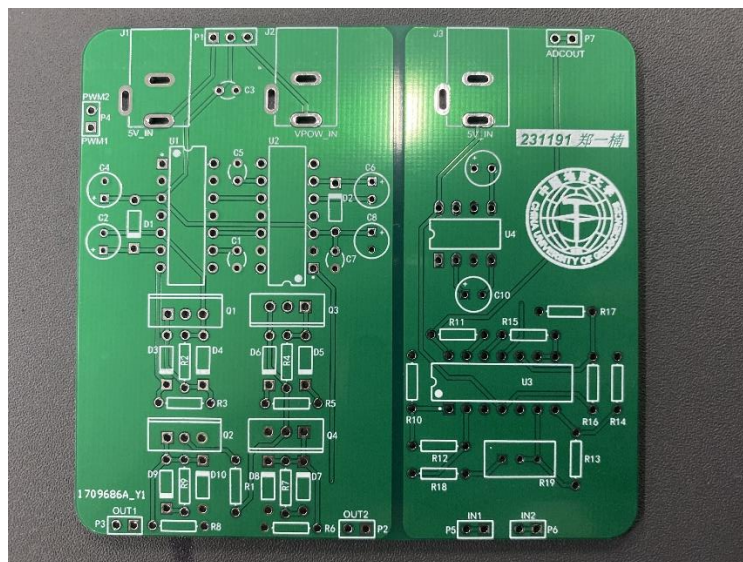
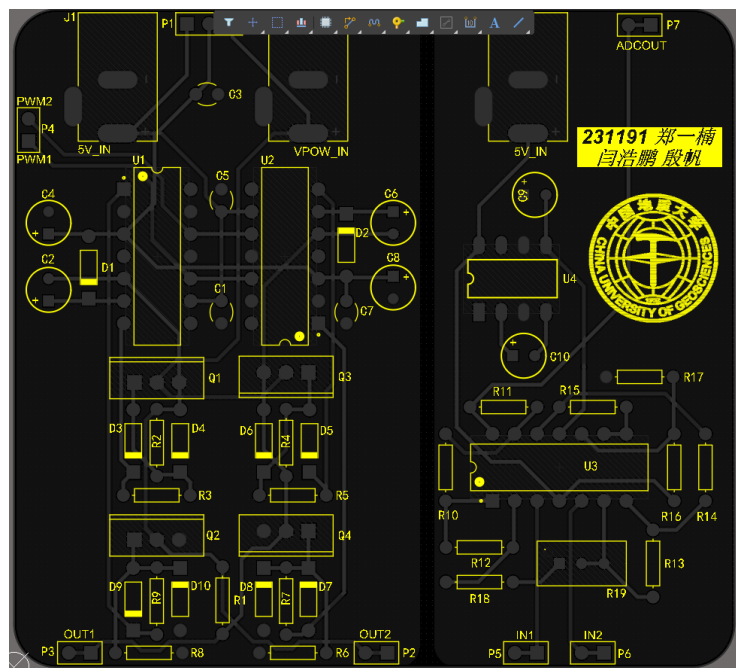




## 9.2 硬件电路板（发射与接收）







### 9.3 软件部分核心代码

```

void MX_TIM1_Init(void)
{

    /* USER CODE BEGIN TIM1_Init 0 */

    /* USER CODE END TIM1_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};
    TIM_OC_InitTypeDef sConfigOC = {0};
    TIM_BreakDeadTimeConfigTypeDef sBreakDeadTimeConfig = {0};

    /* USER CODE BEGIN TIM1_Init 1 */

    /* USER CODE END TIM1_Init 1 */
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 71;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 19999;
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
    htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim1) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim1, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_TIM_PWM_Init(&htim1) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sConfigOC.OCMode = TIM_OCMODE_PWM1;

```

```

sConfigOC.Pulse = 0;
sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
sConfigOC.OCNPolarity = TIM_OCNPOLARITY_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
sConfigOC.OCIdleState = TIM_OCIDLESTATE_RESET;
sConfigOC.OCNIdleState = TIM_OCNIDLESTATE_RESET;
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
{
    Error_Handler();
}
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_2) != HAL_OK)
{
    Error_Handler();
}
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_3) != HAL_OK)
{
    Error_Handler();
}
if (HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_4) != HAL_OK)
{
    Error_Handler();
}
sBreakDeadTimeConfig.OffStateRunMode = TIM_OSSR_DISABLE;
sBreakDeadTimeConfig.OffStateIDLEMode = TIM_OSSI_DISABLE;
sBreakDeadTimeConfig.LockLevel = TIM_LOCKLEVEL_OFF;
sBreakDeadTimeConfig.DeadTime = 0;
sBreakDeadTimeConfig.BreakState = TIM_BREAK_DISABLE;
sBreakDeadTimeConfig.BreakPolarity = TIM_BREAKPOLARITY_HIGH;
sBreakDeadTimeConfig.AutomaticOutput = TIM_AUTOMATICOUTPUT_DISABLE;
if (HAL_TIMEx_ConfigBreakDeadTime(&htim1, &sBreakDeadTimeConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM1_Init 2 */

/* USER CODE END TIM1_Init 2 */
HAL_TIM_MspPostInit(&htim1);

}

void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

```



```

/* GPIO Ports Clock Enable */
__HAL_RCC_GPIOD_CLK_ENABLE();
__HAL_RCC_GPIOA_CLK_ENABLE();
__HAL_RCC_GPIOB_CLK_ENABLE();

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(TRIG_GPIO_Port, TRIG_Pin, GPIO_PIN_RESET);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOB, BEEP_Pin|LED1_Pin|LED2_Pin|LED3_Pin,
GPIO_PIN_RESET);

/*Configure GPIO pin : PtPin */
GPIO_InitStruct.Pin = TRIG_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(TRIG_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pin : PtPin */
GPIO_InitStruct.Pin = FLY_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
GPIO_InitStruct.Pull = GPIO_PULLUP;
HAL_GPIO_Init(FLY_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pins : PBPIn PBPIn PBPIn PBPIn */
GPIO_InitStruct.Pin = BEEP_Pin|LED1_Pin|LED2_Pin|LED3_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/* EXTI interrupt init*/
HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);

}

void MX_USART1_UART_Init(void)
{

/* USER CODE BEGIN USART1_Init 0 */

```

```

/* USER CODE END USART1_Init 0 */

/* USER CODE BEGIN USART1_Init 1 */

/* USER CODE END USART1_Init 1 */
huart1.Instance = USART1;
huart1.Init.BaudRate = 115200;
huart1.Init.WordLength = UART_WORDLENGTH_8B;
huart1.Init.StopBits = UART_STOPBITS_1;
huart1.Init.Parity = UART_PARITY_NONE;
huart1.Init.Mode = UART_MODE_TX_RX;
huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart1.Init.OverSampling = UART_OVERSAMPLING_16;
if (HAL_UART_Init(&huart1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN USART1_Init 2 */

/* USER CODE END USART1_Init 2 */

}

```