

基于ESP32与phyphox的DIS实验制作（3）—ESP32与phyphox的电压传感器实验

原创

MakeMaker 于 2021-12-25 09:18:00 发布 2468 收藏 3

分类专栏: [基于ESP32与phyphox的DIS实验制作](#) 文章标签: [单片机](#) [物联网](#) [计算机视觉](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qg_33378028/article/details/122139239

版权



[基于ESP32与phyphox的DIS实验制作](#) 专栏收录该内容

6 篇文章 1 订阅

订阅专栏

在第二篇里介绍了如何使用phyphox来进行物理实验, 我也用它做了一个简单的小实验, 测量电梯下降时加速度的变化, 不知道大家有没有算出我家住在几楼呢? 前两篇内容其实应该早有人写过类似的, 但是从这一节开始, 每一节都是满满的干货哦!

用phyphox的确可以做许多有趣的小实验, 但是毕竟手机上的传感器有限, 能进行的实验也很有限了。再加上物理中最常见的一些传感器例如电压电流等电学传感器、测量位移类的传感器、力学传感器等等的缺失, 导致phyphox仅仅依靠手机的传感器根本无法完成中学物理中常见的那些实验。

但是phyphox可以使用蓝牙连接ESP32 (或者Arduino), 这样phyphox就可以借助连接到ESP32上的传感器, 完成几乎所有的传感器实验了。我在这大胆的预言, 在未来的几年里, 使用ESP32和phyphox开发自制DIS实验将会席卷物理自制教具圈! 而我的这些教程, 希望能加快这个进程, 让更多的老师更快的加入到这个行列中来! 在这一篇, 我们就来完成一个最基础的电压传感器的实验, 大家请先看两段视频:

(观看视频请关注微信公众号: 宁中物理创新实验室)

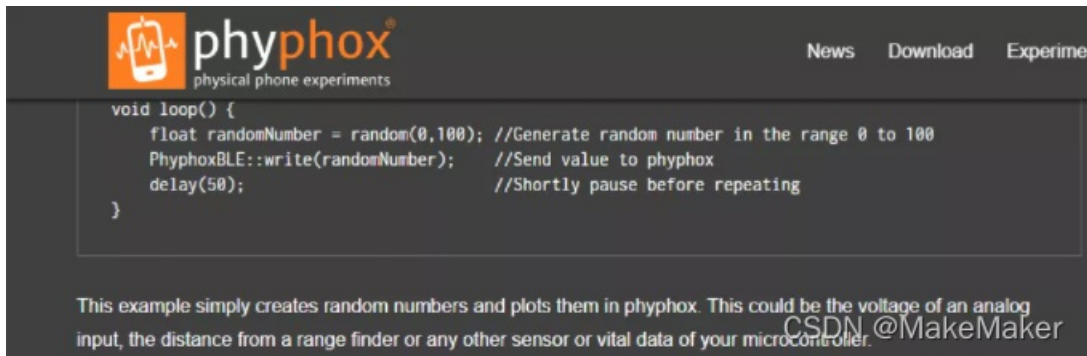
如何制作视频中的电压传感器呢, 跟着下面的教程, 一起来制作吧!

1、从www.phyphox.org网站中下载所需要的库文件

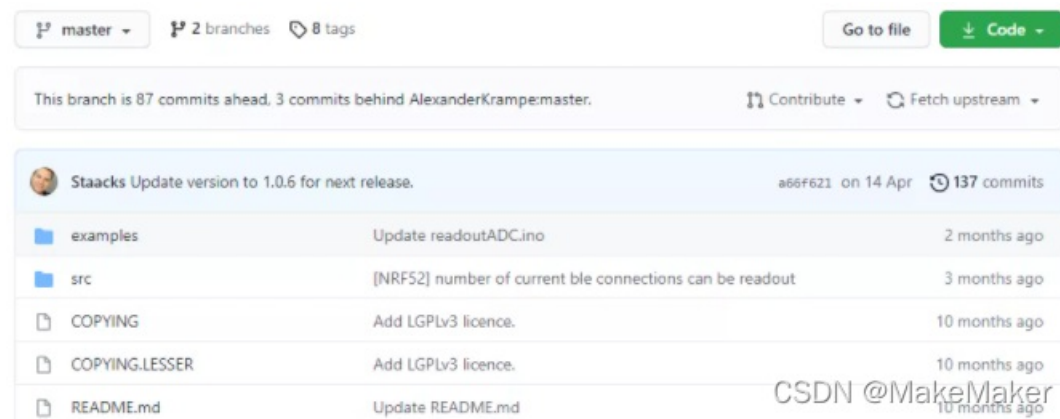
在More下的Material下的Arduino library内:

```
![在这里插入图片描述](https://img-blog.csdnimg.cn/a5e429b20c674f7a97b357f009bbac00.png?x-oss-process=image/watermark,type_d3F5LXplbmh1aQ,shadow_50,text_Q1NETiBATWFrZU1ha2Vy,size_13,color_FFFFFFFF,t_70,g_se,x_16)
```

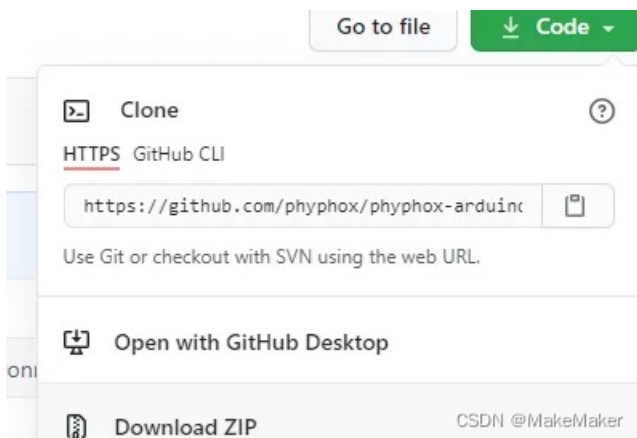
点击后进入下面界面，然后再找到Learn more on github.其中github是个超级链接，点击后就进入到了github里的文件页面。



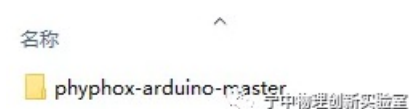
当然你也可以直接在浏览器里输入<https://github.com/phyphox/phyphox-arduino>直接进入：



点击Code,选择Download ZIP



下载完后将文件解压：



然后将文件改名成phyphox

名称	修改日期	类型	大小
phyphox	2021/4/14 22:25	文件夹	

注意此文件夹打开后就是下列文件

名称	修改日期	类型	大小
examples	2021/4/14 22:25	文件夹	
src	2021/4/14 22:25	文件夹	
COPYING	2021/4/14 22:25	文件	35 KB
COPYING.LESSER	2021/4/14 22:25	LESSER 文件	8 KB
library.properties	2021/4/14 22:25	PROPERTIES 文件	1 KB
RFADMF.md	2021/4/14 22:25	MD 文件	6 KB

然后再回到上一层目录，将phyphox整个文件夹复制，并将其粘贴到第一篇时讲过的Arduino目录内的libraries文件夹内：

名称	修改日期	类型	大小
drivers	2021/5/14 16:24	文件夹	
examples	2021/5/14 16:25	文件夹	
hardware	2021/5/14 16:25	文件夹	
java	2021/5/14 16:25	文件夹	
lib	2021/5/14 16:25	文件夹	
libraries	2021/5/14 16:25	文件夹	
reference	2021/5/14 16:25	文件夹	
tools	2021/5/14 16:25	文件夹	
tools-builder	2021/5/14 16:25	文件夹	
arduino	2021/5/14 16:25	应用程序	72 KB
arduino.l4j	2021/5/14 16:25	配置设置	1 KB
arduino_debug	2021/5/14 16:25	应用程序	69 KB
arduino_debug.l4j	2021/5/14 16:25	配置设置	1 KB
arduino-builder	2021/5/14 16:25	应用程序	23,156 KB
libusb0.dll	2021/5/14 16:24	应用程序扩展	43 KB
msvcp100.dll	2021/5/14 16:25	应用程序扩展	412 KB
msvcr100.dll	2021/5/14 16:25	应用程序扩展	159 KB

粘贴完后如下：

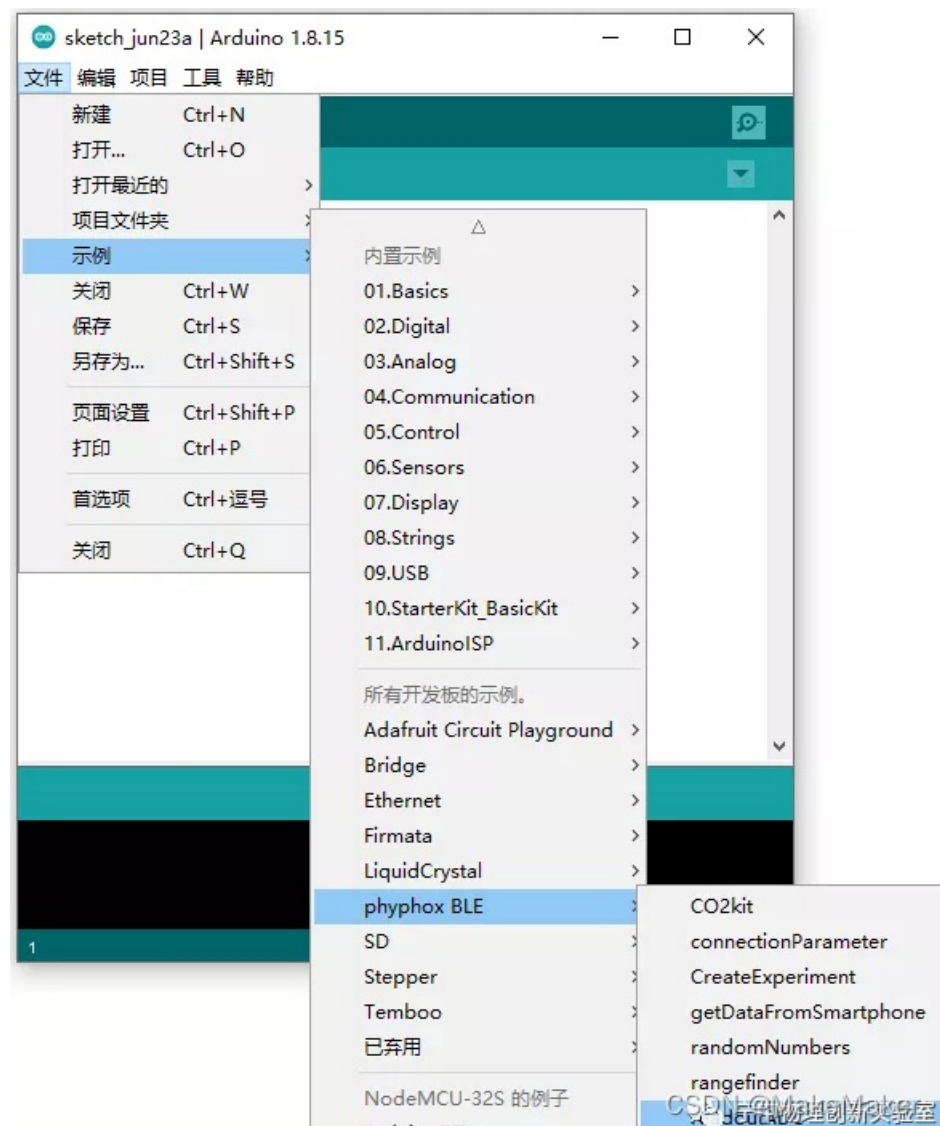
名称	修改日期	类型	大小
Adafruit_Circuit_Playground	2021/5/14 16:25	文件夹	
Bridge	2021/5/14 16:25	文件夹	
Esplora	2021/5/14 16:25	文件夹	
Ethernet	2021/5/14 16:25	文件夹	
Firmata	2021/5/14 16:25	文件夹	
GSM	2021/5/14 16:25	文件夹	
Keyboard	2021/5/14 16:25	文件夹	
LiquidCrystal	2021/5/14 16:25	文件夹	
Mouse	2021/5/14 16:25	文件夹	
phyphox	2021/6/23 19:49	文件夹	
Robot_Control	2021/5/14 16:25	文件夹	
Robot_Motor	2021/5/14 16:25	文件夹	
RobotIRremote	2021/5/14 16:25	文件夹	
SD	2021/5/14 16:25	文件夹	
Servo	2021/5/14 16:25	文件夹	
SpacebrewYun	2021/5/14 16:25	文件夹	
Stepper	2021/5/14 16:25	文件夹	
Temboo	2021/5/14 16:25	文件夹	

2、开始第一个ESP32与phyphox的实验程序

然后回到Arduino的根目录，双击打开Arduino.exe应用程序

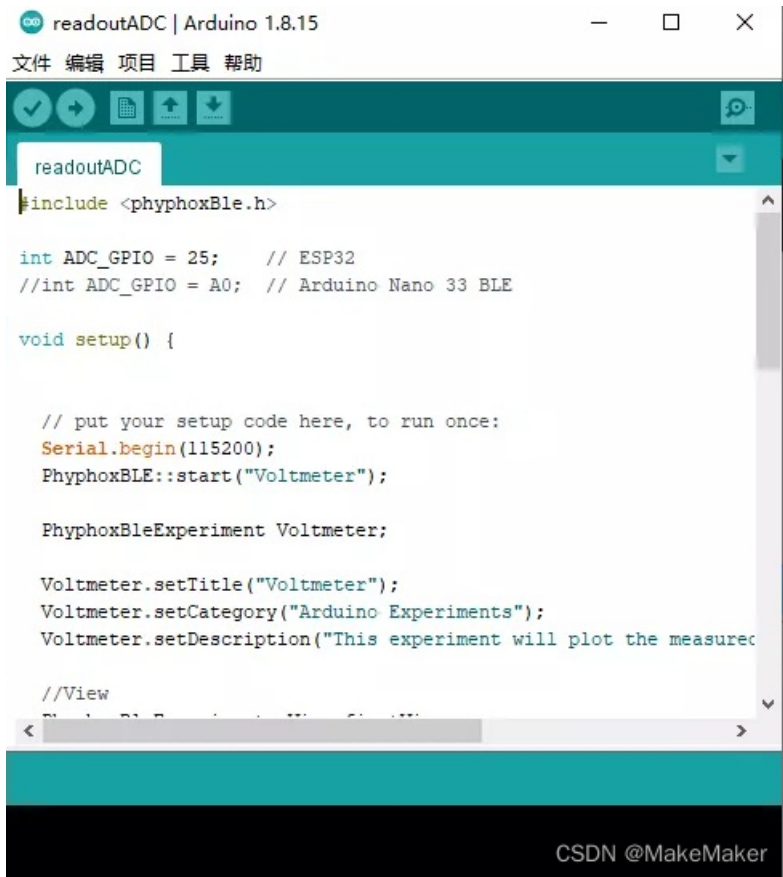
名称	修改日期	类型	大小
drivers	2021/5/14 16:24	文件夹	
examples	2021/5/14 16:25	文件夹	
hardware	2021/5/14 16:25	文件夹	
java	2021/5/14 16:25	文件夹	
lib	2021/5/14 16:25	文件夹	
libraries	2021/6/23 19:49	文件夹	
reference	2021/5/14 16:25	文件夹	
tools	2021/5/14 16:25	文件夹	
tools-builder	2021/5/14 16:25	文件夹	
arduino	2021/5/14 16:25	应用程序	72 KB
arduino.l4j	2021/5/14 16:25	配置设置	1 KB
arduino_debug	2021/5/14 16:25	应用程序	69 KB
arduino_debug.l4j	2021/5/14 16:25	配置设置	1 KB
arduino-builder	2021/5/14 16:25	应用程序	23,156 KB
libusb0.dll	2021/5/14 16:24	应用程序扩展	43 KB
msvcp100.dll	2021/5/14 16:25	应用程序扩展	412 KB
msvcr100.dll	2021/5/14 16:25	应用程序扩展	153 KB

这是可以在文件——示例中找到phyphox BLE的示例程序，我们打开最后一个程序：readoutADC



打开后如下

打开后如下：



```
readoutADC | Arduino 1.8.15
文件 编辑 项目 工具 帮助

readoutADC
#include <phyphoxBle.h>

int ADC_GPIO = 25; // ESP32
//int ADC_GPIO = A0; // Arduino Nano 33 BLE

void setup() {

    // put your setup code here, to run once:
    Serial.begin(115200);
    PhyphoxBLE::start("Voltmeter");

    PhyphoxBleExperiment Voltmeter;

    Voltmeter.setTitle("Voltmeter");
    Voltmeter.setCategory("Arduino Experiments");
    Voltmeter.setDescription("This experiment will plot the measured

//View

```

CSDN @MakeMaker

当然此刻你可以直接将ESP32插上电脑，点击上传按钮将这段程序上传到你的ESP32上，上传完后手机上的phyphox就可以通过蓝牙连接ESP32了，但是在此之前我向大家简单解释一下这个程序。别看程序很短，那是因为phyphox已经帮我们写好了库函数，也就是所有的工作其实它都已经在库函数里帮我们做好了，我们仅仅是通过几句特定的指令调用它已经写好的库函数而已。在程序中，“//”后面的，或者“/*...*/”中间包括的都为注释语句，看起来也是灰色的字体，也就是说这些是不会运行的，仅仅为了说明程序而写的，完全可以删除。所以我以后解释程序时也都会写在程序注释里。

```

#include <phyphoxBle.h>
int ADC_GPIO = 25;    // ESP32
//int ADC_GPIO = A0; // Arduino Nano 33 BLE
void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    PhyphoxBLE::start("Voltmeter");
    PhyphoxBleExperiment Voltmeter;
    Voltmeter.setTitle("Voltmeter");
    Voltmeter.setCategory("Arduino Experiments");
    Voltmeter.setDescription("This experiment will plot the measured voltage over time.");
    //View
    PhyphoxBleExperiment::View firstView;
    firstView.setLabel("Rawdata"); //Create a "view"
    //Graph
    PhyphoxBleExperiment::Graph firstGraph;    //Create graph which will plot random numbers over time
    firstGraph.setLabel("Voltmeter");
    firstGraph.setUnitX("s");
    firstGraph.setUnitY("V");
    firstGraph.setLabelX("time");
    firstGraph.setLabelY("Voltage");
    /* Assign Channels, so which data is plotted on x or y axis
       first parameter represents x-axis, second y-axis
       Channel 0 means a timestamp is created after the BLE package arrives in phyphox
       Channel 1 to N corresponding to the N-parameter which is written in server.write()
    */
    firstGraph.setChannel(0, 1);
    firstView.addElement(firstGraph);    //attach graph to view
    Voltmeter.addView(firstView);    //Attach view to experiment
    PhyphoxBLE::addExperiment(Voltmeter);    //Attach experiment to server
}
void loop() {
    //put your main code here, to run repeatedly:
    float voltage = 3.3 * analogRead(ADC_GPIO)/4095;
    delay(1);
    PhyphoxBLE::write(voltage);
    Serial.print("Voltage = ");
    Serial.println(voltage);
    delay(20);
}

```

以上是英文注释的程序，下面是我用中文注释后的程序，大家可以对比看看我修改了哪些。大家英文不好的可以看下面的程序，另外程序里有些地方也可以写中文的，这样手机端phyphox界面上也会相应的显示中文。里面有些地方可以改成你自己想要显示的文字，自己尝试的改吧。

```

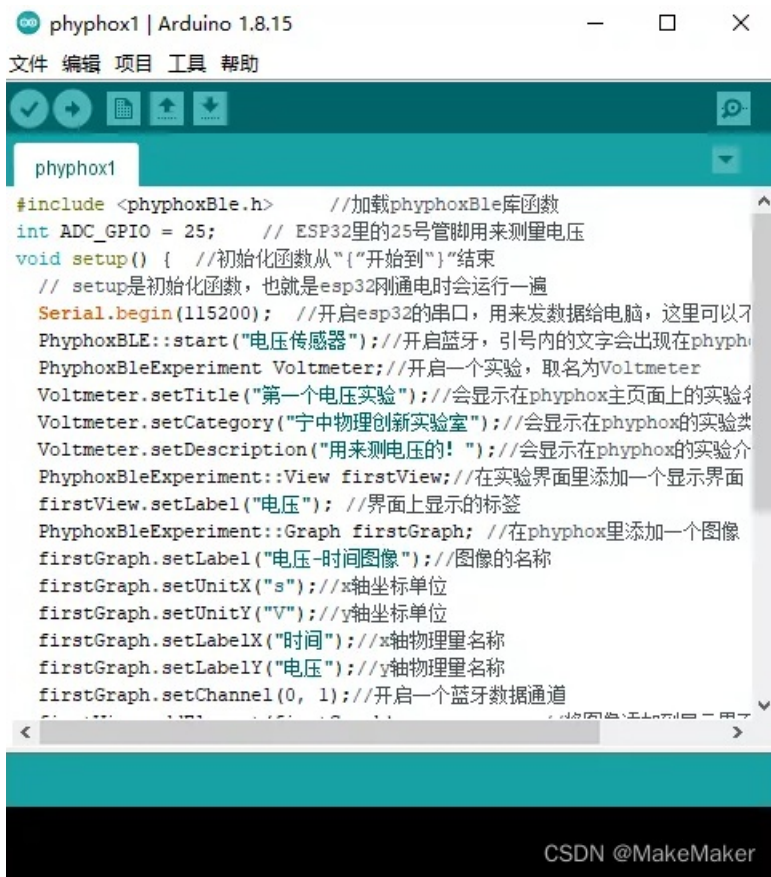
#include <phyphoxBle.h> //加载phyphoxBle库函数
int ADC_GPIO = 25; // ESP32里的25号管脚用来测量电压
void setup() { //初始化函数从“{”开始到“}”结束
  // setup是初始化函数,也就是esp32刚通电时会运行一遍
  Serial.begin(115200); //开启esp32的串口,用来发数据给电脑,这里可以不用
  PhyphoxBLE::start("电压传感器");//开启蓝牙,引号内的文字会出现在phyphox搜索蓝牙设备时
  PhyphoxBleExperiment Voltmeter;//开启一个实验,取名为Voltmeter
  Voltmeter.setTitle("第一个电压实验");//会显示在phyphox主页面上的实验名称
  Voltmeter.setCategory("宁中物理创新实验室");//会显示在phyphox的实验类别
  Voltmeter.setDescription("用来测电压的!");//会显示在phyphox的实验介绍里
  PhyphoxBleExperiment::View firstView;//在实验界面里添加一个显示界面
  firstView.setLabel("电压");//界面上显示的标签
  PhyphoxBleExperiment::Graph firstGraph;//在phyphox里添加一个图像
  firstGraph.setLabel("电压-时间图像");//图像的名称
  firstGraph.setUnitX("s");//x轴坐标单位
  firstGraph.setUnitY("V");//y轴坐标单位
  firstGraph.setLabelX("时间");//x轴物理量名称
  firstGraph.setLabelY("电压");//y轴物理量名称
  firstGraph.setChannel(0, 1);//开启一个蓝牙数据通道
  firstView.addElement(firstGraph); //将图像添加到显示界面上
  Voltmeter.addView(firstView); //将显示界面添加到实验
  PhyphoxBLE::addExperiment(Voltmeter); //将实验设置通过蓝牙发送到phyphox
} //初始化函数结束,在初始化函数里其实设置了一些phyphox中要显示的元素
// 以下为主程序, esp32在运行时主程序内的程序是循环运行的
// 也就是第一遍结束第二遍又开始,周而复始,直到断电
void loop() { //主程序开始
  float voltage = 3.3 * analogRead(ADC_GPIO)/4095;//voltage就是通过计算得到的电压值
  delay(1); //运行暂停1毫秒,让蓝牙模块准备好
  PhyphoxBLE::write(voltage);//将voltage通过蓝牙发送到手机端的phyphox
  delay(4);//暂停4毫秒,这里经测试最小可以填4,再小esp32就会卡死,也就是差不多200Hz
} //主程序结束

```

注意,代码中 `float voltage = 3.3 * analogRead(ADC_GPIO)/4095;`

是由于ESP32中包含了12位的ADC模块,也就是他能将模拟电压转换为数字,最大电压3.3V对应的是二进制里的“111111111111”,也就是十进制中的4095,所以需要将ESP32读取到的数字电压值乘以3.3再除以4095算得电压的模拟值。

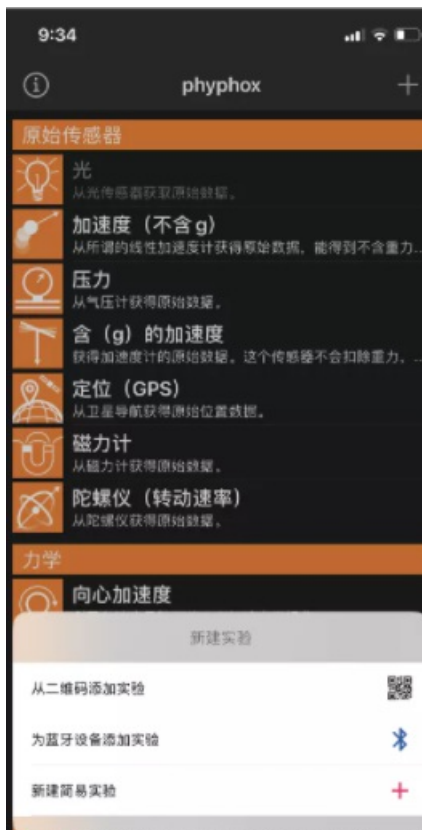
将我上面的代码复制好，粘贴到Arduino IDE内，然后点击上传按钮（编辑下面向右的箭头），在此之前需要在工具里设置好板类型和端口（请看第一篇的介绍），稍等片刻，直到下方黑色框内显示上传完毕就可以了。

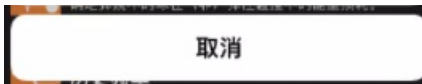


```
#include <phyphoxBle.h> //加载phyphoxBle库函数
int ADC_GPIO = 25; // ESP32里的25号引脚用来测量电压
void setup() { //初始化函数从“开始到”“结束”
  // setup是初始化函数，也就是esp32刚通电时会运行一遍
  Serial.begin(115200); //开启esp32的串口，用来发数据给电脑，这里可以不
  PhyphoxBLE::start("电压传感器");//开启蓝牙，引号内的文字会出现在phyphox
  PhyphoxBLEExperiment Voltmeter;//开启一个实验，取名为Voltmeter
  Voltmeter.setTitle("第一个电压实验");//会显示在phyphox主页面上的实验名称
  Voltmeter.setCategory("宁中物理创新实验室");//会显示在phyphox的实验分类
  Voltmeter.setDescription("用来测电压的！");//会显示在phyphox的实验介绍
  PhyphoxBLEExperiment::View firstView;//在实验界面里添加一个显示界面
  firstView.setLabel("电压");//界面上显示的标签
  PhyphoxBLEExperiment::Graph firstGraph;//在phyphox里添加一个图像
  firstGraph.setLabel("电压-时间图像");//图像的名称
  firstGraph.setUnitX("s");//x轴坐标单位
  firstGraph.setUnitY("V");//y轴坐标单位
  firstGraph.setLabelX("时间");//x轴物理量名称
  firstGraph.setLabelY("电压");//y轴物理量名称
  firstGraph.setChannel(0, 1);//开启一个蓝牙数据通道
```

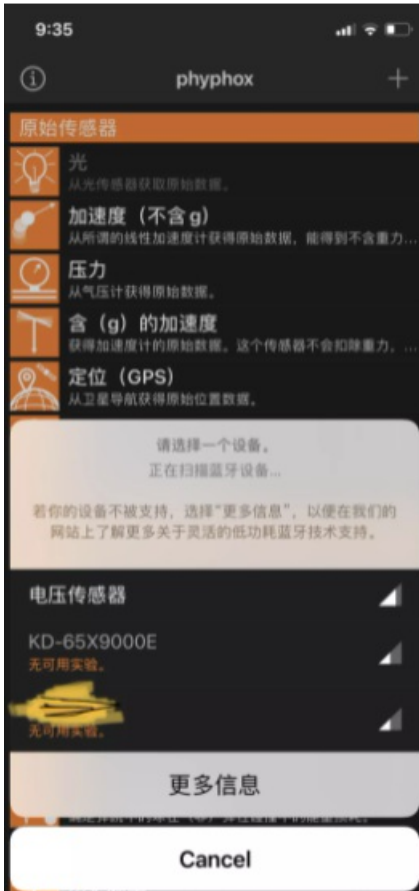
3、phyphox端的操作

经过上面的操作，我们已经完成了ESP32端的程序，怎么样，还是比较简单的吧，虽然可能看不懂，不过以后写多了就慢慢的懂了。





点击为蓝牙设备添加实验：



选择我们之前命名的“电压传感器”，打开后保存至收藏。此时就可以开始实验了。



可以在phyphox的主页面中找到我们刚才收藏的实验。





在视频中我们发现ESP32只测得了正向的电压，这是因为ESP的模拟端口只能测量正电压，而无法测量负电压，关于如何制作可以测量正负电压的电压传感器，在之后的篇章中会讲到。

视频中的实验器材由宁中物理创新实验室的小潘老师制作，我仅仅提供技术支持。也祝贺他在省自制教具评比中获得了二等奖。

这篇到这里就结束了，回顾一下这一篇的内容：

- (1) 从phyphox网站中下载库文件并粘贴到arduino中。
- (2) 开始学习第一个ESP32与phyphox的程序
- (3) 使用提供的示例程序完成了电压传感器的制作，并用它完成了电磁感应定量实验。