虚谷号体验文档目录



**体验人： 产品序列号 : 日期:**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 体验活动 | 是否完成 | 建议 |
| 1 | 观看虚谷号视频 |  |  |
| 2 | 浏览虚谷号官网 |  |  |
| 3 | 关注虚谷号公众号 |  |  |
| 4 | 认识虚谷号接口 |  |  |
| 5 | U盘模式体验 |  |  |
| 6 | U盘内容检查 |  |  |
| 7 | 查看 Arduino 控制13引脚的示例代码 |  |  |
| 8 | 拷贝Blink.ino 到指定路径控制D13LED灯 |  |  |
| 9 | Arduino支持成功，备份程序及log |  |  |
| 10 | 通过修改Arduino 程序控制D13 LED灯闪烁规律体 |  |  |
| 11 | .Python  GPIO案例测试（注：运行了Python程序则不能运行Arduino程序） |  |  |
| 12 | Wi-Fi连接 |  |  |
| 13 | 浏览器链接虚谷号的jupyter |  |  |
| 14 | vvBoard的格式化恢复 |  |  |
| 15 | 虚谷号主机模式连接外部设备 |  |  |
| 16 | 桌面程序体验 |  |  |
| 17 | 文本编辑保存 |  |  |
| 18 | WI-FI连接 |  |  |
| 19 | Arduino  IDE 编写点灯程序 |  |  |
| 20 | python编程 |  |  |
| 21 | 体验人工智能三个案例 |  |  |

虚谷号介绍：

**1虚谷号视频观看：**

复制如下链接，在浏览器上打开：

<https://v.youku.com/v_show/id_XNDA4MDAzMzY4NA==.html?spm=a2hzp.8244740.0.0>

**2虚谷号官网浏览：**

复制如下链接，在浏览器上打开：

<http://www.vvboard.com.cn/>

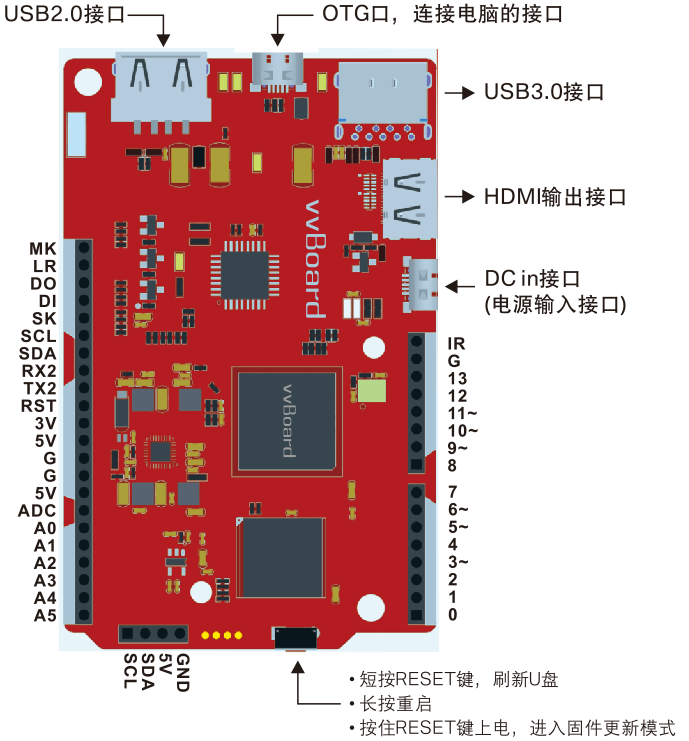
**3关注微信公众号：**



虚谷号接口说明：

**4 虚谷号接口：**

了解掌握虚谷号各接口功能，各接口的定义图如下：



**5 虚谷号U盘功能连接**

活动目标：电源灯显示红色，上电成功；

操作步骤：用USB线小头端连接虚谷号板OTG接口 另一端连接windos10、windos7电脑的USB口，如图5-1 所示:

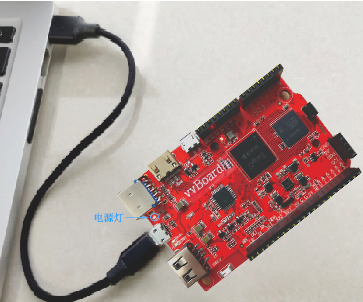


图5-1 电源灯位置见图中红框处

**6 虚谷号U盘内容检查**

活动目标：核对U盘内文件是否正确；

操作步骤：

a.连接好虚谷号之后，稍等20秒左右，在屏幕的右下角可以看到一个U盘图标如图6-1，在我的电脑里可以看到vvBoard盘符，如图6-2



图6-1

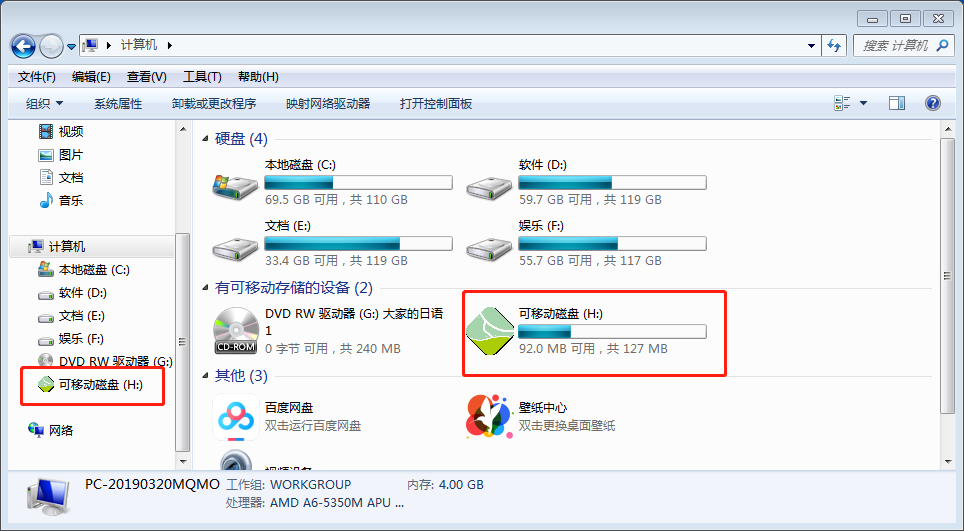


图6-2

b.打开U盘，可以看到以下文件，如图6-3，文件介绍如表6-1

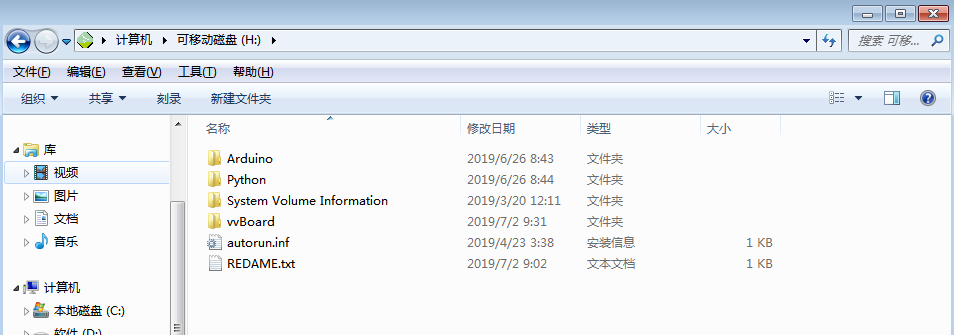


图6-3

|  |  |
| --- | --- |
| 名称 | 作用 |
| Arduino | 存放Arduino相关的文件 |
| Python | 存放Python相关的文件 |
| System Volume Information | 插入Windows电脑端显示出来的 |
| vvBoard | 存放vvBoard相关的文件 |
| autorun.inf | 配置U盘的盘符 |
| REDAME.txt | 查看版本信息 |

表6-1

c.进入Arduino，可以看到以下文件，如图6-4，文件介绍如表6-2

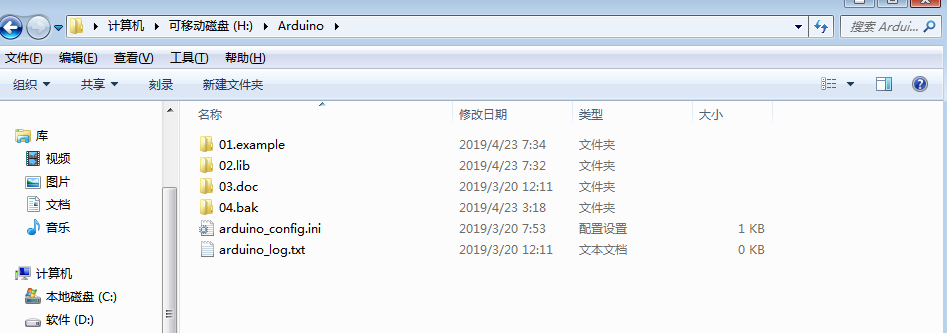


图6-4

|  |  |
| --- | --- |
| 名称 | 作用 |
| 01.example | 存放Arduino IDE里的所有案例 |
| 02.lib | 存放Arduino的库文件 |
| 03.doc | 存放Arduino相关的资料 |
| 04.bak | 执行完Arduino的程序放入这里 |
| arduino\_config.ini | 配置要执行的Arduino程序 |
| arduino\_log.txt | 执行Arduino程序打印的log |

表6-2

d.进入Python，可以看到以下文件，如图6-5，文件介绍如表6-3

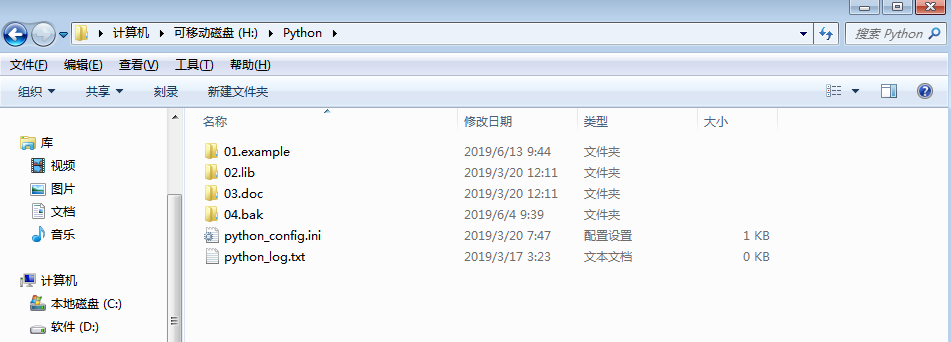


图6-5

|  |  |
| --- | --- |
| 名称 | 作用 |
| 01.example | 存放Python的案例 |
| 02.lib | 存放Python库文件 |
| 03.doc | 存放Python相关的资料 |
| 04.bak | 不做定义，为了和Arduino结构统一 |
| python\_config.ini | 配置要执行的Python程序 |
| python\_log.txt | 执行Python程序打印的log（未开通） |

表6-3

e.进入vvBoard，可以看到以下文件，如图6-6，文件介绍如表6-4

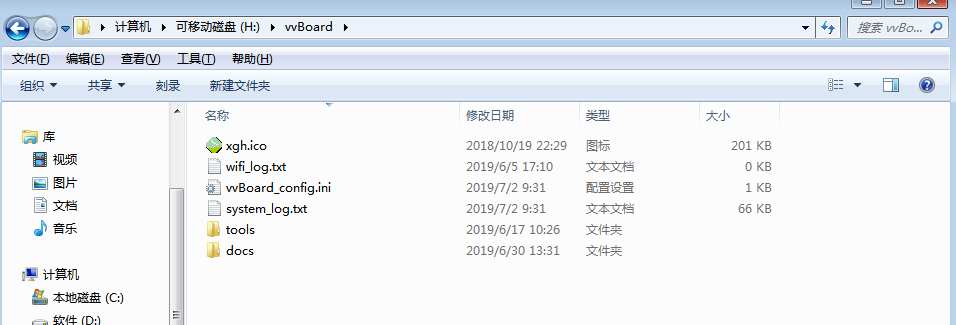


图6-6

|  |  |
| --- | --- |
| 名称 | 作用 |
| Docs | 存放vvBoard的说明文档 |
| system\_log.txt | 系统log |
| Tools | 存放工具 |
| vvBoard\_config.txt | 配置Wi-Fi和jupyter |
| wifi\_log.txt | 连接Wi-Fi打印的log |
| xgh.ico | U盘盘符 |

表6-4

**7** 查看 Arduino 控制13引脚的示例代码

活动目标：查看 Arduino 控制13引脚的示例代码 （Blink.ino）

操作步骤：

a、点击 Arduino文件夹，进入如图7-1



图7-1

b、依次点击进入01.example\01.Basics\Blink文件夹内， 如图7-2

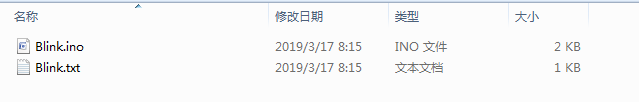


图7-2

c、打开查看Blink.ino，使用记事本打开，如图7-3，如果打开排版和图中不一样，关闭，选中文件右击选择“打开方式”->选择“更多应用（其它程序）”->选择“写字板”打开

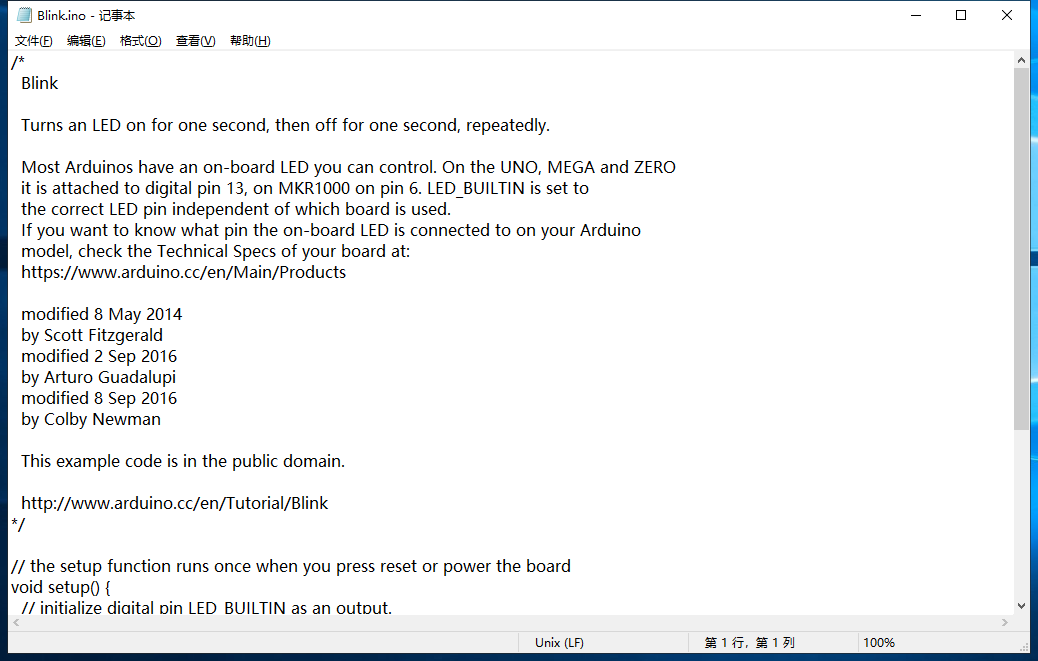


图7-3

如果文件名称中没有显示后缀名.ino，需要设置windos10系统显示扩展名，Windows10系统的进入“我的电脑”->选择“查看”->勾选“文件扩展名”，如图7-4，

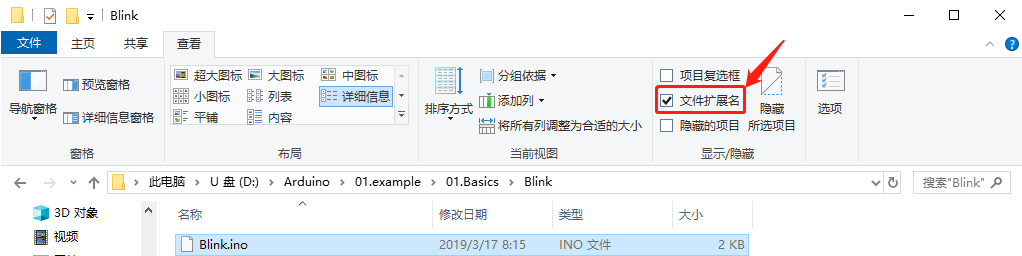


图7-4

如果文件名中没有后缀名.ino文件，需要设置windos7系统显示扩展名，Windows7系统的进入“我的电脑”->选择“工具”->选择“文件夹选项”->点击查看->在高级设置中找到“隐藏已知文件类型的扩展名”把勾选去掉，如图7-5

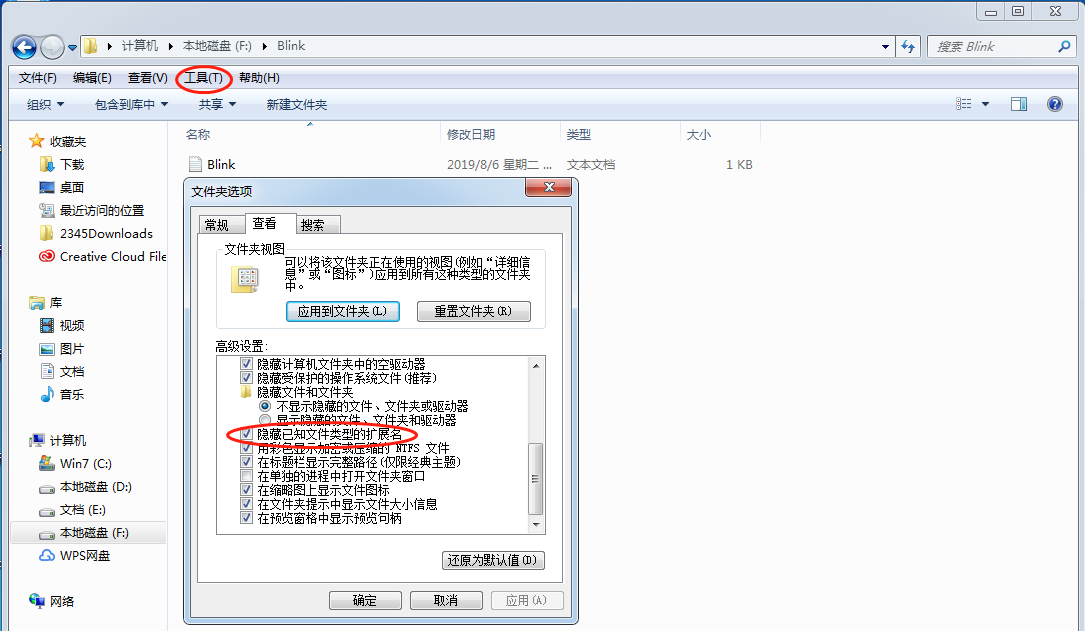


图7-4

d、关闭打开的记事本

**8拷贝Blink.ino 到指定路径控制D13LED灯**

活动目标：拷贝Blink.ino 到指定路径控制D13 LED灯成功

操作步骤：

a.将Arduino\01.example\01.Basics\Blink里的Blink.ino文件复制到Arduino文件夹内，如图8-1所示：

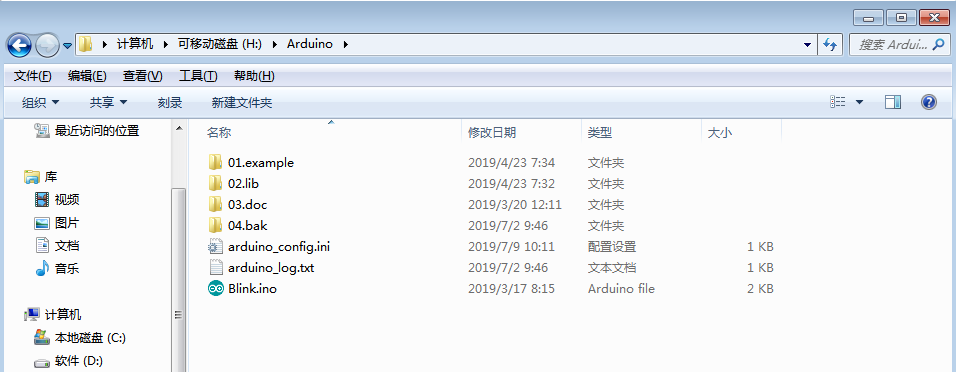


图8-1

b.用记事本打开Arduino文件夹里的arduino\_config.ini，将内容改为Arduino=Blink.ino，如图8-2，保存后关闭退出。

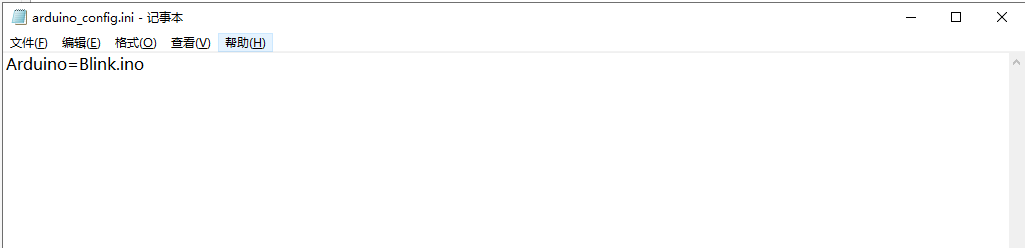


图8-2

c.查看虚谷号板子上D13 LED灯为亮1秒熄灭1秒的闪烁，短按reset键，U盘图标就会消失，等待20秒后。会看到虚谷号板子上的D13 LED灯在快速闪烁，U盘图标重新出现，D13 LED灯会按程序写的亮1秒熄灭1秒的闪烁。D13 LED灯和reset按键位置如图8-3所示：

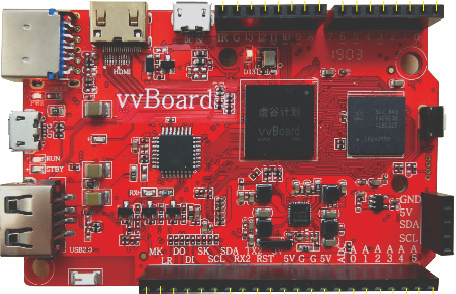


图8-3

注：红色框的为D13 LED灯，绿色框的为reset按键

**9Arduino支持成功，备份程序及log**

活动目标：Arduino支持成功，备份程序及log成功确认

操作步骤：

a.Arduino文件夹里的Blink.ino文件消失。

b.在Arduino/04.bak文件夹中能够看到Blink.ino文件，如图9-1所示：

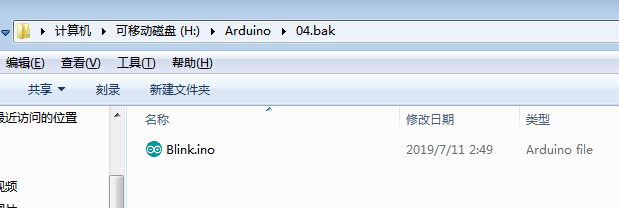


图9-1

c.在Arduino/ardruino\_log.txt文件中可以看到执行完成的log，使用记事本打开，如图9-2所示：

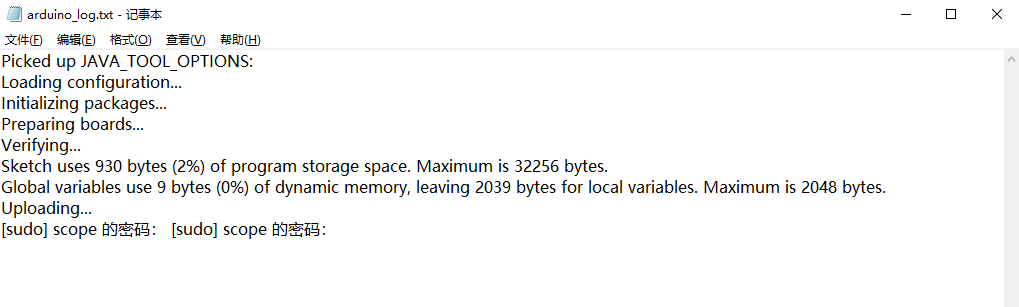


图9-2

高级：

**10修改Arduino 程序控制D13 LED灯闪烁规律**

活动目标：通过修改Arduino 程序控制D13LED灯闪烁规律

操作步骤：

a.将Blink.ino从Arduino/04.bak复制到Arduino文件夹内

b.使用记事本打开Blink.ino文件，里面的digitalWrite(LED\_BUILTIN, HIGH)为灯亮，digitalWrite(LED\_BUILTIN, LOW)为灯灭，通过修改延迟时间（delay（））里面的值来控制灯亮灭的时间

c.进行更改灯的闪烁规律，例如改为亮3秒，熄灭5秒，如图10-3，保存关闭

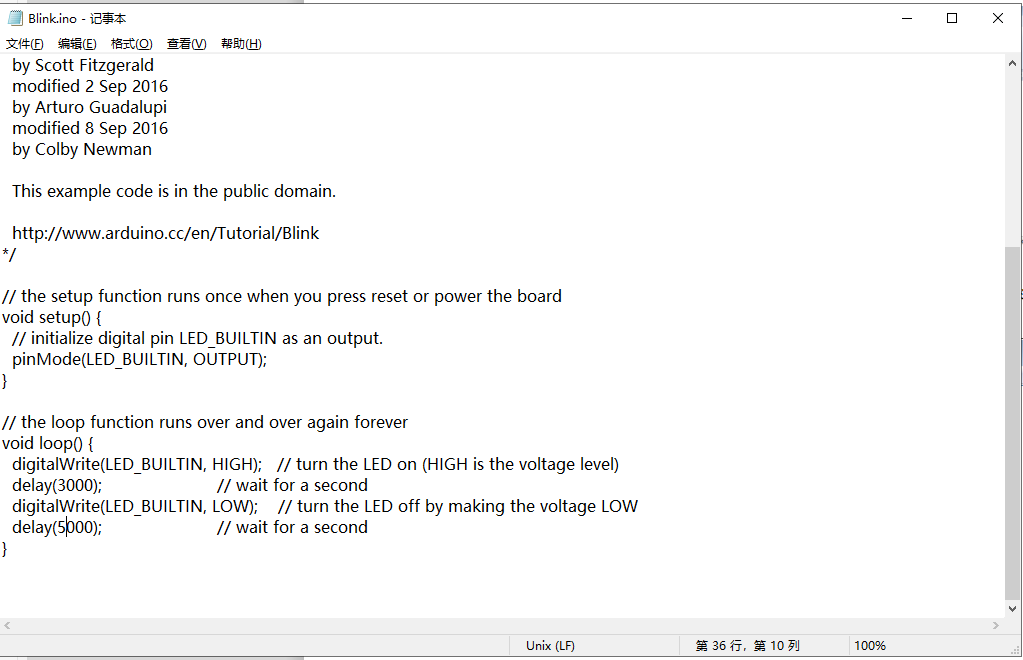


图10-3

d.用记事本打开Arduino文件夹里的arduino\_config.ini，将内容改为Arduino=Blink.ino，如图10-4，保存关闭。

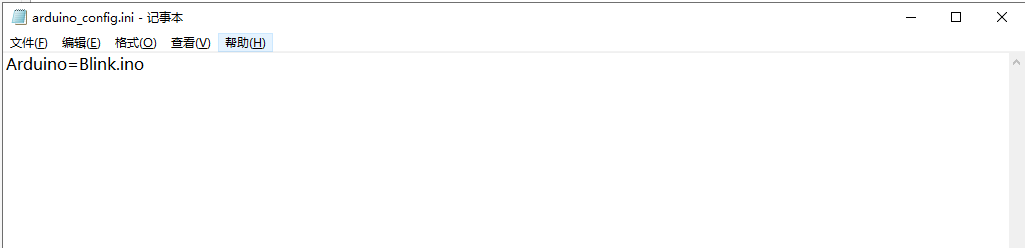


图10-4

e.短按reset键，U盘图标会消失，看到D13灯在快速闪烁，U盘图标重新出现，之后便可以看到D13灯亮3秒熄灭5秒的闪烁。D13灯和reset按键位置如图10-5所示：

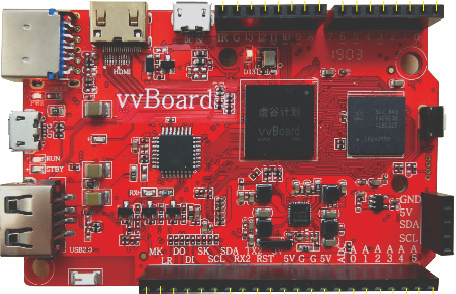


图10-5

**注：**红色框的为D13LED灯，绿色框的为reset按键

e.确认Arduino文件夹里Blink.ino文件消失

f.在Arduino/04.bak文件夹中能够看到Blink.ino文件，使用写字板打开Arduino/04.bak/Blink.ino，查看内容为更改后的内容，如图10-6所示：

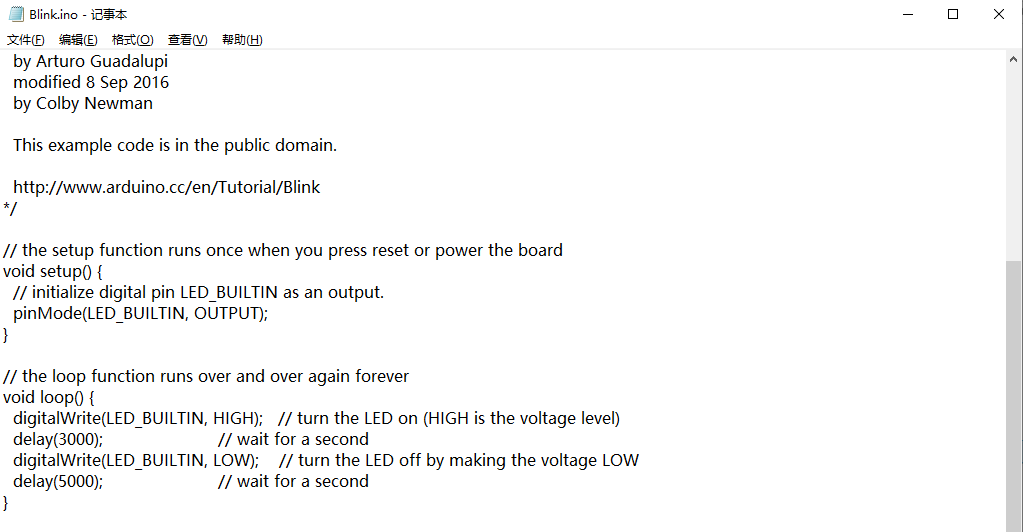


图10-6

g、在Arduino/ardruino\_log.txt文件中可以看到执行完成的log，使用记事本打开，如图10-7所示：

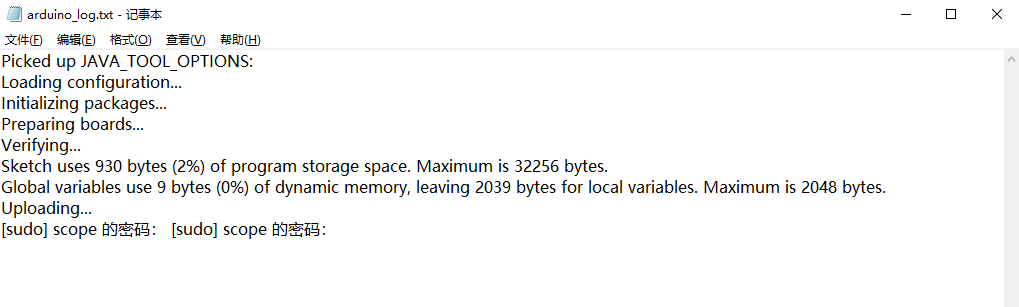


图10-7

**11Python  GPIO案例测试（注：运行了Python程序则不能运行Arduino程序）**

活动目标：可以成功执行Python\01.example\10.GPIO中的每个案例

操作步骤：

a.进入Python\01.example\10.GPIO，查看是否存在以下案例，如表11-1

|  |  |
| --- | --- |
| 案例名称 | 执行后现象 |
| analog\_input.py |  |
| analog\_write.py |  |
| digital\_input\_master.py |  |
| led\_pin.py | D13灯亮1秒熄灭1秒的闪烁 |
| pin\_out\_high.py |  |
| pin\_out\_low.py |  |
| read\_analog\_20\_times.py |  |
| servo.py |  |
| xugu-blink.py | D13灯亮1秒熄灭1秒的闪烁 |

表11-1

b.以led\_pin.py为例，将led\_pin.py拷贝到Python文件夹里，如图11-1所示：

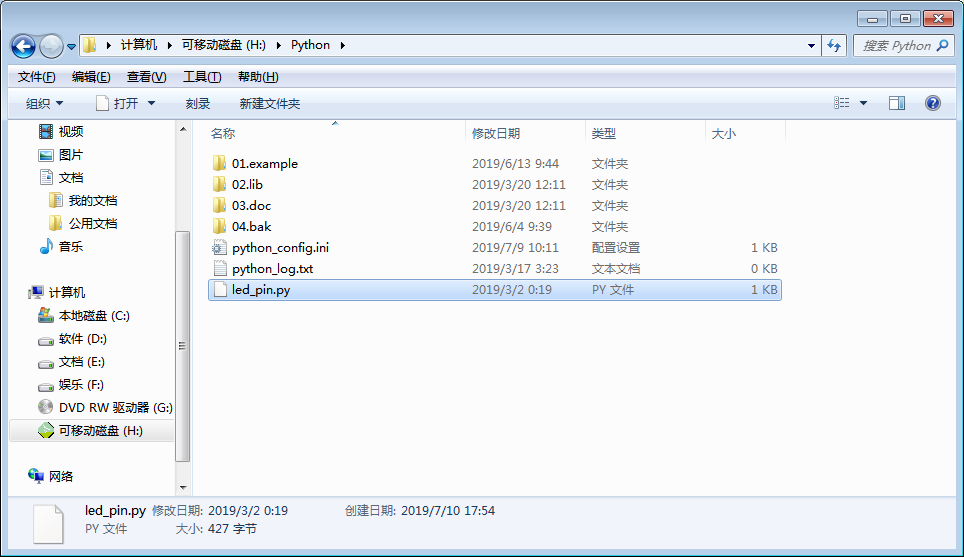


图11-1

用记事本打开Python文件夹内的python\_config.ini，将内容改为Python=led\_pin.py，如图11-2，保存关闭



图11-2

c.短按reset键，U盘图标会消失，直到D13灯快速闪烁结束后（一定要等到这个现象出现），便看到D13灯会亮1秒熄灭1秒的闪烁，并且U盘图标会重新出现

d.Python\01.example\10.GPIO里的其他的案例都按照led.py的操作步骤运行一遍，运行完后的现象会如表d-1所示

**12WI-FI 连接(只支持2.4G的网络）**

活动目标：通过log文件，可以获取到当前WI-FI的IP地址，以Wi-Fi用户名为SCOPE-Sam，密码为scope12345为例 （用户请输入自己的WI-FI用户名和账号）；

操作步骤：

a.用记事本打开vvBoard文件夹内的vvBoard\_config.ini，将内容改为图12-1所示，保存关闭

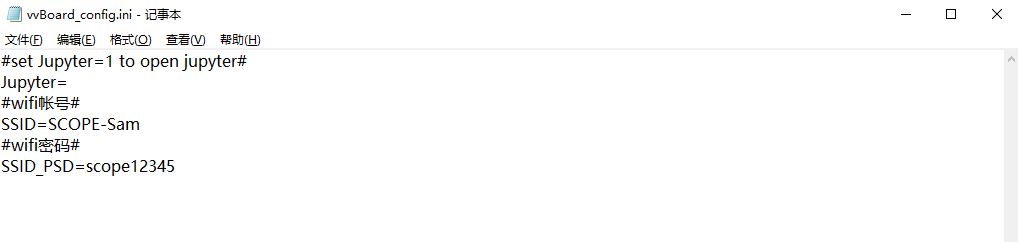


图12-1

b.短按reset按键，等U盘图标重新出现，则会连接WI-FI，并且用记事本打开vvBoard文件夹内的wifi\_log.txt看到的log如图12-2所示：



图12-2

**注：**有图中红框标内容说明Wi-Fi连接上了，红框内的地址为当前WI-FI的IP地址（后面会用到）

c.如果没有连接上，打开vvBoard文件夹内的wifi\_log.txt看到的log如图12-3所示

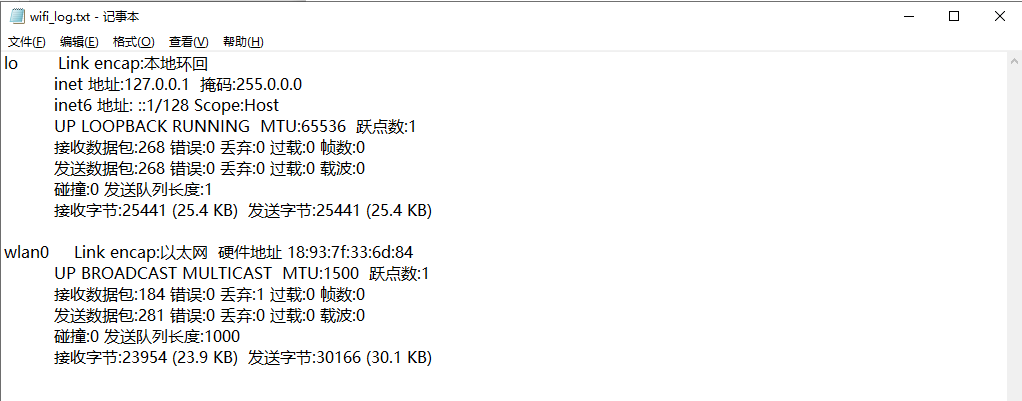


图12-3

**注：**图12-3的红框内容说明Wi-Fi未连接

**13浏览器链接虚谷号的jupyter**

活动目标：成功在电脑上使用浏览器链接虚谷号的jupyter；

操作步骤：

a.使用此功能前需先连接WI-FI，并且虚谷号连接的WI-FI（上一个测试连接的WI-FI）要和电脑端连接的WI-FI一致（或者电脑端网线和你使用的WI-FI是同一个路由器上的）。

b.用记事本打开vvBoard文件夹内的vvBoard\_config.ini，内容改为Jupyter=1，如图13-1，保存关闭

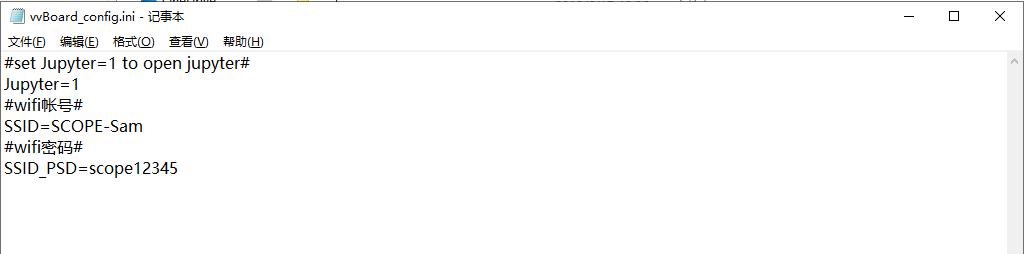


图13-1

c.短按reset按键，等U盘图标重新出现

d.在电脑端打开浏览器，输入http://xxx.xxx.x.xxx:8888（xxx.xxx.x.xxx为图13-2红框内的IP地址)，进入网页后会提示输入密码，密码为scope，便可以链接虚谷号，浏览和操作虚谷号上文件 如图13-2，如果打开为空白页，如图13-3，则更换浏览器打开

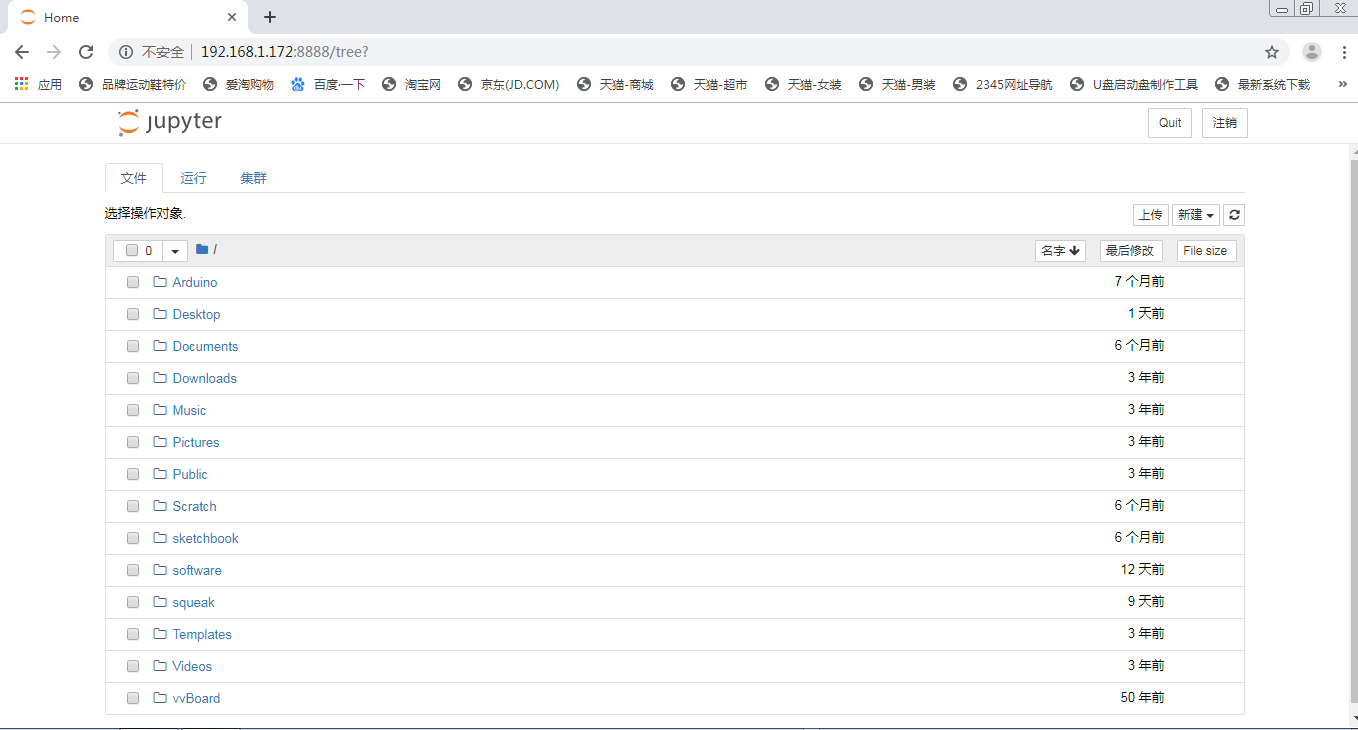


图13-2

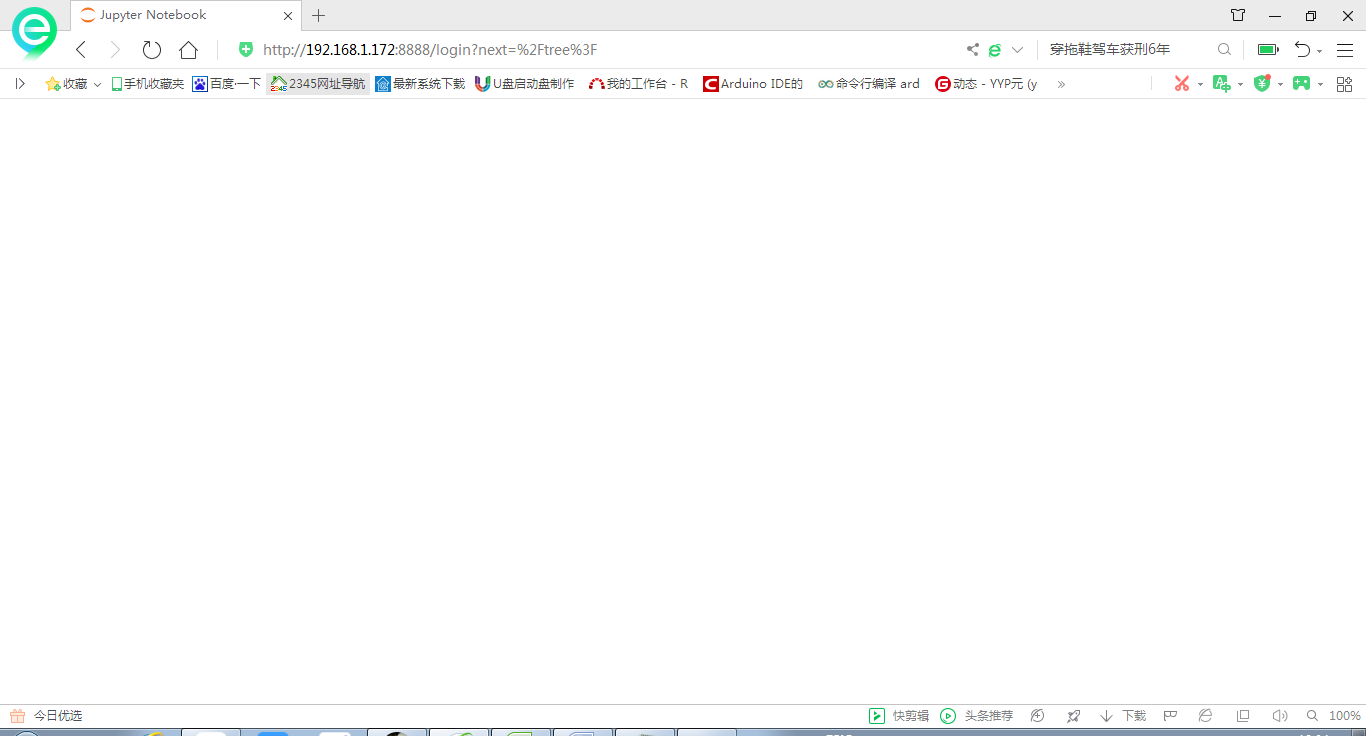


图13-3

**14vvBoard的格式化恢复**

活动目标：对vvboard盘内进行删除或格式化等操作，可以按reset 恢复；

操作步骤：

a.删除文件夹或文件，例如：进入vvboard盘，删除Arduino文件夹或Arduino下的arduino\_config.ini文件

b.短按reset按键，等U盘图标重新出现，发现vvboard 盘内删除的文件恢复，和核对盘内文件初始状态一样

主机模式：

**15虚谷号主机模式连接外部设备**

活动目标：电源灯正常亮起和显示器上可以看到系统进入到桌面

操作步骤：

a.连接方法，对照鼠标、键盘、电源、显示屏（LCD）的连接方式连接，如图15-1所示：

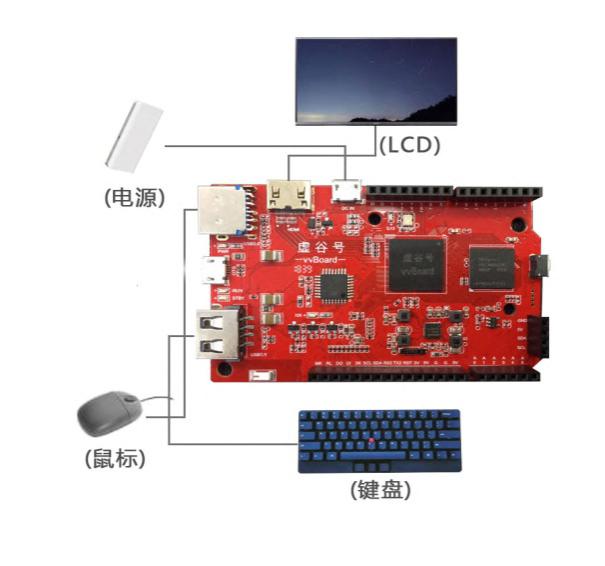


图15-1

b.查看电源灯是否有亮，显示屏是否有显示，进入桌面，桌面如图15-2所示：



图15-2

**16桌面程序**

确认桌面程序都存在，并了解大体功能,如表16-1：

|  |  |  |
| --- | --- | --- |
| **名称** | **图标** | **介绍** |
| **回收站** |  | 回收站，删除的文件会放到这里面 |
| **VvBoard** | vvboard | 虚拟U盘目录，可用于与其他电脑连接 |
| **Jupyter** | jupyter | 是一个交互式笔记本, 支持运行40+种编程语言. 可以用来编写漂亮的交互式文档 |
| **Python（3.5）** | python3 | Python3.5的编辑器，可以运行python代码 |
| **Arduino IDE** | arduino-arduinoide | Arduino IDE是Arduino产品的软件编辑环境。主要用于各类单片机程序的开发、编译和烧入的工具 |
| **S4A** | s4a | S4A是使Arduino开源硬件平台能够简单编程的Scratch修改版 |
| **Scratch** | scratch | Scratch是麻省理工学院的“终身幼儿园团队”（Lifelong Kindergarten Group）开发的图形化编程工具，版本1.4 |
| **LX终端** | lxterminal | 命令行输入窗口 |

表16-1

**17 文本编辑保存**

活动目标：能进行文本编辑保存

操作步骤：

a.在桌面空白处右击，选择创建文档->空文件，弹出创建新文件弹框，如图17-1所示：

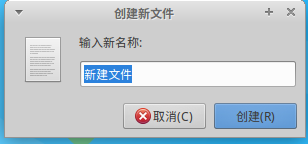


图17-1

b.在弹框内输入创建的文件名，例如text.txt，选择创建，便可在桌面看到text.txt文件，如图17-2所示：



图17-2

c.选中文件右击选择用”gedit”打开。

d.打开文件后，在里面随意输入内容。

e.点击右上角的保存，便保存了编辑的内容，然后点击右上角的“×”关闭文本，右上角图标如图17-3所示：



图17-3

f.再重新打开文本便可以看到上次保存的内容

**18 WI-FI连接**

活动目标：进行WI-FI连接

操作步骤：

a.找到连接WI-FI的图标为以下两种情况：图18-1为没连接WI-FI的图标，图18-2为连接上WI-FI的图标:

5.png 6.png

图18-1 图18-2

b.右击图标，可以看到所有可以连接的Wi-Fi网络，如图18-3所示：

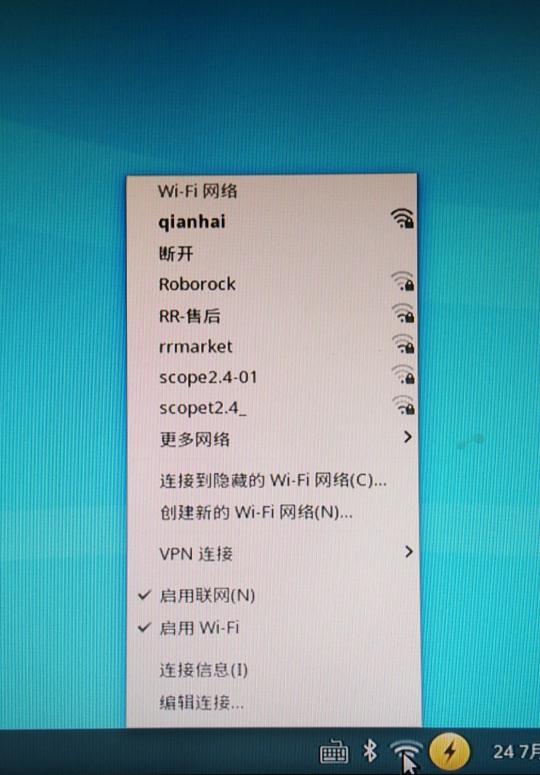


图18-3

c.选择你需要连接的WI-FI名（如果Wi-Fi已经连接，你要选择另一个Wi-Fi），点击进入，会弹出需要Wi-Fi网络认证的弹框，如图18-4所示：

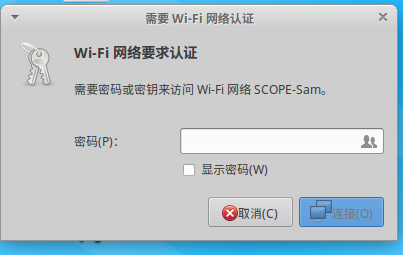


图18-4

d.在密码框内输入WI-FI密码，点击连接，便可连接，且图标如图18-2所示。

**19Arduino  IDE 编写点灯程序**

活动目标：通过Arduino  IDE编辑器可以执行修改Arduino程序，控制D13LED灯

操作步骤：

a.打开桌面的Arduino IDE，便可以看到打开了Blink程序，如图19-1所示：



图19-1

b.点击12.png，便可以编译执行Blink程序，编译执行成功会显示 Done uploading…，如图19-2红框内容：

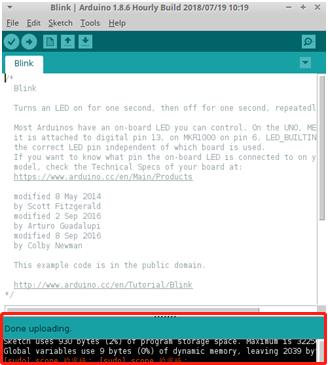


图19-2

c.完成后，D13灯会闪烁,规律为亮1秒暗1秒。

d.修改程序里面delay得值，来控制D13灯，如图19-3所示，修改后会变成亮5秒，暗2秒的闪烁规律：

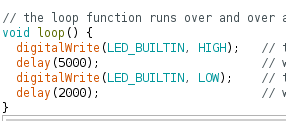
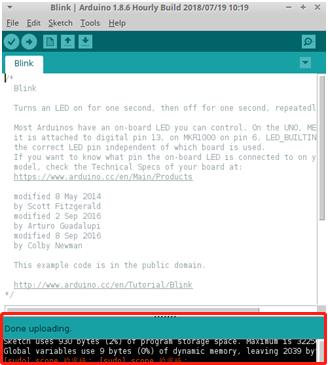


图19-3

e.点击12.png，便可以编译执行Blink程序，编译完成标志如图19-4红框内容：



图c-2

f.完成后，D13灯会亮5秒熄灭2秒的闪烁，关闭窗口。

**20python编程**

活动目标：python编程通过控制Arduino 13脚变换D13 LED灯的闪烁规律，例：D13LED 灯亮2秒熄灭4秒的闪烁

操作步骤：

a.在桌面创建一个文件，命名为led.py

b.打开led.py，手动输入下面方框内的代码（或将桌面vvBoard里的Python\01.example\10.GPIO\xugu\_blink.py文件内容复制到led.py内，并改成和下面的代码一样），保存关闭。

import time # 导入time模块

from xugu import LED # 从xugu库中导入LED类

led = LED(13) # 初始化LED类

while True: # 用循环实现持续地开灯关灯，到达闪烁的效果

led.on() # 点亮连接13号引脚的LED灯

time.sleep(2) # 持续2秒

led.off() # 关闭LED灯

time.sleep(4) # 持续4秒

c.在桌面空白处右击，选择Open Terminal Here，则会打开终端，如图20-1所示：

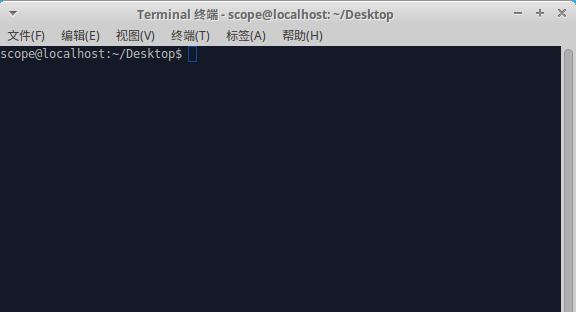


图20-1

d.输入python led.py，按下回车键（Eenter），稍等一会，会出现如图20-2所示内容：

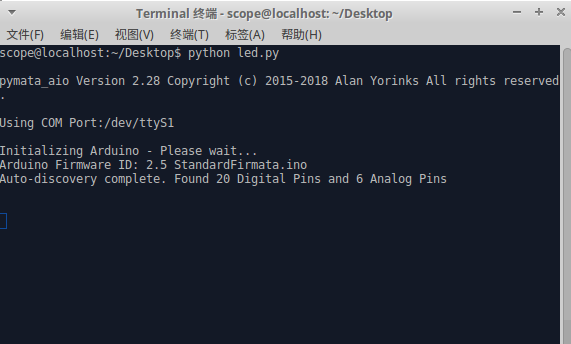


图20-2

e.执行成功，D13LED灯会亮2秒熄灭4秒的闪烁。关闭窗口

**21人工智能三个案例**

活动目标：浏览人工智能三个案例参考文档

操作步骤：双击打开vvBoard图标，进入Python\03.doc\90.python进阶文件夹内，能看到人工智能三个案例的参考文档如表21-1所示：

|  |  |
| --- | --- |
| 案例名称 | 参考文档 |
| 动植物识别 | 动植物识别操作文档.pdf |
| 人脸识别 | 人脸识别操文档.pdf |
| 数字识别 | 数字识别操作文档.pdf |

表21-1