

小质谱仪实验改进

张鹤天 06300720394

指导教师：俞熹

内容提要：近代物理实验室中现有的系统是通过机械控制的 XY 记录仪控制扫描电源的电压变化，并直接记录在 XY 记录纸上。在使用的时候具有扫描参数不可调节，所得结果不是电子化不利于后期数据处理等缺点，和现代物理实验的习惯相脱节。故笔者尝试使用 Arduino 单片机对本实验进行改装，使其可以全程通过电脑来进行参数控制并得到数字化的实验结果。

本改装系统主要分为 Arduino 与电脑之间的串口通信、及板载程序的编写和对实验仪器的控制、传感器信号放大四个部分。最终在 PC 端使用 Labview 编程制作总控程序，实现通过电脑来全程控制实验过程，并分析和储存实验数据。

关键词：小质谱仪 数字化改装 Arduino 微电压放大

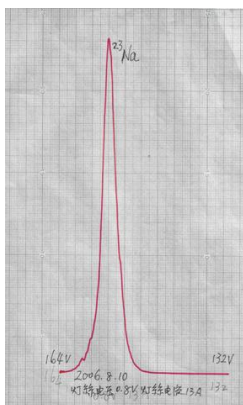
Abstract: The existing system in the Modern Physics Laboratory is controlled by mechanical XY recorder which controls voltage power supply and directly records data in the XY recording paper. The scan parameters cannot be adjusted and the results are not-electronic data that is inconvenient to process. So I try to use the Arduino microcontroller to modify the experiment so that it can be full parameter control via a computer and get digital results.

The modified system consists of Arduino and the serial communication between computers and Arduino, Onboard Programming to control laboratory instruments, Sensor signal amplification. The final program in the production of PC-using is programmed by Labview, to achieve easy control by computer and to analysis & store experimental data.

Keywords: Mass spectrometry Digital conversion Arduino Voltage amplification

一. 小质谱仪实验简介

质谱仪是一种使试样分子电离成离子，并通过磁场，利用带电粒子在电磁场中的运动来实现将粒子按照质荷比来进行分离，从而探测物质的组分与含量的实验仪器，在测定原子质量和寻找同位素，组分分析与组分分离的领域都有重要的应用。



近代物理实验室中现有的系统是通过机械控制的XY记录仪控制扫描电源的电压变化，并直接记录在XY记录纸上。得到的结果大抵如左图所示。这个系统简洁明了，但是在使用的时候具有扫描参数不可调节，所得结果不是电子化不利于后期数据处理等缺点，和现代物理实验的习惯相脱节。故笔者尝试使用Arduino单片机对本实验进行改装，使其可以全程通过电脑来进行参数控制并得到数字化的实验结果。

二. Arduino 单片机概况介绍

Arduino，是一个基于开放源代码的软硬件平台，构建于开放源代码 simple I/O 接口版，并且具有使用类似 Java，C 语言的 Processing/Wiring 开发环境。它可以通过串行借口与电脑相连接并取电，也可以使用外接的 5V 直流供电。

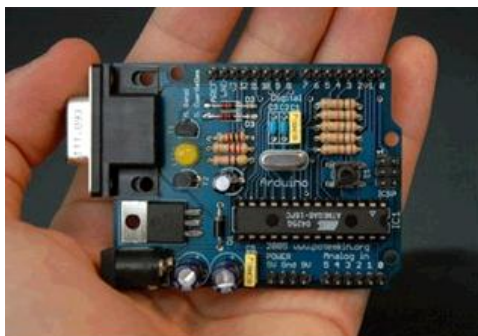


Photo by Nicholas Zambetti

一块 Arduino 板的大小如左图所示。现在我们采用 USB 转串口的连接线将之与个人电脑相连并安装驱动程序¹之后即可使用。Arduino 上专属的编程环境可以在 <http://arduino.cc/en/Guide/HomePage> 下载得到，在选择端口号和机器类型之后即可开始编程²。总体来说这个环境和 C 语言较为接近，如果熟悉 C 语言的话应该可以迅速上手。但是值得注意的是由于单片机计算环境的薄弱，其变量类型的长度会与 C 语言不同。例如其 int 型数据就只有八位，而不是 C 语言中的十六位³。如果不注意这种细节则会在具体的编程实践中遇到很大的困扰。

实验室的 Arduino-ATMAGE1280 卡拥有六个 PWM 输出端口，输出范围为 0-5V，在编程

¹ 最新版驱动程序可以在 <http://www.ftdichip.com/Drivers/VCP.htm> 得到

² 端口号请在设备管理器中查询，设备类型复旦物理系购买的型号为 ATMEGA1280

³ 参见 Arduino 官方网站中的技术文档：“Integers are your primary datatype for number storage, and store a 2 byte value. This yields a range of -32,768 to 32,767 (minimum value of -2^{15} and a maximum value of $(2^{15} - 1)$.” 原始地址：<http://arduino.cc/en/Reference/Int>

控制端则使用 0-255 的数字进行控制。即 0 对应 0V，255 对应 5V，线性变化。

三. 扫描电源改进方案

扫描电源的基本思路是通过电脑通过串口传递命令(数字信号)给 Arduino, 然后 Arduino 通过 PWM 的方式输出模拟信号(即直流电压信号)给程控电源。实验室中采用的电源为北京大华无线电公司生产的 DH1722A 型稳压程控电源⁴(如右图所示)。其后面板上有一个标准 RS232 接口来接受外控⁵。其输出电压 E_0 与外



控电压 E_1 ⁶ ($0 \leq E_1 \leq 5V$) 之间的关系表达式如下:

$$E_0 = \frac{1}{5} E_{\max} * E_1 * \alpha \quad (V)^7$$

其中 E_{\max} 是最大额定电压, α 是比例系数, 确定于电源前面板的电压调节电位器(图中左边一个旋钮, 右边的是电流调节电位器)。顺时针最大时为 1, 逆时针最大时为 0。

笔者最初的想法是在电脑端编程, 通过用户指定扫描电压的最小值、最大值与扫描步长、间隔时间, 将每一次扫描的具体命令都在电脑端计算好再通过串口传输给 Arduino。但经过尝试之后发现这么做是有问题的, 因为标准的 RS232 串口通信一般选择 9600 的波特率, 9600 比特每秒, 即 1200 字节每秒。考虑到串口传输的异步性, 扣除了起始位与终结位的 20% 损失之后, 就只有 960 字节每秒。这样的传输速度在快速的扫描过程中是不够的(因为每次命令的传输一般需要 8 或者 16 个字节, 这样的话在毫秒级别的变化过程中就会产生问题)。况且单片机同时还要向 PC 端传输扫描的到的数据, 这样的话对带宽的要求就又提高了一倍。这样算来的话电脑端编程所能适用的扫描参数范围远不够广。处于这种考虑我决定将运算端放在 Arduino 中, 虽然这样会增加一定的编程难度(Arduino 语言的灵活性比 C 语言稍弱), 但可以将 PC 与 Arduino 之间的数据传输量大大减少——每次扫描只需要传输一个最高电压, 一个最低电压, 与扫描步长延时。

在具体编写串口通信模块的时候还遇到了几个问题, 首先是 PC 端传给 Arduino 的数据不正常, 全是一些毫无规律的数字。经过漫长的调试, 发现串口传输数据是按位来传输的。即 PC 端如果送出数字“123”, 则实际上通过串口输送的数据为“49、50、51”, 即“1、2、3”的 ASCII 码, 即串口通信的过程中只能传送 ASCII 码中存在的数字。如果需要传送长数据

⁴ 额定电压 0-300V 输出电流 0-1A

⁵ 虽然只有其中两个针脚是有用的, 用电脑上常见的跳线来进行连接较为方便。

⁶ 这个电压和 Arduino 中 PWM 输出范围正好相同。

⁷ 参见 DH1722A 型电源的产品说明书

的话，就需要在数据的首尾添加识别符。其次，Arduino 语言中的串口读取函数 `(int)Serial.read()`⁸的作用机理类似于 C 语言中的 `getchar()`而不是 `scanf()`，即它是一位一位分别读取并储存的。即如果 `Serial.available()`函数判断出串口有数据送来，则可调用 `Serial.read()`函数去按位读取，且每读到一个 ASCII 码数据之后，返回读取到的这个数据⁹，并在串口缓存区擦去刚读到这个数据并继续判断是否有进一步的数据传来。由此则需要创建一个数组来存储读到的这些数据，并把这些 ASCII 码数据换算成正常的字符，并识别出其中的首尾标识符，将字符再换算成计算机端想要传输的本地数据。最后，`Serial.read()`函数读取数据并清空串口缓存区的速度是有限的，如果使用一个 `for` 循环来进行数据读取的话，则很可能两次读到同一个数值——即 100 很可能就被读取成 110000 这样。所以在读取数据的循环中也需要加入一定的延时。考虑到这里传送的数据量并不大，所以延时就设定成了非常稳妥的 75 毫秒。具体这个模块的编程如下：

```
int readnumber()
{
    int i=0;

    int data[10];

    int final=0;    //初始化变量
    for(i=0;i<10;i++) //当读取的数字为 0 时认为读取结束，注意这里的 0 是 ASCII 码中
        data[i]=0;    //的 0，而不是数字 0。数字 0 对应的 ASCII 码是 48.
    for(i=0;Serial.available();i++) //Serial.available()函数是 Arduino 语言中的库函数，
    {
        //若返回非 0 则代表串口缓存区还有数据
        data[i]=Serial.read();
        final=(data[i]-48)+final*10;//将 ASCII 码换算成实际数字，并将其组合成原始数据
        delay(75);    //设定延时
    }
    Serial.flush();    //清空串口缓存区
    return final;
}
```

四.AD624 放大电路与计算机之间的通信

⁸ Arduino 语言区分大小写，`Serial.read()`与 `serial.read()`并不是同一个函数。

⁹ 所以这个函数只会返回 `int` 型变量，因为其实读到的都是 ASCII 码

模块的另一个主要问题是由离子分析室出来的微电流信号 ($10^{-11} \sim 10^{-9}$ A) 经过微电流计转换成直流电压信号之后依然只有毫伏数量级, 远远不能被 Arduino 的模拟输入针脚所识别 (其识别范围和输出范围一样, 都是 0-5V)。最初考虑通过一个变压器来实现电压的放大, 但经过略微的查询后即发现变压器对于毫伏级电压放大的损耗率太高¹⁰, 不适合采样信号的传输。后尝试在模拟电子学教程中寻找类似于范例模板的放大电路, 却发现模拟电子学中所处理的电压信号都是交流的, 而直流电平信号只是提供一种基础偏置, 在通常的模电问题中是没有意义, 需要被清除的部分。最后经过查询, ADI 公司¹¹生产的运算测量放大芯片¹²可以比较始于处理这样的情况:

FEATURES

Low Noise: 0.2 μ V p-p 0.1 Hz to 10 Hz

Low Gain TC: 5 ppm max (G = 1)

Low Nonlinearity: 0.001% max (G = 1 to 200)

High CMRR: 130 dB min (G = 500 to 1000)

Low Input Offset Voltage: 25 μ V, max

Low Input Offset Voltage Drift: 0.25 μ V/ $^{\circ}$ C max

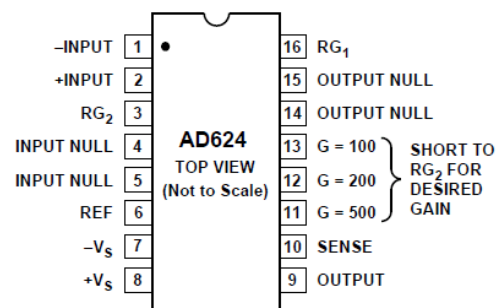
Gain Bandwidth Product: 25 MHz

Pin Programmable Gains of 1, 100, 200, 500, 1000

No External Components Required

Internally Compensated

CONNECTION DIAGRAM



FOR GAINS OF 1000 SHORT RG_1 TO PIN 12 AND PINS 11 AND 13 TO RG_2

实验中最后选用的 AD624 芯片是第二代集成 IA 的典型芯片, 内部采用激光修正技术, 性能较第一代有所提高。且内置了调节电阻, 只需要连接针脚的跳线即可调节其放大增益的倍数及进行偏置增益调零, 对于本实验非常的方便。AD624 的技术参数和封装形式大抵如上图所示¹³, AD624 的失调电压和温度漂移都是在 μ V 数量级, 对于处理 mV 数量级电压的本实验来说足堪使用。其中 V_s 端接 18V 的直流外接电源。Pin6 的 REF 端接地, pin10 的 Sense 端一般是为了消除输出回路阻抗在 AD624 内部两级放大器之间的影响, 将之与 pin9 的输出端相连就好。而放大倍数的调节则使用针脚的连接进行调节。

¹⁰ 参见 <http://zhidao.baidu.com/question/71008725.html>

¹¹ 全称 Analog Devices, Inc.

¹² Instrumentation Amplifiers, 简称 IA

¹³ 引自 AD624 的产品说明书。 <http://wenku.baidu.com/view/c966889951e79b8968022615.html>

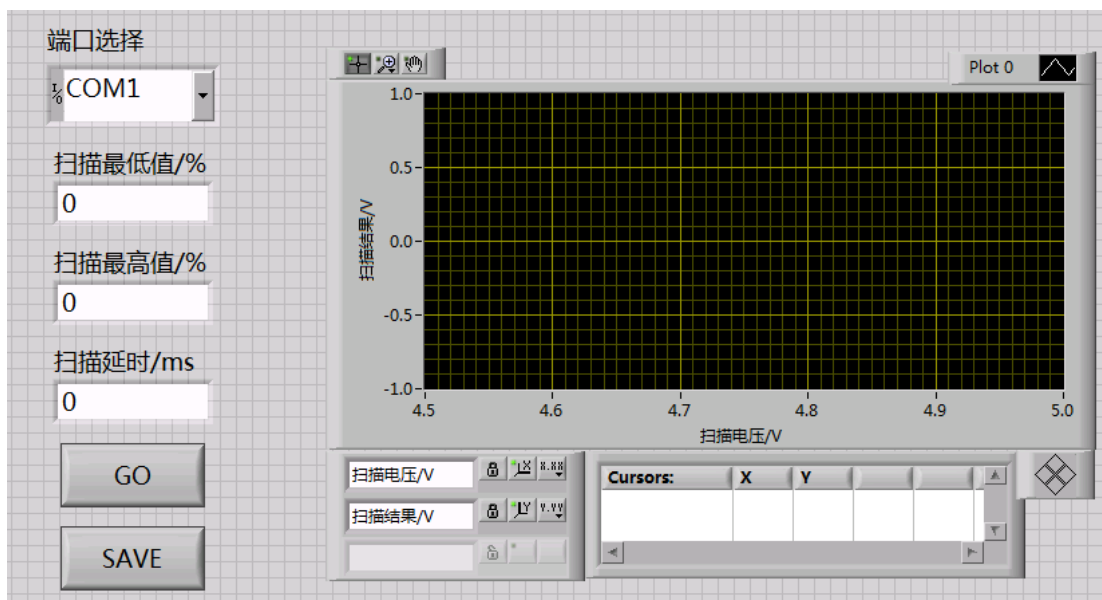
在具体的实验中，我采用实验室中的信号发生器来进行毫伏级电压的输出¹⁴，将 40db 和 20db 的两个减益按钮都按下，并将频率调整到最低（2Hz）之后，用万用表测得其输出电压在 1-4mV 左右震荡¹⁵。将其引入 AD624 放大电路之后（放大增益设定为 831 倍），得到的电压为 2-5V。并可以成功地被 Arduino 读取后通过串口传入 PC 端。采用的测试程序很简单，如下所示：

```
void loop()                                     //setup 模块仅有一个设定波特率的部分，略去
{
    int sensorValue = 0;
    sensorValue =analogRead(5);                //读取引脚 5
    Serial.println(sensorValue, DEC);          //值得说明的是 analogRead()函数返回的是从
    delay(200);                                //0-1023 之间的一个值，线性对应于 0-5V。
}                                               //溢出部分则只显示到边界值。
```

并发现信号发生器发出的方波的最低电压并不是 0，而存在一个 1-2mV 的基底，这个值在直接观察时几乎观察不到，但经过放大电路之后便显得相当明显。当然这可能也与信号发生器频率变化的弛豫时间有关，但其工作在最低频率时，理应不存在这样的问题。

四. 计算机端程序的编程

电脑端编程内容比较简单，主要是串口通信问题和最后的数据读取与保存。为了方便制



¹⁴ 因为小质谱仪实验抽真空的过程旷日持久，为了调试仪器而专门去做小质谱仪实验在时间上确实是不划算，所以就用信号发生器来模拟毫伏级的信号。

¹⁵ 因为信号发生器发出的信号始终都是方波信号，虽然频率已经调到最低，但始终还是在一个电压范围内震荡。且实验室中所使用的万用表并不是数字的，且直流电压的最小量程为 0.1V，故此读数并不是非常的精确。

作窗口化环境和数据的作图，最后选择了 Labview 进行编程。最后得到的界面如图所示：

其功能主要有以下几点：

1. 串口通信的基本参数选择

2. 对扫描参数的发送，并自动添加首控制字符进行识别。并对串口接收到的数据进行识别和处理

3. 对数据格式的转换，将扫描参数换算到 0-500，将接收到的参数从 0-1023 换算到 0-5V

4. 将最后得到的数据进行绘图，并将其自动保存到文本文件中。于程序过长，我把它截成了两张图。同时为了能压缩在两张图里面，对框体位置略有修改，可能造成了程序可读性的下降。原始的框图请见附录。

整个程序实现的都很简单。这里也不需要过多的说明。

五. 整体解决方案及感想

将以上各个部分拼合在一起之后，即可通过 PC 端控制软件控制实验的整个过程。初步实现了本改进方案的初衷。经过进一步的封装和调试之后，应该可以作为近代物理实验室小质谱仪实验的控制环境。作为更进一步的设想，还可以将实验中抽真空部分的控制与检测集成到本软件之中，不过那需要对仪器更进一步的改装与集成了。

本次实验作为笔者第一次，或许也是最后一次在实验室独立完成探索研究性实验，在这个过程中遭遇了许多一名新手所应遭遇的种种困难。从理论设计到实际操作，这其中总有种种超乎人意的麻烦，无论是程控电源点不亮还是串口通信失真，还是说数字放大模块的选择与调试，无一不涉及到自己完全陌生的领域。在海量的搜索结果中迅速寻找出适用于自己问题的合理信息，这种能力大概是物理系四年的教育所留给我的最宝贵财富了吧。当然还要感谢我的导师俞熹老师，如果没有他的帮助与建议，我这个从没有自己独自做过课题的新手恐怕无法完成这个项目。

六. 参考网站与文献

Arduino 官方网站

<http://arduino.cc>

Arduino 中文开发文档

<http://chaishushan.googlecode.com/svn/doc/arduino/r106/index.html>

AD624 使用说明书

<http://wenku.baidu.com/view/c966889951e79b8968022615.html>

复旦大学物理教学实验中心

<http://phylab.fudan.edu.cn/doku.php>

Arduino 北京俱乐部

<http://www.douban.com/group/308665/>

中国电子开发网

<http://www.ourdev.cn/bbs/index.jsp>

DH1227A 型程控电源使用说明书

随电源购买附赠

中国科技大学电子工程系关于 IA 器件的讲义

<http://home.ustc.edu.cn/~wdy/course/chap6.pdf>

七. 致谢

首先感谢我的导师俞熹老师，在我整个论文进行过程中给了我无私的建议与指导。让我一名实验新手渐渐找到了在实验室工作的感觉。

感谢魏心源老师，在 Arduino 编程过程中解答了我许多技术上的疑问。

感谢在网上热情分享自己经验，回答我的问题的各位先进，让我在从未涉足过的领域能够快速入手。

感谢我的父母，支持我完成我大学的学业，时时刻刻为我提供关怀与指导。

附录：Arduino 程序源代码

```
int apin = 9;           // 程控电压的 PWM 控制端相连于 pin 9
8
```



```
int readValue=10;

long volt=0;

int con=1;

int sensorValue = 0;

int value=0;

int maxv=0;

int minv=0;

int pinlv=0;          //初始化变量

void setup()          //Arduino 语言中的程序初始化模块
{
    Serial.begin(9600);          //串口通信设置波特率

    pinMode(apin, OUTPUT);      // 设定引脚的输出模式，只有 PWM 输出需要设置。输入或者
}                                //数字输出不需要

void loop()            //Arduino 语言中的主要部分，即循环部分
{
    while (Serial.available()) //如果有数据传来，则 Serial.available()不为 0
    {
        readcontrol(&con,&value); //readcontrol 是一个类似于上文段落中的函数，添加了首控制
        delay(75);                //符的识别。由于 Arduino 语言中函数返回两个值，于是采用
        Serial.flush();           //指针传递的方法来读取控制符与数值
        delay(1000);
        Serial.print(con);        //串口输出语句
        switch(con)               //通过 switch 语句来实现对从支付的识别
        {
            case 1:                //1 就是识别符
            {
                maxv=huansuan(value); //对最大电压的处理，及返回 PC 端的反馈信息
                Serial.print("maxvolt typed is ");
                Serial.print(value/5); //从 PC 端从来的数据是 0-500 的，对应扫描范围是
                Serial.println("%");   //0-5V,方便处理。这一步换算在电脑端程序中完成
            }
        }
    }
}
```

```
        break;
    }
case 2:                //最小电压
    {
        minv=huansuan(value);
        Serial.print("minvolt typed is ");
        Serial.print(value/5);
        Serial.println("%");
        break;
    }
case 3:                //扫描间隔时间
    {
        pinlv=value;
        Serial.print("mode typed is ");
        Serial.print(value);
        break;
    }
default:
    break;
}
}

if(maxv*minv*pinlv)    //当三者都读到有效数值之后，开始扫描
{
    saomiao(maxv,minv,pinlv);
    maxv=minv=pinlv=0; //扫描完成之后，将所有数据清零，防止 loop 函数再次调用扫
    Serial.println("saomiaowancheng");//描模块
}

}

void readcontrol(int* a,int* b) //对前文提到的 readnumber 函数的改写，传入的参数为方便
{
    //操作的指针，以克服函数返回值只能有一个的问题。
}
```

```
int i=0;

int data[10];

int final=0;

for(i=0;i<10;i++)

    data[i]=0;

for(i=0;Serial.available();i++)

{

    data[i]=Serial.read();

    if(i>0)          //对首控制符的判断

        final=(data[i]-48)+final*10;

    delay(75);

}

*a=data[0]-48;      //将函数中读取到的数值返回到全局变量中

*b=final;

Serial.flush();    //清空串口缓存区

}

int huansuan(int v) //换算函数，Arduino 的 PWM 输出的值为 0-255。写成函数方便

{

    //调用

    long ans=0;

    ans=(long)v*255/500; //在这里 int 型会溢出，所以一定要做一个 long 型的转换

    return (int)ans;

}

void saomiao(int maxv,int minv,int sudu) //扫描模块

{

    int sensorValue=0;

    for(;minv<maxv;minv++)          //这里的参数都是换算好的 0-255 格式的数据

    {

        analogWrite(apin,minv);    //PWM 输出函数

        delay(pinlv);

        sensorValue =analogRead(5); //模拟端口读入函数，定义为 pin5 的输入。Arduino
```

```
    //板上的不同部分的输入输出引脚编号都是独立的。但他们共享同一个接地端口
    Serial.println(sensorValue, DEC); //将读入的数据通过串口传回电脑，DEC 是格式控
  }                                     //制符，代表十进制
}
```