



Edison 使用入门

余翀

INTEL SSG

- 网络资料

- 英特尔物联网开发技术资源:

- https://software.intel.com/zh-cn/articles/iot_developer_resource

- 英特尔物联网开发人员社区:

- <https://software.intel.com/zh-cn/iot>

- 如何开箱起步:

- <https://software.intel.com/zh-cn/iot/library/edison-getting-started>

- 下载相关软件:

- <https://software.intel.com/zh-cn/iot/hardware/edison/downloads>

- 官方论坛:

- <https://communities.intel.com/community/tech/edison/content>

- 官方技术文档:

- <https://www-ssl.intel.com/content/www/us/en/support/edison-documents-and-guides.html>

- 常见FAQ与技术博客:

- <https://www-ssl.intel.com/content/www/us/en/support/edison-troubleshooting.html>

- <https://software.intel.com/zh-cn/iot/hardware/edison/docs>

主要内容

- Edison连线相关
 - MINI USB
 - 串口
- 更新OS
- 网络连接
 - WIFI
 - USB转Ethernet
- 应用开发
 - Arduino
 - Mraa
 - opkg

第一部分：连线相关

...

Mini USB, 串口, 驱动

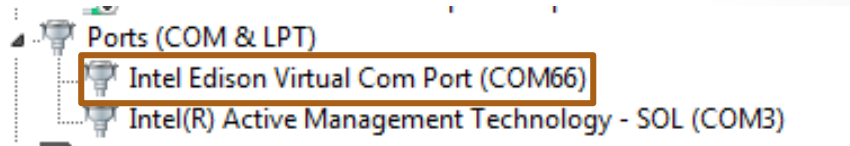
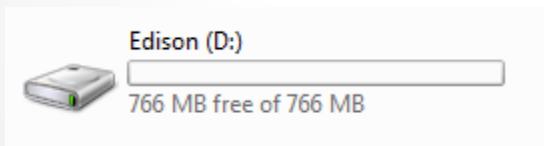
连线



1. 外接电源, 7-15V
2. 标准USB
 - 使用时必须外部供电
3. USB选择开关
 - 标准和mini只能选择一个使用
4. Miniusb: 用于
arduino下载, OS
下载
 - 不推荐用他供电
5. 串口 (自带USB转
串口)

USB驱动

- 该驱动安装后，mini-USB才有用
- 下载
<http://downloadmirror.intel.com/24909/eng/IntelEdisonDriverSetup1.2.1.exe>
- 安装
- 用连接Edison mini-USB和电脑（注意USB选择开关）
- 成功的表现



串口驱动

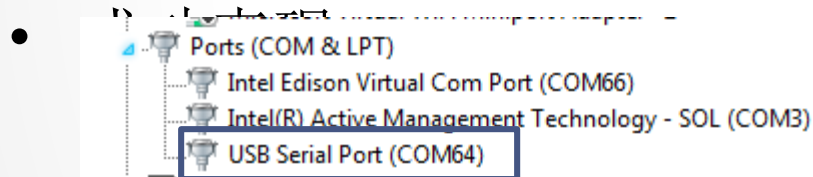
- 安装DTFI驱动后，可以用串口访问Edison

- 下载

<http://www.ftdichip.com/Drivers/D2XX.htm>

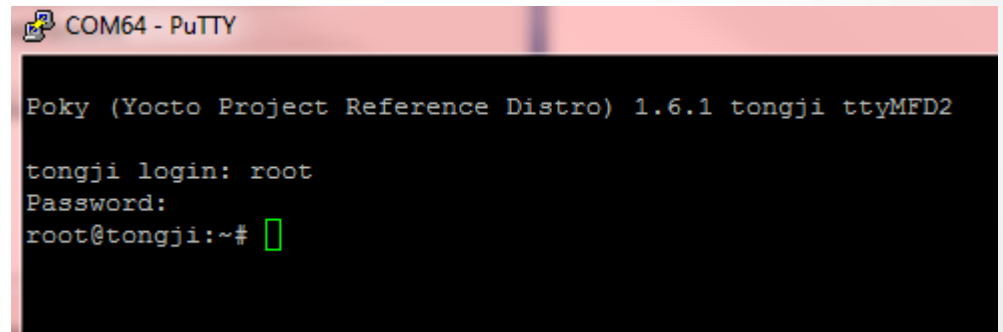
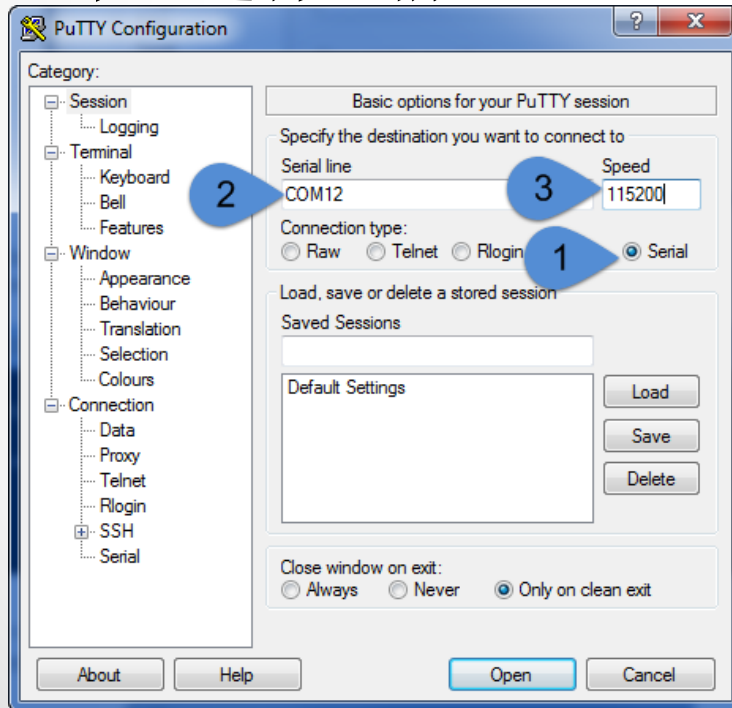
- 安装

- 连接Edison串口与电脑



串口登录Edison

- 下载安装putty或类似软件
<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>
- 串口选择之前USB Serial Port的号码



这里使用最新OS，且设置了hostname和密码之后会讲述设置方法

- <https://software.intel.com/en-us/setting-up-serial-terminal-intel-edison-board>

第二部分：更新OS

...

Flash Tool

为什么要更新

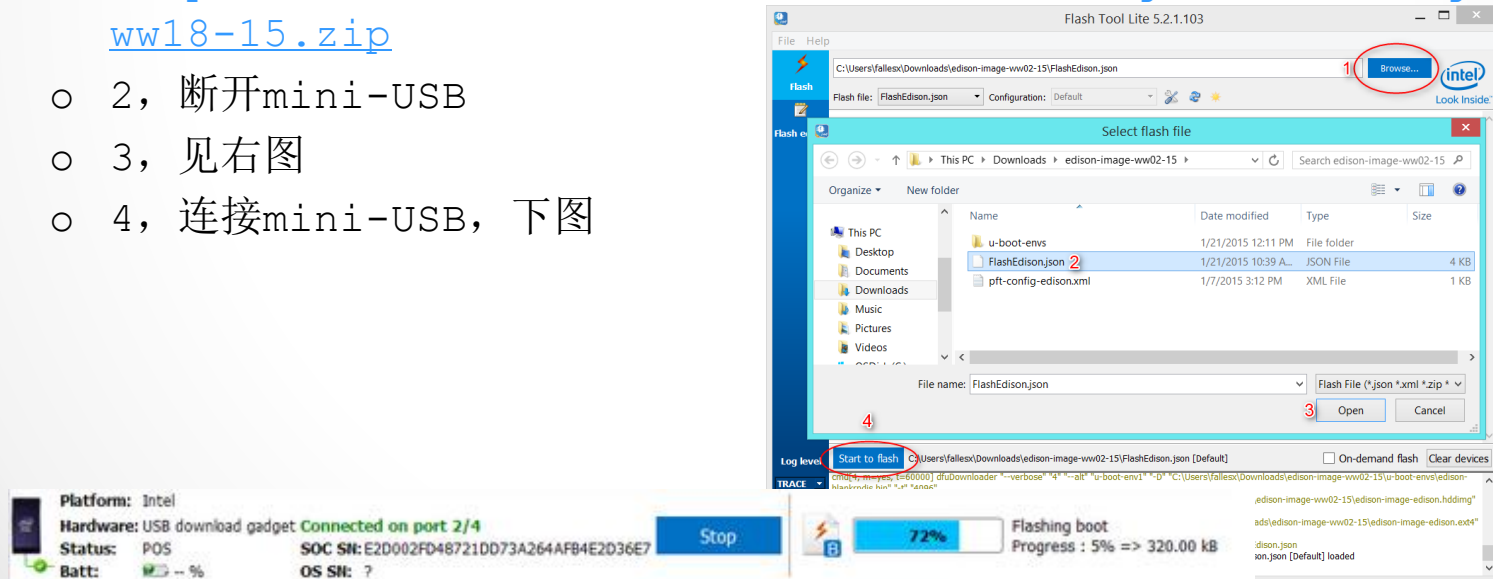
- 初始版本的OS (image, firmware) 有较多bug, 且功能较弱
 - 串口吞字
 - Rootfs空间太小 (在edison中, “df -h”)
 - GPIO7影响Wifi
 - . . .
- 查看BSP Release Note可以看到不同版本OS的功能和已知bug列表 (能做什么, 不能做什么)
 - http://download.intel.com/support/edison/sb/edison_rn_332032009.pdf

更新方法选择

- `reboot ota`
 - 最简单，但是不是完整的更新，无法改变rootfs大小
 - 跨平台
- `Sudo ./flashall.sh`
 - 完整更新
 - 只能在ubuntu下
- **Flash tool**
 - 官方推荐，完整更新
 - 需要安装软件和额外驱动

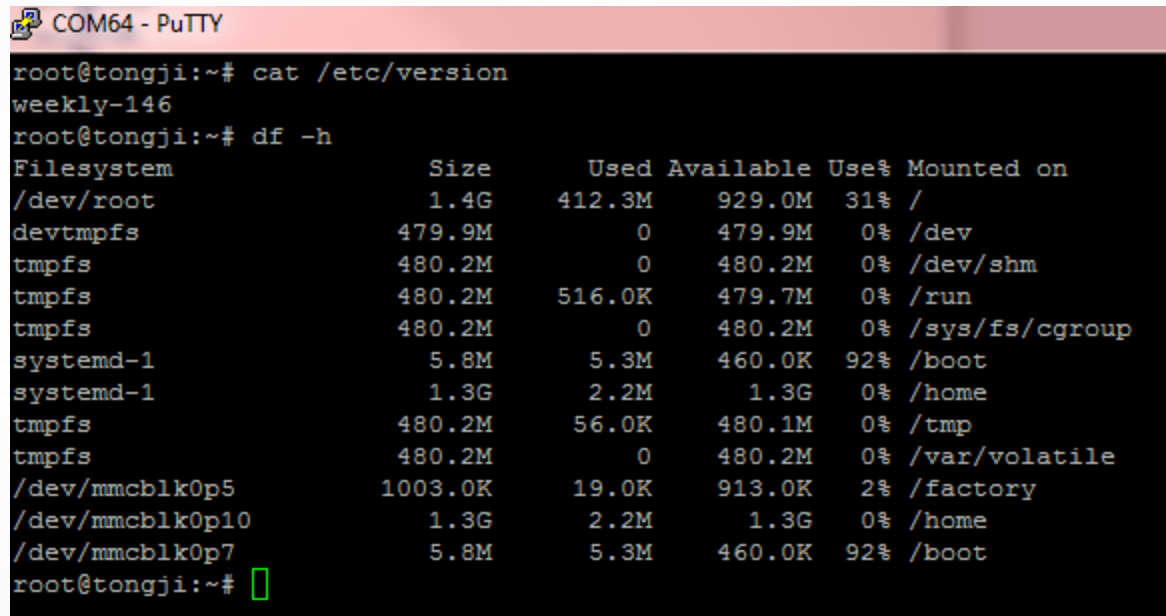
Flash Tool

- 下载
http://downloadmirror.intel.com/24910/eng/PhoneFlashToolLite_5.2.4.0_win32.exe
- 安装（会自动安装相关驱动）
- 使用
 - 1，下载最新OS： WW18-15
<http://downloadmirror.intel.com/24910/eng/edison-image-ww18-15.zip>
 - 2，断开mini-USB
 - 3，见右图
 - 4，连接mini-USB，下图



成功标志

- 1, 登录Edison
- 2,



```
COM64 - PuTTY
root@tongji:~# cat /etc/version
weekly-146
root@tongji:~# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                  1.4G    412.3M    929.0M   31% /
devtmpfs                  479.9M          0    479.9M    0% /dev
tmpfs                     480.2M          0    480.2M    0% /dev/shm
tmpfs                     480.2M    516.0K    479.7M    0% /run
tmpfs                     480.2M          0    480.2M    0% /sys/fs/cgroup
systemd-1                  5.8M      5.3M    460.0K   92% /boot
systemd-1                  1.3G      2.2M      1.3G    0% /home
tmpfs                     480.2M    56.0K    480.1M    0% /tmp
tmpfs                     480.2M          0    480.2M    0% /var/volatile
/dev/mmcblk0p5             1003.0K    19.0K    913.0K    2% /factory
/dev/mmcblk0p10            1.3G      2.2M      1.3G    0% /home
/dev/mmcblk0p7             5.8M      5.3M    460.0K   92% /boot
root@tongji:~#
```

定制OS

- 如果需要其他内核模块，详见BSP User Guild
http://download.intel.com/support/edison/sb/edisonbsp_ug_331188007.pdf
- 例：要安装USB camera的uvc驱动

```
Device Drivers
Multimedia support
  Media USB Adapters [*]
    USB Video Class [M]
      UVC input events device support [*]
```

第三部分：网络连接

...

Wifi Ethernet SSH

WIFI

- 官方文档: Wi-Fi user guild
<http://www.intel.com/support/edison/sb/CS-035380.htm>
- 经验:
 - 第一次登录时 `configure_edison --setup`, 因此设置密码 hostname 和 Wi-Fi
 - 之后每次启动会自动连接 Wi-Fi, 如果 Wi-Fi 改动了, `configure_edison --wifi`
 - 务必执行一次 **`--setup`**, 开启密码后才能使用 **`ssh`** 服务

USB转Ethernet

- 利用标准USB口接入网线
 - 注意USB选择开关

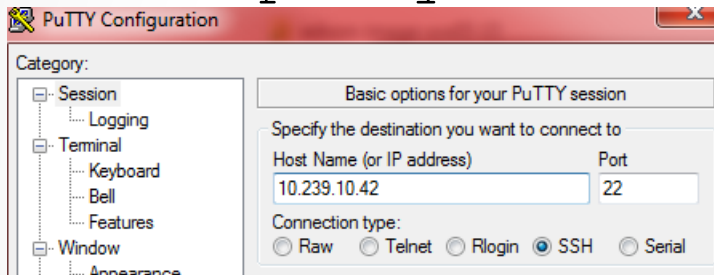


1. 登录Edison, 输入 `systemctl start connman`
 - 之后WiFi会被禁用, 。重启后恢复
 - Systemctl 还有其他形式的命令, enable, disable等
2. `connmanctl enable ethernet`

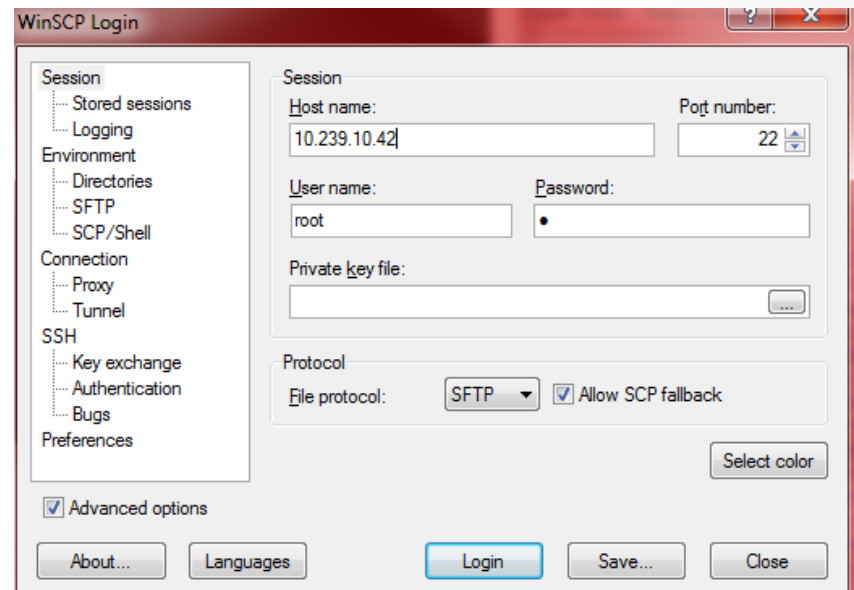
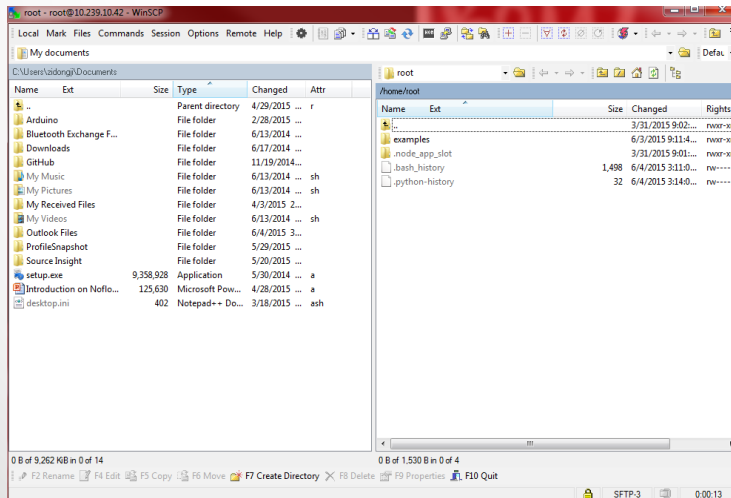
```
root@tongji:~# ifconfig
enp0s17u1 Link encap:Ethernet  HWaddr 00:24:9c:04:50:06
          inet addr:10.239.10.42  Bcast:10.239.10.255  Mask:255.255.255.0
          inet6 addr: fe80::224:9cff:fe04:5006/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST DYNAMIC MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:2 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1804 (1.7 KiB)  TX bytes:2962 (2.8 KiB)
```

SSH连接

- 主要运行了`-setup`，ssh就开启了
- 可以使用putty进行ssh登录（不需要串口线了）



- 也可以使用winscp传文件



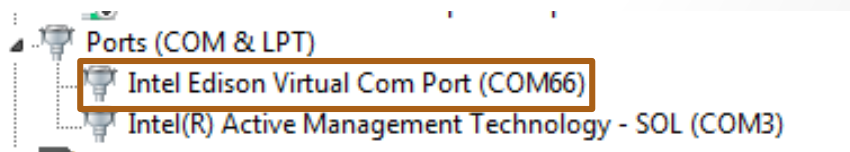
第四部门：应用开发

...

Arduino mraa opkg

Arduino开发

- 1, 下载Arduino IDE
<http://downloadmirror.intel.com/24783/eng/IntelArduino-1.6.0-Windows.7z>
- 2, 解压即可, 运行arduino.exe
- 3, 编写Arduino代码 (或者直接用file->example)
- 4, 用mini-USB连接Arduino和电脑
- 5, Tools里选择board->Edison, Port->COM66
- 6, upload即可, 程序会下载进入Edison并马上运行。
之后开机即运行



Arduino例子

- Serial: 就是mini-USB
 - 输入结果在Tools->serial monitor
- Serial1: Arduino上的串口 (GPIO0,1)
 - 用于和其他模块的通信

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

Arduino的原理

- 每一个Arduino的API都是Edison Yocto Linux下的C实现。本质就是在写一个C程序
 - 代码详见 `\IntelArduino-1.6.0-Windows\arduino-1.6.0+Intel\hardware\intel\i686`
 - 所以可以自行添加其他libc和stdc++的头文件，并调用相应库函数
- 编译完成后，可执行程序在`sketch->show sketch folder`中
- 利用mini-USB将该程序命名为`sketch.elf`存入Edison `/sketch`中
 - 如果驱动出错，传输失败，就用winscp手动传输
- 启动脚本会自动执行`/sketch/sketch.elf`
 - 不想要执行，`rm /sketch/sketch.elf`
 - 或者 `systemctl disable clloader`

MRAA/UPM

- Mraa: C/C++/python/nodejs库, 操作底层IO
 - <https://github.com/intel-iot-devkit/mraa>
- UPM: C/C++/python/nodejs库, 操作主流sensors
 - 底层用mraa
 - <https://github.com/intel-iot-devkit/upm>
- **Edison**上已经安装了这两个库, 在自己的程序中使用该库, 即可操作外设。
- 使用方法:
 - 直接在Edison上开发 `gcc -o hello hello.c -lmraa` (我的做法)
 - 直接在Edison上写python/nodejs程序来调用 (我的做法)
 - 使用eclipse: <https://software.intel.com/en-us/installing-the-eclipse-ide-intel-edison-board>
 - 在开发机上建立交叉编译环境:
<http://www.intel.com/support/edison/sb/CS-035382.htm>

MRAA例子

```
#include "mraa.hpp"

int
main()
{
    uint16_t adc_value;
    float adc_value_float;
    mraa::Aio* a0;

    a0 = new mraa::Aio(0);
    if (a0 == NULL) {
        return MRAA_ERROR_UNSPECIFIED;
    }

    for (;;) {
        adc_value = a0->read();
        adc_value_float = a0->readFloat();
        fprintf(stdout, "ADC A0 read %X - %d\n", adc_value, adc_value);
        fprintf(stdout, "ADC A0 read float - %.5f\n", adc_value_float);
    }

    return MRAA_SUCCESS;
}
```

```
import mraa

print (mraa.getVersion())

try:
    x = mraa.Aio(0)
    print (x.read())
    print ("%5f" % x.readFloat())
except:
    print ("Are you sure you have an ADC?")
```

```
var m = require('mraa');

var analogPin0 = new m.Aio(0);
var analogValue = analogPin0.read();
var analogValueFloat = analogPin0.readFloat();
console.log(analogValue);
console.log(analogValueFloat.toFixed(5));
```


OPKG

- Edison安装Yocto Linux，可以认为就是在linux上开发。但是apt在哪里呢？
- Yocto用opkg进行包管理。有一个还不错的非官方源

1. 写/etc/opkg/base-feeds.conf

```
src/gz all http://repo.opkg.net/edison/repo/all
src/gz edison http://repo.opkg.net/edison/repo/edison
src/gz core2-32 http://repo.opkg.net/edison/repo/core2-32
```

2. opkg update

3. 按需下载 opkg install

- opkg install vim
- opkg install cmake
- ...

Thank You